The minimum G_c cut problem

Itamar Elem [1], Refael Hassin [2], and Jérôme Monnot[4,3]

1. itamar.elem@gmail.com

2. School of Mathematical Sciences, Tel-Aviv University, Tel-Aviv 69978, Israel. hassin@post.tau.ac.il

3. PSL, Université Paris-Dauphine,
Place du Maréchal de Lattre de Tassigny, 75775 Paris Cedex 16, France
4. CNRS, LAMSADE, UMR 7243
monnot@lamsade.dauphine.fr

Abstract. In this paper we study the complexity and approximability of the G_c-cut problem. Given a complete undirected graph $K_n = (V; E)$ with $|V| = n$, edge weighted by $w(v_i, v_j) \geq 0$ and an undirected cluster graph, $G_c = (V_c, E_c)$, with $|Vc| = k$, a k-cut is a partition $V_1, \ldots, V_k$ of $V(G)$ such that $V_i \neq \emptyset$ for $i = 1, \ldots, k$. The G_c-cut problem is to compute a k-cut minimizing $\sum_{(i,j)\in E_c} w(V_i, V_j)$ where $w(V_i, V_j) = \sum_{p\in V_i, q\in V_j} w(p, q)$. Denote G_c as cluster graph and its vertices as clusters. We show that the G_c-cut problem is **NP**-hard and even not approximable in the general case and remains **NP**-hard for cluster trees. In particular, we give a complete characterization of hard cases for cluster graphs with at most four vertices by proving that the G_c-cut problem is **NP**-hard if and only if G_c is isomorphic to $2K_2$. We also identify some cases where the G_c-cut problem is either polynomial or **NP**-hard. Finally, we propose polynomial approximation results for the G_c-cut problem when the edge weights of G satisfy the triangle inequality, or when the weights are strictly positive.

Keywords: Cut in graphs, **NP**-hardness, polynomial, approximation algorithms.

1 Introduction

The problem considered in this paper is a generalization of the the minimum k-cut problem, and it can be defined as follows:

Definition 1. *Let $K_n = (V, E)$ be a complete undirected graph with $|V| = n$ and edge weights $w(v_i, v_j) \geq 0$. Given is also an undirected cluster graph, $G_c = (V_c, E_c)$, with $|V_c| = k$. The G_c-*CUT PROBLEM *is to compute a k-cut minimizing $\sum_{(i,j)\in E_c} w(V_i, V_j)$, where $w(V_i, V_j) = \sum_{(p,q)\in V_i\times V_j,(p,q)\in E} w(p, q)$. The restriction to metric distance w (i.e., satisfying triangular inequality[1]) is called the* METRIC G_c-CUT PROBLEM.

Cut problems in graphs are important optimization problems because VLSI system design, parallel computing systems, clustering, network reliability and cutting planes, etc. appearing in real-life situations may often be modeled as graph partitioning problems (see for instance [1, 22]). A survey on the approximability of cut problems can be found in Shmoys [23]. The k-CUT PROBLEM has been well studied in the literature and consists of finding a partition $V_1, \ldots, V_k$ such that $V_i \neq \emptyset$, $i = 1, \ldots, k$ (called k-cut) of the vertices

[1] $\forall x, y, z \in V$, $w(x, y) \leq w(x, y) + w(y, z)$.

$V(G)$ of a simple graph $G = (V, E)$ edge weighted by $w(v_i, v_j) \geq 0$, minimizing $\sum_{1 \leq i < j \leq k} w(V_i, V_j)$. Goldschmidt and Hochbaum [6] proved that the problem in ordinary graphs is **NP**-hard when k is part of the input and gave the first polynomial-time algorithm for fixed k with running time $n^{O(k^2)}$. Since the results of Goldschmidt and Hochbaum [6] on the minimum k-cut problem, many other results are appeared in the literature. For instance, the running time of their algorithm has been improved by Kamidoi et al. [15] and Xiao [28]. Currently, the best results are the $O(n^{2(k-1)} \log n^3)$-time Monte Carlo algorithm due to Karger and Stein [12] and the $O(n^{2k})$-time deterministic algorithm due to Thorup [26]. Furthermore, Nagamochi et al. [19, 20] proved that the minimum k-cut problem can be solved in $O(mn^k)$ time for $k = 4, 5, 6$. The minimum k-cut problem has also drawn much attention in the literature for small values of k. The minimum 2-cut problem is commonly known as the minimum cut problem. Another version, the minimum 2-way cut problem, is the minimum (s, t) cut problem, which asks to find a minimum cut that separates two given vertices s and t. These two problems are fundamental problems in the subject of graph connectivity. For ordinary graphs, the minimum cut problem can be solved in $O(mn + n^2 \log n)$ time by Nagamochi and Ibaraki's algorithm [19] or Stoer and Wagner's algorithm [24], and the minimum (s, t) cut problem can be solved in $O(mn \log \frac{n^2}{m})$ time by Goldberg and Tarjan's algorithm [8]. For the minimum 3-cut problem in ordinary graphs, Kapoor [10] and Kamidoi et al. [15] showed that it can be solved by using $O(n^3)$ maximum flow computations. Burlet and Goldschmidt [3] and Nagamochi and Ibaraki [19] improved the result to $O(n^2)$. The Multiway k-cut problem for $k \geq 2$ is one generalization of the minimum (s, t) cut problem. This problem also known as the Multiterminal k-cut problem can be defined as follow: given a weighted complete graph, $K_n = (V, E)$ and a set of terminals $S = \{s_1, \ldots, s_k\}$, a multiway cut is a set of edges that leaves each of the terminals in a separate component. In other words, the goal of the Multiway k-cut problem is to find a k-cut $(V_1, \ldots, V_k)$ where $s_i \in V_i$ of minimum weight. The Multiway k-cut problem is know to be polynomial for $k = 2$ and and **NP**-hard when $k \geq 3$ is fixed [4].

When the cluster graph G_c is a k-clique, the k-CUT PROBLEM and the G_c-CUT PROBLEM coincide. In contrast, we show that the G_c-CUT PROBLEM is **NP**-hard when k is fixed.

In this paper, we mainly study the complexity and the approximability of the G_c-CUT PROBLEM according the structure of the cluster graph G_c. In Section 2, the notations and main definitions are introduced. In Section 3, complexity results are presented while the Section 4 gives some polynomial solvable cases for the G_c-CUT PROBLEM. For instance, as a corollary of the results given in this paper we will show for the cluster graphs G_c with at most 4 vertices, the G_c-CUT PROBLEM is **NP**-hard if and only if $G_c = 2K_2$. Finally, in Section 5, we propose polynomial approximation results when the weights are either positives or satisfy the triangle inequality. More exactly for the general case, we present a α-approximation in linear time where $w_{\min} = \min_{e \in E} w(e)$, $w_{\max} = \max_{e \in E} w(e)$, and $\alpha = \frac{w_{\max}}{w_{\min}}$ (here, we assume that $w_{\min} > 0$) and a 3-approximation is given for the METRIC G_c-CUT PROBLEM when the number of vertices of the cluster graph is fixed.

2 Definitions and preliminaries

All graphs in this paper are finite, simple and loopless. Let $G = (V, E)$ be a graph. An edge between u and v will be denoted (u, v). For a vertex $v \in V$, let $N_G(v)$ denote the set of vertices in G that are adjacent to v, i.e., the neighbors of v, and the degree of v is $d_G(v) = |N_G(v)|$. A *leaf* is a vertex v such that $\deg_G(v) = 1$. For $S \subset V(G)$, the *neighborhood* of S is $N_G(S) = \{v \in V : \exists u \in S, (u, v) \in E\}$. In particular, $N_G^2(v) = N_G(N_G(v))$. Vertex u is a *nested neighbor* of vertex v if $(u, v) \notin E$ and $N_G(u) \subseteq N_G(v)$. They are *twins* if $N_G(u) = N_G(v)$. The *contracted graph of* G *from* S, denoted $G(S)$, is the simple graph $G(S) = (V', E')$ where $V' = V \setminus S \cup \{v_S\}$ and $(u, v) \in E'$ if $u, v \notin S \cup \{v_S\}$ and $(u, v) \in E$ or if

$u = v_S$, $v \notin S \cup \{v_S\}$ and $\exists s \in S$ with $(s,v) \in E$. Throughout this paper, we use the following notation for a edge weighted graph (G,w): for $E^{'} \subseteq E(G)$, $w(E^{'}) = \sum_{e \in E'} w(e)$ and for $v \in V(G)$ and $U \subseteq V(G)$, $w(v,U) = \sum_{u \in U} w(v,u)$.

Now, we indicate some classes of graphs used in this paper:

Definition 2. Consider a graph $G = (V,E)$ such that $|V| = k$.

1. G is a *matching graph*, denoted wK_2, if $k = 2w$ and $\deg(v) = 1 \ \forall v \in V$.
2. G is a *path graph*, denoted P_k, if its edges form an induced path on k vertices.
3. G is a *cycle graph*, denoted C_k, if its edges constitute an induced cycle on k vertices.
4. $G = K_{d_1,\ldots,d_r} = (V,E)$ is a *complete r-partite graph* if there exists a partition $L_1,\ldots,L_r$ of V with $d_i = |L_i|$, and $\sum_{i=1}^{r} d_i = k$ such that for every $i \neq j \in \{1,\ldots,r\}$, $u,v \in L_i \Rightarrow (u,v) \notin E$ and $u \in L_i, v \in L_j \Rightarrow (u,v) \in E$. The sets L_i are the *color classes* of G. A *biclique* is a complete bipartite graph.
5. $G = (S,K;E)$ is a *split graph* if $V(G) = S \cup K$ where $K \cap S = \emptyset$, S is a stable set and K is a clique of G of maximum size. It is called *restricted split graph* if $deg_G(v) < |V(G)| - 1$ for every $v \in K$. For instance, $P_4 = (v_1,v_2,v_3,v_4)$ is a restricted split graph with $K = \{v_2,v_3\}$ and $S = \{v_1,v_4\}$.

Let us start by giving some definitions:

Definition 3. Let G_c be a cluster graph with $|V(G_c)| = k$. The MULTIWAY G_c-CUT PROBLEM is the G_c-CUT PROBLEM where given $I = (G,w)$, $S \subseteq V(G_c)$ and $|S|$ vertices $\{v_1,\ldots,v_{|S|}\} \subseteq V$ with $|S| \leq k$, we want to find an optimal G_c-cut $V_1,\ldots,V_k$ on I such that $v_i \in V_i$ for $i \in S$. The d-SIZE RESTRICTED G_c-CUT PROBLEM is the G_c-cut problem where an integer vector $(d_1,\ldots,d_k)$ is given with the instance. The goal is to find an optimal G_c-cut $(V_1,\ldots,V_k)$ on I where $n \geq \sum_{i=1}^{k} d_i$ and such that $|V_i| \geq d_i$. Finally, the RESTRICTED G_c-CUT PROBLEM is the d-SIZE RESTRICTED G_c-CUT PROBLEM when $n = \sum_{i=1}^{k} d_i$. In other words, $|V_i| = d_i$ for every $i = 1,\ldots,k$.

Note that if the cluster graph G_c is a complete r-partite for some r (in particular, a complete graph), then the MULTIWAY G_c-CUT PROBLEM with $|S| = 1$ is equivalent to the G_c-CUT PROBLEM, and then is polynomial if k is fixed.

Theorem 1. *The complexity of the* MULTIWAY K_k-CUT PROBLEM *when $k \geq 2$ is fixed is polynomial when $|S| = 2$ and **NP**-hard when $|S| > 2$.*

Proof. We divide the proof to the following sub cases.

1. $|S| = k$.
   (a) $|S| = 2$. The problem is equivalent to the minimum (s,t) cut problem. Thus, the problem is polynomial.
   (b) $|S| \geq 3$. The problem is equivalent to the minimum Multiway k-cut problem with $|S| \geq 3$. Hence, the problem is **NP**-hard.
2. where $k > |S|$. Denote $l = k - |S| > 0$
   (a) $|S| > 2$. Let $K_n = (V,E), S \subset V, |S| > 2$ be an instance of the minimum Multiway $|S|$-cut problem. Consider an instance of the MULTIWAY K_k-CUT PROBLEM described as follow:
      – $S' = S$.
      – $K_{n+l} = (V',E')$.
      – $V' = V \cup \{x_1,\ldots,x_l\}$
      – $E' = E \cup E_1 \cup E_2$ where

  – $E_1 = \{(x_i, v) : i = 1, \ldots, l, v \in V\}$ and $w(e) = 0$ for $e \in E_1$.
  – $E_2 = \{(x_i, x_j) : 1 \leq i < j \leq l\}$ and $w(e) = 0$ for $e \in E_2$.

  Solving the MULTIWAY G_c-CUT PROBLEM where $G_c = K_k$ for K_{n+l}, S' optimally we must assign each $x_i, i \in 1, \ldots, l$ to a single cluster of K_k so that the vertices of V must arrange in an optimal minimum multiway cut on the rest of the $|S|$ clusters of K_k, and the result follow.

(b) $|S| = 2, S = \{s, t\}$. We use the same construction as in [6] in a small modification that we enumerate over all the cores S such that $s \in S$, and terminals T such that $t \in t$ instead of enumerating all the cores S and terminals T as done at [6]. The rest of the proof is exactly like in [6].

Lemma 1. *Assume that vertex* 1 *is a nested neighbor of vertex* 2 *in the cluster graph* G_c*. In any feasible* G_c*-cut* $(V_1, \ldots, V_k)$ *of* $I = (G, w)$*, one can assume that* $|V_2| = 1$.

Proof. Let $(V_1, \ldots, V_k)$ be a G_c-cut $(V_1, \ldots, V_k)$ of I. Assume that $|V_2| > 1$ and let $x \in V_2$. Consider the G_c-cut $(V'_1, \ldots, V'_k)$ where $V'_i = V_i$ if $i \neq 1, 2$, $V'_1 = V_1 \cup (V_2 \setminus \{x\})$, $V'_2 = \{x\}$. The value of $(V'_1, \ldots, V'_k)$ is not larger than the value of $(V_1, \ldots, V_k)$ because w is non-negative, $(1, 2) \notin E_c$ and $N_{G_c}(1) \subseteq N_{G_c}(2)$.

Using Lemma 1, we deduce the following result for the leaves:

Corollary 1. *Assume that vertex* 1 *is a leaf of* G_c *where* $|V(G_c)| = k$*. In any feasible* G_c*-cut* $(V_1, \ldots, V_k)$ *of* $I = (G, w)$*, one can assume that* $|V_i| = 1$ *for all* $i \in N^2_{G_c}(1) \setminus \{1\}$.

Proof. If 1 is a leaf of G_c, then for every $i \in N^2_{G_c}(1) \setminus \{1\}$, vertex 1 is a nested neighbor of vertex i in G_c.

3 Complexity results of the minimum G_c-cut problem

In this section we show that the complexity of G_c-cut problem depends on the structure of the cluster graph G_c. We will use several reductions from the Biclique Vertex-Partition problem.

Definition 4. BICLIQUE VERTEX-PARTITION:
Instance: A graph G and positive integer k.
Question: Does G have a biclique vertex partition of size at most k consisting of mutually vertex-disjoint bicliques? (where the bicliques are (not necessarily vertex-induced) subgraphs of G).

For every fixed $k \geq 3$, Biclique vertex-partition is **NP**-complete, and remains **NP**-complete for bipartite graphs, see [5]. The case $k = 2$ has been open since a long time, but very recently, Biclique vertex-partition with $k = 2$ has been proved **NP**-complete, [17]. Because the case $k = 1$ is polynomial, the case $k = 2$ is equivalent to Biclique vertex-partition of size exactly 2.

The K_2-CUT PROBLEM is polynomial because it is exactly the minimum cut problem. Surprisingly, by replacing K_2 by $2K_2$ (two disjoint edges), the problem becomes much harder.

Theorem 2. *The* $2K_2$*-*CUT PROBLEM *is **NP**-hard.*

Proof. We propose a polynomial reduction from biclique vertex-partition. Let $G = (V, E)$ with $|V| = n$ and $k = 2$ be an instance of biclique vertex-partition. Consider the complete graph (G, w) defined as follows: $w(e) = 0$ if $e \in E$ and $w(e) = 1$ otherwise.

We claim that there exists a $2K_2$-cut of G with value 0 iff G admits a biclique vertex-partition of size exactly 2. Let $G_i = (A_i, B_i; E_i)$ with $i = 1, 2$ be a biclique vertex-partition of G. Clearly, $V_{2i-1} = A_i$, $V_{2i} = B_i$ for $i = 1, 2$ is a G_c-cut of G with value 0. Conversely, let $(V_i)_{i \leq 4}$ be a a $2K_2$-cut of G with value 0. Thus, for every $i \leq 2$, $G_i = (V_{2i-1}, V_{2i}; E_i)$ is a biclique of G and then, (G_1, G_2) is a biclique vertex-partition of G of size 2.

Corollary 2. *The* $2K_2$-CUT PROBLEM *is not approximable.*

Proof. In proof of Theorem 2, we have shown that for the $2K_2$-CUT PROBLEM, it is **NP**-complete to distinguish between $\text{opt} \leq 0$ and $\text{opt} > 0$, where opt is the value of an optimal G_c-cut. So, the result follows.

Now, we propose a way to extend the $2K_2$ case to larger cluster graphs and thus, preserving the hard cases via the notion of H-extension:

Definition 5. H-EXTENSION:
Let H and G be two graphs and $T \subseteq V(H)$. $G' = G + H$ is an H-extension of G with terminal T if (i) G' is connected (all edges between G and H) are incident in H to some vertices of $V(H) \setminus T$) and (ii) for every induced subgraph G_0 of G' isomorphic to H (given by the bijection f) such that $deg_{G_0}(f(v)) = deg_H(v)$ for $v \in T$, we get $G \subseteq G' - G_0$.

Roughly speaking, an H-extension G' of G with terminal T is such that $deg_H(v) = deg_{G'}(v)$ for any $v \in T$ and for any induced subgraph G_0 isomorphic to H (given by f) with the same restriction (i.e.,, $deg_{G_0}(v) = deg_{G'}(v)$ for any $v \in f(T)$), G is a subgraph of $G' - G_0$

For instance, Figure 1 gives an P_3-extension of $2K_2$ with $P_3 = (u_1, u_2, u_3)$ and terminal $T = \{u_2, u_3\}$. Actually, the only induced subgraphs of G' isomorphic to P_3 satisfying the condition of 5 are $G_1 = (v_1, v_2, u_1)$, $G_2 = (v_3, v_4, u_1)$, and $G_2 = (u_1, u_2, u_3)$. Finally, for every $i = 1, 2, 3$, we have: $2K_2 = G' - G_i$. The tree G' will be called the 3-star of length 2 and denoted by S_3^2 (more generally, the p-star of length 2 is given by $S_p^2 = \{(r, v_1^i), (v_1^i, v_2^i) : i = 1, \ldots, p\}$).

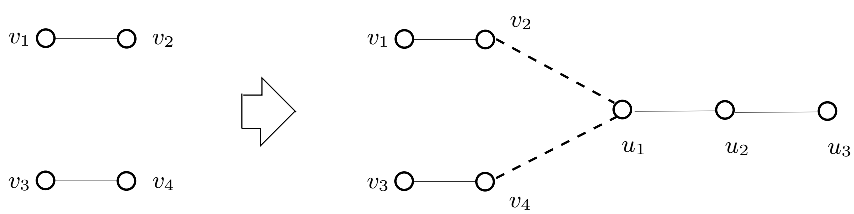

Fig. 1. Example of P_3-extension where $G = 2K_2$, $H = P_3 = (u_1, u_2, u_3)$, $G' = S_3^2$ and $T = \{u_2, u_3\}$.

Figures 2 and 3 give another P_4-extension of $2K_2$ or $3K_2$ with $P_4 = (u_1, u_2, u_3, u_4)$ and terminal $T = \{u_2, u_3\}$.

In Figure 3, for the Ψ-graph, the only induced P_4 satisfying the condition of 5 are $G_0 = (u_1, u_2, u_3, u_4)$ and $G_1 = (u_1, v_1, v_2, u_4)$ and we get Ψ-graph$-G_i = 3K_2$ for $i = 0, 1$ while for the κ-graph, the only induced P_4 satisfying the hypothesis are $G_0 = (u_1, u_2, u_3, u_4)$, $G_1 = (u_1, v_1, v_2, u_4)$, $G_2 = (v_2, v_1, u_1, u_2)$ and $G_3 = (u_3, u_2, u_1, v_1)$. Moreover, we have κ-graph$-G_i = 3K_2$ for $i = 0, 1$ and $3K_2 \subset \kappa$-graph$-G_i$ for $i = 2, 3$.

Now, we present some polynomial reductions preserving approximation from the G_c-CUT PROBLEM to itself depending on the structure of cluster graph G_c.

Theorem 3. *There exists a polynomial reduction preserving approximation from the* G_c-CUT PROBLEM *to the* G_c'-CUT PROBLEM *in the following cases:*

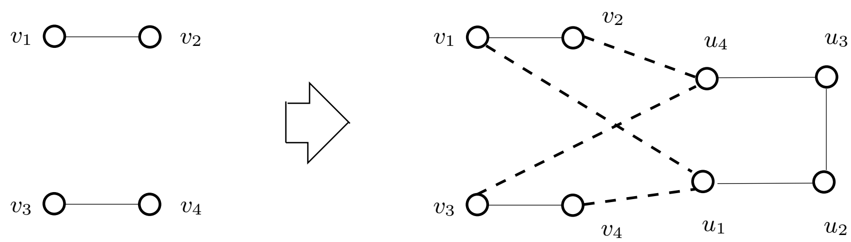

Fig. 2. Example of P_4-extension where $G = 2K_2$, $H = P_4 = (u_1, u_2, u_3, u_4)$ and $T = \{u_2, u_3\}$.

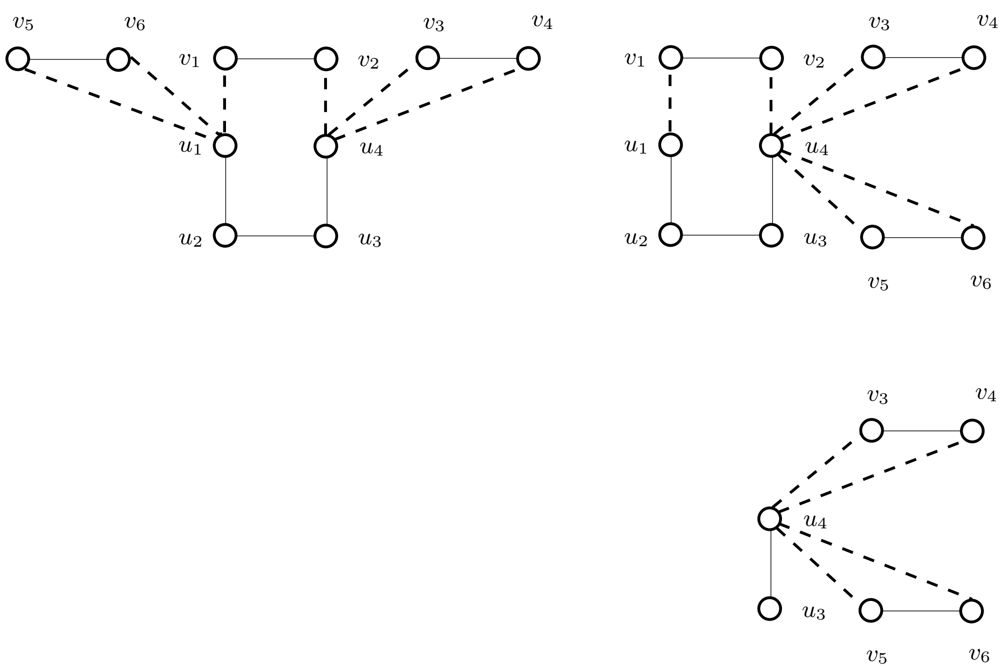

Fig. 3. Another example of P_4-extension where $G = 3K_2$, $H = P_4 = (u_1, u_2, u_3, u_4)$ and $T = \{u_2, u_3\}$. On the top, the Ψ-graph (left and top) and the κ-graph (right and top). On the bottom, the κ-graph minus $G_2 = (v_2, v_1, u_1, u_2)$. We get $3K_2 \subset \kappa$-graph$-G_2$.

(i) *Assume that the smallest connected component of the cluster graph G_c has $s \geq 2$ vertices. $G'_c = G_c + H$ where H is a connected graph of at least 2 vertices and at most s vertices, disconnected from G_c and if $|V(H)| = s$, then H is contained in every connected component of G_c with exactly s vertices.*

(ii) *$G'_c = G_c + P^0$ is an P_i-extension of G_c with $i \geq 3$, $P_i = (k+1, \ldots, k+i)$ and terminal $T = \{k+1, \ldots, k+i\}$.*

(iii) *$G'_c = G_c + P^0$ is an P_i-extension of G_c with $i \geq 4$, minimum degree 2, $P_i = (k+1, \ldots, k+i)$ and terminal $T = \{k+2, \ldots, k+i-1\}$.*

Proof. For (i). Let G_c and H be two graphs satisfying the condition at 5 and consider the cluster graph $G'_c = G_c + H$ where $V(H) = \{k+1, \ldots, k+p\}$, $2 \leq p \leq s$. Let (K_n, w) be an instance of the G_c-cut problem and consider the instance (K_{n+p}, w'), $V(K_{n+p}) \setminus V(K_n) = \{u_1, \ldots, u_p\}$ of the G'_c-cut problem defined as follows: if $u, v \in V(K_n)$, then $w'(u,v) = w(u,v)$. If $u \in V(K_n)$ and $v \notin V(K_n)$, $w'(u,v) = \infty$ [2] Finally, $w'(u_i, u_j) = 0$ if $(i,j) \in E(H)$ and $w'(u_i, u_j) = \infty$ otherwise.

Clearly, any G_c-cut of (K_n, w) can be converted into a G'_c-cut of (K_{n+p}, w') with same value by setting $V_{k+i} = \{u_i\}$. Conversely, consider any G'_c-cut $(V_1, \ldots, V_{k+p})$ of (K_{n+p}, w'). From the previous part, we can assume that this G'_c-cut has a finite value. Assume $u_1 \in V_{i_1}$. We get $V_{i_1} \cap V(K_n) = \emptyset$ because each connected component of G'_c has a size at least two and $V_{i_2} \subseteq \{u_1, \ldots, u_p\}$ for every $(i_1, i_2) \in E(G'_c)$. Hence, we deduce $V_j \subseteq \{u_1, \ldots, u_p\}$ if $V_j \cap \{u_1, \ldots, u_p\} \neq \emptyset$. Now, we must get $V_{i_j} = \{u_j\}$ for $j = 1, \ldots, p$ because each connected component G'_c has a size at least 2 and at most p. Hence, the subgraph G induced by $\{i_1, \ldots, i_p\}$ is a connected component of G'_c. If $p < s$ or G is isomorphic to H, then clearly, we must get $G = H$ and the restriction of this G'_c-cut to (K_n, w') is a G_c-cut of (K_n, w) with same value. Now, assume $p = s$ and $G \neq H$. Since, by assumption $E(H) \subseteq E(K_n)$, we get $E(K_n) \setminus E(H) \neq \emptyset$ and then the value of the G'_c-cut restricted to H has an infinite value, leading to contradiction. Hence, $G = H$ and the result follows.

For (ii). We first prove the case $i = 3$. Let (K_n, w) be an instance of the G_c-cut problem where $G_c = (V_c, E_c)$ is a graph with $|V_c| = k \geq 1$ vertices, and let $P^0 = (k+1, k+2, k+3)$. Now, let $G'_c = (V'_c, E'_c) = G_c + P^0$ be any P_3-extension of G_c with terminal $T = \{k+2, k+3\}$ (which means that the edges between the P^0 and G_c are only connected to endpoint $k+1$). Consider the following instance (K_{n+3}, w') of the G'_c-cut problem: $V(K_{n+3}) \setminus V(K_n) = \{u_1, u_2, u_3\}$ and $w'(u,v) = w(u,v)$ for $u, v \in V$, $w'(u_1, v) = 0$, $w'(u_2, v) = w'(u_3, v) = +\infty$ for $v \in V$, and $w'(u_1, u_2) = w'(u_3, u_2) = 0$, and $w'(u_1, u_3) = +\infty$.

Any G_c-cut of (K_n, w) can be converted into a G'_c-cut of (K_{n+3}, w') with same value by setting $V_{k+i} = \{u_i\}$ for $i = 1, 2, 3$. Conversely, assume that $(V_1, \ldots, V_{k+3})$ is a G'_c-cut of (K_{n+3}, w') with finite value. Assume that $u_2 \in V_{i_2}$ and $(i_3, i_2) \in E'_c$ (because G'_c is a connected graph with at least 4 vertices). We get $V_{i_3} \cap V(K_n) = \emptyset$, $V_{i_2} \cap \{u_1, u_3\} = \emptyset$ and $V_{i_3} \subseteq \{u_1, u_3\}$ because by construction $w'(u_2, v) = w'(u_3, v) = +\infty$ for $v \in V$ and $w'(u_1, u_3) = +\infty$. Hence, we deduce $V_{i_2} = \{u_2\}$ since $V_{i_2} \cap V(K_n) = \emptyset$.

If $V_{i_3} = \{u_1, u_3\}$, then vertex i_1 must be a leaf of G'_c and vertex i_2 has a neighbor $i_1 \neq i_3$ in G'_c (because G'_c is connected with at least 4 vertices). But $V_{i_1} \subseteq V(K_n)$ and $w'(u_3, v) = +\infty$ for $v \in V(K_n)$, contradiction. Now, since $w'(u_3, v) = w'(u_3, u_1) = +\infty$ for $v \in V(K_n)$ and G'_c is connected with at least 4 vertices we get $V_{i_3} = \{u_3\}$ and vertex i_3 is a leaf of G'_c. Because i_3 is a leaf of G'_c, then vertex i_2 must get exactly one neighbor $i_1 \neq i_3$ and $V_{i_1} = \{u_1\}$. So, $P = (i_3, i_2, i_1)$ is an induced P_3 of G'_c with terminal $\{i_2, i_3\}$. Since, G'_c is an P_3-extension of G_c, then the value of the G'_c-cut is minimum if $V_{k+i} = \{u_i\}$ for $i = 1, 2, 3$ (because $G'_c - P^0 = G_c$. Actually, if we flip the sets corresponding to P by the sets corresponding to P^0, the value of the G'_c-cut does not increase). Hence, the restriction of this G'_c-cut to (K_n, w') is a

[2] In the rest of the paper, we set $+\infty$ in order to simplify, but the sufficient value will be for instance $(n+1)w_{\max}$ where $w_{\max} = \max_{e \in E(G)} w(e)$.

G_c-cut of (K_n, w) with same value.

For the general case, let $P^0 = (k+1, \ldots, k+i)$ with $i \geq 3$. We replace (K_{n+3}, w') by (K_{n+i}, w') where:

– $V(K_{n+i}) \setminus V(K_n) = \{u_1, \ldots, u_i\}$ and $w'(u, v) = w(u, v)$.
– For $u, v \in V(K_n)$, $w'(u_1, v) = 0$, $w'(u_j, v) = +\infty$ for $v \in V$ and $j = 2, \ldots, i$.
– Finally, $w'(u_j, u_{j+1}) = 0$, for $j = 1, \ldots, i-1$ and $w'(u_j, u_{j'}) = +\infty$ otherwise.

The rest of the proof is completely similar to the previous one.

For (iii). We first prove the case $i = 4$. Let (K_n, w) be an instance of the G_c-cut problem where the cluster graph $G_c = (V_c, E_c)$ has $k \geq 1$ vertices and let $P^0 = (k+1, \ldots, k+4)$. Now, let $G'_c = (V'_c, E'_c) = G_c + P^0$ be any P_4-extension of G_c with terminal $T = \{k+2, k+3\}$ such that G'_c is without a leaf. Consider the following instance (K_{n+4}, w') of the G'_c-cut problem: $V(K_{n+4}) \setminus V(K_n) = \{u_1, \ldots, u_4\}$, and $w'(u, v) = w(u, v)$ for $u, v \in V(K_n)$. Moreover, $w'(u_j, v) = 0$ for $j = 1, 4$, and $w'(u_j, v) = +\infty$ for $j = 2, 3$. Finally, $w'(u_j, u_{j+1}) = 0$, for $j = 1, \ldots, 3$, and $w'(u_j, u_{j'}) = +\infty$ otherwise.

Any G_c-cut of (K_n, w) can be converted into a G'_c-cut of (K_{n+4}, w') with same value by setting $V_{k+i} = \{u_i\}$ for $i = 1, \ldots, 4$. Conversely, assume that $(V_1, \ldots, V_{k+4})$ is a G'_c-cut of (K_{n+4}, w') with finite value. Assume that $u_3 \in V_{i_3}$ and $(i_3, i_4), (i_3, i_2) \in E'_c$ (because G'_c has minimum degree 2). By construction, we get $V_{i_3} \cap V = \emptyset$ because otherwise $V_{i_j} \cap V = \emptyset$ for $j = 1, 3$ and $V_{i_j} \cap \{u_4, u_2\} \neq \emptyset$ for every $j = 2, 4$ (thus, this G'_c-cut will get an infinite value because either $u_1 \in V_{i_j}$ or $u_2 \in V_{i_j}$ for some $j = 2, 4$). Hence, we deduce $V_{i_j} \subseteq \{u_4, u_2\}$ for $j = 2, 4$ and then we can assume $V_{i_j} = \{u_j\}$ for $j = 2, 4$. Moreover, i_3 must have a degree 2 in G'_c and $(i_2, i_4) \notin E(G'_c)$. Now, because i_2 has a degree has at least 2 in G'_c, there is an edge $(i_1, i_2) \in E(G'_c)$ with $i_1 \notin \{i_3, i_4\}$. Thus, $V_{i_1} = \{u_1\}$, and on the one hand i_2 must have a degree 2 in G'_c, and on the other hand $(i_1, i_4) \notin E(G'_c)$. Hence $P = (i_1, \ldots, i_4)$ is an induced P_4 of G'_c with terminal $\{i_2, i_3\}$. Finally, since G'_c is a P_4-extension of G_c with terminal $\{k+2, k+3\}$, we can assume that $V_{k+i} = \{u_i\}$ for $i = 1, \ldots, 4$. In conclusion, the restriction of this G'_c-cut to (K_n, w') is a G_c-cut of (K_n, w) with same value.

For the general case, let $P^0 = (k+1, \ldots, k+i)$ with $i \geq 4$. We replace the instance (K_{n+4}, w') by (K_{n+i}, w') where:

– $V(K_{n+i}) \setminus V(K_n) = \{u_1, \ldots, u_i\}$.
– $w'(u, v) = w(u, v)$ for $u, v \in V(K_n)$.
– $w'(u_j, v) = 0$ for $j = 1, i$.
– $w'(u_j, v) = +\infty$ for $v \in V(K_n)$ and $j = 2, \ldots, i-1$.
– Finally, $w'(u_j, u_{j+1}) = 0$, for $j = 1, \ldots, i-1$ and $w'(u_j, u_{j'}) = +\infty$ otherwise.

The rest of the proof is completely similar to the previous one.

We saw at all the above constructions that the new i added vertices placed at the new i added clusters in an optimal solution and the original vertices must placed in an optimal way at the original clusters. Since the construction can perform in polynomial time the result are follow.

Corollary 3. *The G_c-CUT PROBLEM is **NP**-hard and not approximable in the following cases:*

(i) $G_c = pK_2$ *with* $p \geq 2$.
(ii) $G_c = S_p^2$ *with* $p \geq 3$.
(iii) $G_c = \Psi$*-graph or* $G_c = \kappa$*-graph.*

Proof. For (i). By applying part (i) of Theorem 3 with $G_c = 2K_2$ and $H = K_2$, we deduce from Theorem 2 that the $3K_2$-cut problem is **NP**-hard and not approximable. By induction on $p \geq 2$, with $G_c = pK_2$ and $H = K_2$ we deduce the claimed result.

For (ii). The $(p+1)$-star of length 2 S_{p+1}^2 (recall that S_p^2 is defined by $\{(r, v_1^i), (v_1^i, v_2^i) : i = 1, \dots, p\}$) is a P_3-extension of pK_2 and $P^0 = (2p+1, 2p+2, 2p+3)$ with terminal $T = \{2p+2, 2p+3\}$. Hence, using part (ii) of Theorem 3 and part (i) of Corollary 3, we get that the S_{p+1}^2-cut problem is **NP**-hard and not approximable for any $p \geq 2$.

For (iii). The Ψ-graph and the κ-graph are P_4-extensions of $3K_2$ and $P^0 = (7, 8, 9, 10)$ with terminal $T = \{8, 9\}$ and are without leaf. Hence, using part (iii) of Theorem 3 and Theorem 2, the result follows.

In part (i) of Theorem3, we have proved that the complexity of the G_c-CUT PROBLEM does not depend on the connectivity of the cluster graph G_c as long as, the size of each connected component is a at least 2. In Section 4, we will see that the G_c-CUT PROBLEM is polynomial time solvable if the cluster graph G_c has a fixed number of vertices and at least one isolated vertex. So now, we will assume that G_c is connected. In Corollary 3, all of the different connected graphs G_c such that the G_c-CUT PROBLEM is **NP**-hard have a maximum degree at least 3. Here, we prove the prove that this result remains true for the MULTIWAY G_c-CUT PROBLEM on a connected graphs G_c of maximum degree 2.

Theorem 4. *The* MULTIWAY P_k-CUT PROBLEM *is* ***NP****-hard and not approximable in the following cases:*

(i) $k = 5$ *or* $k \geq 8$*, even when only one vertex is specified (i.e.,* $|S| = 1$*).*
(ii) $k = 6$*, even when only two vertices are specified (i.e.,* $|S| = 2$*).*

Proof. We give a reduction preserving approximation from the $2K_2$-CUT PROBLEM proved **NP**-hard and not approximable in Theorem 2 and Corollary 2. Let $I = (K_n, w)$ be an instance of the $2K_2$-cut problem.

For (i) and $k = 5$, consider the instance $I' = (K_{n+1}, w')$ where $V(K_{n+1}) \setminus V(K_n) = \{x\}$, $w'(u, v) = w(u, v)$ if $u, v \neq x$, and $w'(u, x) = 0$ for $u \in V(K_n)$. Let $S = \{3\}$ with $x \in V_3$. Assume that $G_c = P_5 = (1, 2, 3, 4, 5)$.

Let (V_1, V_2, V_3, V_4) be any $2K_2$-cut of I. $(V_1', V_2', V_3', V_4', V_5')$ with $V_1' = V_1$, $V_2' = V_2$, $V_4' = V_3$, $V_5' = V_4$ and $V_3' = \{x\}$ is a P_5-cut of I' with same value. Conversely, let $(V_1', V_2', V_3', V_4', V_5')$ be any P_5-cut of I' such that $x \in V_3$. Using Corollary 1 with leaf 1 and $N_{P_5}^2(1) \setminus \{1\} = \{3\}$, we know that we can assume that $V_3' = \{x\}$. Hence, (V_1, V_2, V_3, V_4) where $V_1 = V_1'$, $V_2 = V_2'$, $V_3 = V_4'$, $V_4 = V_4'$ is a $2K_2$-cut of I with the same value.

For $k \geq 8$, using the P_i-extension of P_5 for $i \geq 3$ given in part 2 of Theorem 3 and the result given above for the MULTIWAY P_5-CUT PROBLEM, the result follows.

For (ii) and $k = 6$, consider the instance $I' = (K_{n+2}, w')$ where $V(K_{n+2}) \setminus V(K_n) = \{x, y\}$, $w'(u, v) = w(u, v)$ if $u, v \neq x$, $u, v \neq y$ and $w'(u, x) = w'(u, y) = 0$ for $u \in V$, Let $S = \{3, 4\}$ with $x \in V_3$ and $y \in V_4$. Assume that $G_c = P_6 = (1, 2, 3, 4, 5, 6)$. Since, vertices 1 and 6 are leaves of P_6 and $N_{P_6}^2(\{1, 6\}) \setminus \{1, 6\} = \{3, 4\}$, the same proof as previously gives the expected result.

In Section 4, we will see that the P_k-CUT PROBLEM is polynomial if $k \leq 4$ (note that these results also holds for the MULTIWAY P_k-CUT PROBLEM).

In conclusion of this section, we have obtained many cases where the C_k-CUT PROBLEM that the problem is **NP**-hard. In particular, when G_c is a tree that is quite surprising. In future research, we leave some open problems: what is the complexity of the C_k-CUT PROBLEM or the MULTIWAY C_k-CUT PROBLEM where $k \geq 5$? Nevertheless, these restrictions are **NP**-hard when the number of vertices is unbounded because the C_n-CUT PROBLEM on (K_n, w) (resp., P_n-cut) is clearly equivalent to solve the traveling salesman problem on (K_n, w), (resp., Hamiltonian path problem). Same question for the the P_k-CUT PROBLEM with $k \geq 5$ since as we will see in Section 4, the C_k-CUT PROBLEM for $k \leq 4$ and the P_k-CUT PROBLEM for $k \leq 4$ are polynomial time solvable.

4 Polynomially solvable cases

In this section, we will see some cluster graphs where the G_c-CUT PROBLEM is polynomial. It is the cases when the cluster graph G_c contains twins (stable set with same neighborhood), two nested neighbors (a stable set of two vertices with included neighborhood), leaves or isolated vertices (see Section 2 for formal definitions).

4.1 Complete r-partite graphs

Here, we mainly show that if the cluster graph G_c is a complete r-partite graph $K_{d_1,\ldots,d_r}$ (see Definition 2) where $k = \sum_{i=1}^{r} d_i$ is fixed, then the G_c-CUT PROBLEM can be solved in polynomial time, using an extension of the algorithm of [6]. Some simple complete r-partite graphs are the following: the stable graph $\bar{K_n}$ (ie., $E = \emptyset$) is complete 1-partite, and the complete graph is complete n-partite. We also look at the case where G_c is a restricted split graphs.

Let us begin by some properties of complete r-partite. As we will see, these graphs are recognizable within polynomial-time. The graph $H_3 = K_2 + K_1$ is the graph $G = (V, E)$ with $|V| = 3$ and $|E| = 1$ depicted in Figure 4.1.

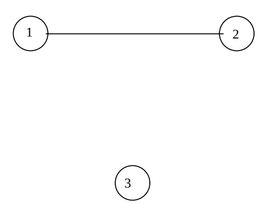

Fig. 4. An $H_3 = K_2 + K_1$ graph

Lemma 2. *$G = (V, E)$ is complete r-partite if and only if G is H_3-free.*

Proof. Suppose that G is complete r-partite. It is clear from the definition that $\forall u, v, w \in V$ we have three cases, none of which defines an H_3 graph:

1. u, v, w reside at different color classes, so the graph induced by them is a 3-clique.

2. u, v reside at the same color class, and w belongs to a different color class, so the graph induced by them contains the edges $(u, w), (v, w)$.
3. u, v, w reside at same color class so the graph induced by them contains no edge.

The opposite direction is done by induction on $n = |V(G)|$. Assume that any graph G with less than n vertices which does not contain H_3 as an induced subgraph is complete r-partite for some $r > 0$, and consider a graph $G = (V, E)$ with n vertices which does not contain H_3 as an induced subgraph. Let $v \in V$. $G' = G - v$ is also H_3-free, and then by inductive hypothesis is a complete r-partite graph. We study two cases.

1. $deg_G(v) = n - 1$. We add v in a new color class L_{r+1}. Obviously, G is complete $(r + 1)$-partite.
2. $deg_G(v) < n - 1$. So, there is $u \in L_i$ such that $(u, v) \notin E$. We add v in the color class L_i. Let us prove that G is complete r-partite. First, $L_i \cup \{v\}$ is a stable set because L_i is a stable set and G is H_3-free. Second, $\forall j \neq i$, $\forall u \in L_j$, $(u, v) \in E$. Otherwise, $\exists u \in L_j$ with $(u, v) \notin E$. Let $w \in L_i$. The graph induced by $\{u, v, w\}$ is isomorphic to H_3, a contradiction.

Using Lemma 2, it is clear that we can check in $O(|V(G)|^3)$ whether a graph is complete r-partite. Actually, a careful analysis of Lemma 2 gives a $O(|E(G)|)$ time algorithm to recognizable such graphs.

Theorem 5. *The $K_{d_1,\ldots,d_r}$-CUT PROBLEM can be solved in polynomial time if $k = \sum_{i=1}^{r} d_i$ is fixed and is **NP**-hard, if there exists d_i and d_j unbounded.*

Proof. Assume $k = \sum_{i=1}^{r} d_i$ fixed and $G_c = K_{d_1,\ldots,d_r}$. Since G_c is a complete r-partite graph, any two vertices $u, v \in L_i$ of the same layer are twins where we recall that a twin is a stable set of two vertices with same neighborhood. Hence, the $K_{d_1,\ldots,d_r}$-cut problem is equivalent to solve the d'-size restricted K_r-cut problem on $I = (G, w)$ where $d'_i = d_i$, for $i = 1, \ldots, r$ which is itself equivalent to computing a minimum r-cut $(V_1, \ldots, V_r)$ of G with the constrains $|V_i| \geq d_i$. Finally, this later problem can be computed in polynomial time using the same lines of the proof that those given in [6].

Now, we first prove that the case of complete bipartite graph is **NP**-complete if $k = d_1 + d_2$ is unbounded. The bisection graph problem consists of finding a cut of minimum size of a graph such that the two cut sets have the same size. Assume n even and consider $d_1 = d_2 = n/2$. The K_{d_1,d_2}-cut problem on (G, w) is equivalent to solve the bisection graph problem on $G = (V, E)$ (by setting $w(e) = 1$ if $e \in E$ and $w(e) = 0$ if $e \notin E$) which is known to be **NP**-hard [7]. Now, we reduce the K_{d_1,d_2} case to the K_{d_1,d_2,d_3} case. An inductive proof on r allows us to conclude the proof.

Given an instance $I = (K_n, w)$ of the K_{d_1,d_2}-cut problem, consider the following instance $I' = (K_{n+d_3}, w')$ of the K_{d_1,d_2,d_3}-cut problem: $V(K_{n+d_3}) \setminus V(K_n) = \{u_1, \ldots, u_{d_3}\}$ and if $u, v \in V(K_n)$, $w'(u, v) = w(u, v)$. If $u \in V(K_n)$ and $v \notin V(K_n)$, $w'(u, v) = 0$. Finally, $w'(u_i, u_j) = \infty$.

Consider a solution of the K_{d_1,d_2,d_3}-cut problem with color classes L_1, L_2, L_3. By construction, $\{u_1, \ldots, u_{d_3}\}$ belongs to the same color class, say L_i because G_c is complete 3-partite and $w'(u_i, u_j) = \infty$. We study two cases:

- $|L_i| \neq d_3$. Wlog., we can assume that d_3 sets of L_i are such that $V_i = \{u_i\}$ with $i = 1, \ldots, d_3$ because G_c is 3-partite. Let $|L_j| = d_3$. We flip the sets of L_i different to V_i in L_j (as new sets). We obtain a new solution of the K_{d_1,d_2,d_3}-cut problem.
- $|L_i| = d_3$. Wlog., we can assume that $\{u_i\} \subseteq V_i$ for $i = 1, \ldots, d_3$, because G_c is 3-partite. We move the vertices of $L_i \setminus \{u_1, \ldots, u_{d_3}\}$ to the color class L_j with $j \neq i$. Again, we obtain a new solution of the K_{d_1,d_2,d_3}-cut problem with $\leq$ value.

Now, let S be the vertices which have been flipped. For each of the two cases, the new solution loses $w(S, C_j)$ and wins $w(S, \{u_1, \ldots, u_{d_3}\}) = 0$. Hence, the new solution has a better cost and then we can assume the restriction of K_n is a K_{d_1,d_2}-cut with same value.

For instance, $K_4 - K_2$ the graph depicted in Figure 5 is a complete 3-partite $K_{2,1,1}$ and then, the $K_4 - K_2$-cut problem is solvable in polynomial time.

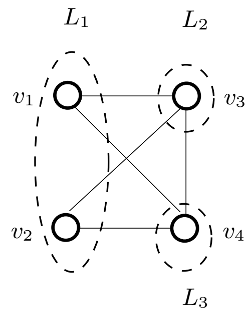

Fig. 5. A $K_4 - K_2$.

If only d_1 depends on the instance and r is fixed, then the complexity of the $K_{d_1,\ldots,d_r}$-cut problem is an open problem.

4.2 Restricted split graphs

Recall that a restricted split graph is a split graph where the degree of each vertex of the clique is at most $n-2$ (see Definition 2).

Theorem 6. *If $G_c = (S_c, K_c; E_c)$ is restricted split graph where $|K_c|$ is upper bounded by a constant, then the G_c-CUT PROBLEM can be solved in polynomial time.*

Proof. Let $G_c = (V_c, E_c)$ be a restricted split graph on $k \geq 3$ vertices and let $OPT = (V_1^*, \ldots, V_k^*)$ be an optimal G_c-cut of $I = (G, w)$. Then, for every $i \in K$, there exists $i' \in S$ such that i' is nested neighbor of vertex i in G_c because G_c is a restricted split graph. Using Lemma 1, we know that $|V_i^*| = 1$ for all $i \in K_c$. Hence, we can guess the $|K_c|$ vertices of $V_i^* = \{v_i^*\}$ for $i \in K_c$. After, consider the following complete bipartite graph $BP = (S_c, V \setminus \{v_i^* : i \in K_c\}; E(BP))$, edge weighted by d where $d(i, v) = \sum_{j \in N_{G_c}(i)} w(v, v_j^*))$, and find a b-matching M saturating S_c of minimum weight d (the algorithm is the same as finding a b-matching of maximum weight d' where $d'(e) = d_{\max} - d(e)$, $d_{\max} = \max_{e \in E(BP)} d(e))$ with $b^-(i) = 1$ and $b^+(i) = |V|$ for $i \in S_c$ and $b^-(v) = b^+(v) = 1$ for $v \in V \setminus \{v_i^* : i \in K_c\}$. Recall that a b-matching of a graph $G = (V, E)$ is a subset M such that if $G' = (V, M)$, then $\forall v \in V$, $b^-(v) \leq deg_{G'}(v) \leq b^+(v)$. A b-matching of maximum weight can be done in polynomial-time $O(|V(G)|^3)$, see [21] section 21 page 337. Since any G_c-cut corresponds to a b-matching M saturating S_c with value $d(M)$ (when v_i^* for $i \in K_c$ have been guessed), the previous algorithm finds an optimal solution in time $O(n^{|K_c|+3})$.

In particular, the P_4-cut problem on (G, w) can be solved in $O(n^5)$ time. However, for the P_4-cut problem on (G, w), we can improve the complexity to $O(n^3)$. Instead of applying a b-matching algorithm, we apply the following greedy algorithm: each vertex v of $V \setminus \{v_2^*, v_3^*\}$ (here $K_c = \{2,3\}$ and $S_c = \{1,4\}$) is assigned to the V_i^* with $i = 1,4$ minimizing its contribution (i.e., $i = \arg\min_{s \in S_c} \sum_{j \in N_{G_c}(s)} w(v, v_j^*)$). Careful attention must be taken to avoid to get $V_1^* = \emptyset$ or $V_4^* = \emptyset$. For instance, if $V_1^* = \emptyset$, then we find $v^* = \arg\min\{d(1,v) - d(4,v) : v \in V \setminus \{v_2^*, v_3^*\}\}$ and we add v^* to V_1^*. The time complexity of this algorithm is $O(n^3)$. Also note that in the same spirit of the proof of Theorem 6 the result holds if $S \cup N_{G_c}^2(S) = V(G_c)$ where S is the leaves of G_c.

Lemma 3. *If $S \cup N_{G_c}^2(S) = V(G_c)$, where S is leaves of G_c, then the G_c-CUT PROBLEM can be solved in polynomial time.*

4.3 When G_c contains isolated vertices

Definition 6. G is an H_0 *graph* if it contains at least one isolated vertex v, i.e., $\deg_G(v) = 0$. Let $S(G) = \{v \in V : \deg_G(v) = 0\}$.

Lemma 4. *Let G_c be an H_0 graph. The G_c-CUT PROBLEM can be solved in polynomial time iff $|V(G_c) \setminus S(G_c)|$ is fixed.*

Proof. Consider the unbounded case of $|V(G_c) \setminus S(G_c)|$. $G_c = C_n + K_1$ is an H_0 graph and the G_c-cut problem is clearly equivalent to solve the minimum traveling salesman problem. Now, let G_c be an H_0 graph such that $|V(G_c) \setminus S(G_c)| = k-1$ is fixed and consider an instance $I = (G, w)$ of the G_c-cut problem.

Denote the clusters corresponding to vertices of $S(G_c)$ as $V_k, \ldots, V_p$ and the $k-1$ clusters corresponding to $V(G_c) \setminus S(G_c)$ by $V_1, \ldots, V_{k-1}$. Enumerate the ordered subsets $H \subset V$, $|H| = k-1$. Insert the vertices from H into the clusters $V_1, \ldots, V_{k-1}$ so that each cluster contains exactly one vertex according to the order, and insert arbitrarily the vertices of $V \setminus H$ into the remaining clusters $V_k, \ldots, V_p$. Let S_H be this solution and S be the minimum one. Since $k-1$ is fixed the complexity is $O\left(\binom{n}{k-1} \cdot (k-1)!(k-1)^2\right) = O(n^{k-1})$. The obtained solution is optimal since any assignment of the vertices of V into the clusters $V(G_c) \setminus S(G_c)$ which results with one of these clusters having more than one vertex, can be improved by moving all these vertices except one from this cluster into any one cluster among $V_k, \ldots, V_p$. This new solution is an improved solution to the original, since the subset of edges in the new solution is strictly contained in the original solution.

Definition 7. A *complete $r - H_0$ graph* is a graph $G = (V, E)$ where $V(G) = \cup_{i=1}^r L_i$ such that (i) for every $i \in \{1, \ldots, r\}$ the graph induced by L_i is an H_0 graph, and (ii) for every $i, j \in \{1, \ldots, r\}$ with $i \neq j$, $(u, v) \in E$ for every $u \in L_i$, $v \in L_j$t.

In particular a complete $1 - H_0$ graph is a H_0 graph and $K_3 \pm K_2$[3] (see Figure 6) is a complete $2 - H_0$ graph (where $L_1 = \{1\}$ and $L_2 = \{2,3,4\}$).

The following result extends Lemma 4 to complete $2 - H_0$ graphs.

Theorem 7. *Suppose that G_c is a complete $2 - H_0$ cluster graph and $|V_c| = k$ is constant. Then the G_c-CUT PROBLEM can be solved in polynomial time.*

Proof. Denote $V_c = V_{c_1} \cup V_{c_2}$ such that the two induced graphs $G_c(V_{c_1}), G_c(V_{c_2})$ are H_0-graphs, $|V_{c_1}| = k_1$, and $|V_{c_2}| = k_2$, where $k_1 + k_2 = k$. Let $I = (G, w)$ be an instance and assume that optimal solution assigns the vertices $V_1^* \subset V$ to the clusters of $G_c(V_{c_1})$ and $V_2^* \subset V$ (with $V_1^* \cup V_2^* = V$) to the clusters of $G_c(V_{c_2})$.

[3] $\pm$ means that K_3 and K_2 share a common vertex.

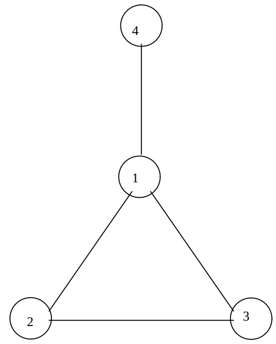

Fig. 6. A $K_3 \underline{+} K_2$.

The optimal solution on V_1^* must assign $k_1 - 1$ vertices to $k_1 - 1$ clusters and the rest of its vertices to the cluster represented by the isolated vertices. Similarly with V_2^*. So again we iterate over all ordered subsets $K_1, K_2 \subset V$ such that $|K_1| = k_1$, $|K_2| = k_2$, and compute a min-cut on G which separates K_1 from K_2. Denote the cost of the assignment of K_1 vertices to the clusters V_{c_1} according to the order as W_1. Denote the cost of the assignment of K_2 vertices to the clusters V_{c_2} according to the order as W_2. Denote the cost of $\min$-cut on G which separates K_1 from K_2 as W_3, and let $W = W_1 + W_2 + W_3$. Compute K_1, K_2 minimizing W. The clustering derived from them is the optimal solution and has a polynomial runtime complexity in $|V|$.

We now list several types of the cluster graph G_c for which the previous Theorems imply a polynomial algorithm.

(1.) If G_c has at most four vertices, then the G_c-CUT PROBLEM is polynomial iff $G_c \neq 2K_2$.
(2.) $V_c = \{0, , \ldots, k\}$, $E_c = \{(0,1), \ldots, (0,k)\} \cup \{(1,2), \ldots, (k-2, k-1)\}$. G_c is a complete $2 - H_0$ graph where $L_1 = \{0\}, L_2 = \{1, \ldots, k\}$ depicted in Figure 7 .

For $(1.)$, the only cases to study are the graphs which contain a P_4 as multiway subgraph, because the remaining cases are the H_0-graphs, $2K_2$, or the connected graphs on at most 3 vertices (and then isomorphic to K_1, K_2, K_3 or $P_3 = K_{1,2}$). Thus, the graphs which contain a P_4 are K_4, $C_4 = K_{2,2}$, $K_3 \pm K_2$ (see Figure 6), $K_4 - K_2 = K_{2,1,1}$ (see Figure 5), but all are polynomial as proved previously.

It is easy to see that complete r-partite graphs generalize the k-cut problem, because G_c is a k-clique and we can look at each vertex as a different color class. It is also clear that complete $r - H_0$ graphs generalize complete r-partite graphs, because each color class is an H_0-graph. We are still left with the open problem whether when G_c is a complete $r - H_0$ graph, the problem is polynomial or **NP**-hard for $r > 2$.

5 Approximation results

In this section, we give some approximation results for the G_c-CUT PROBLEM when the weights satisfy the triangle inequality or are positive.

The version of the k-cut problem on (K_n, w) with the additional requirement that cluster V_i must have a size of $d_i \in \mathbb{N}$, where $\sum_{i=1}^{k} d_i = n$, is studied in [18], and it is shown there that under the triangle inequality

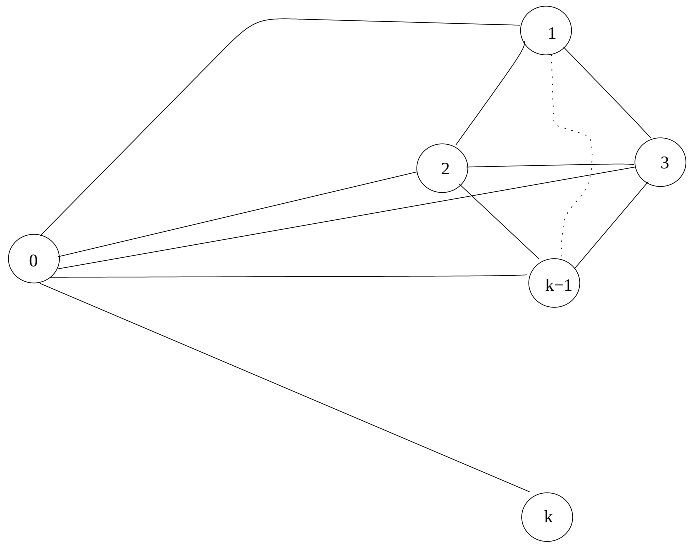

Fig. 7. A complete 2 H0 Example.

and fixed k it possible to obtain an approximation of at most three times the optimal value. We extend this result to the cluster graphs in two steps: first, we demonstrate the idea assuming that G_c is a ring (ie., an induced cycle C_k on k vertices). Second, we apply the same arguments to any cluster graph on k vertices.

We use an auxiliary problem, the MIN-ADJACENT-STAR PROBLEM, as explained in the next lines. For a given $G_c = (V_c, E_c)$ and a given set of centers $C = \{c_1, \ldots, c_k\} \subseteq V$ with $c_i \in V_i$, we wish to arrange the vertices of $V \setminus C$ into $|V_c| = k$ clusters. After the arrangement we will get $|V_i| = d_i$ and we want to minimize $\sum_{i=1}^{k}(\sum_{\{j|(i,j)\in E_c\}} \sum_{v\in V_j\setminus\{c_j\}} w(c_i, v))$. Thus, we want to arrange the clusters so that the arrangement yields the minimum sum of distances from the rest of the vertices to the given centers of the k stars according to the neighborhood relations between the stars.

5.1 The metric restricted cyclic k-cut problem

Let $K_n = (V, E)$ be a complete undirected graph with $|V| = n$. The edges $e \in E$ have nonnegative weights $w(e) \geq 0$ that satisfy the triangle inequality (i.e., $\forall x, y, z \in V$, $w(x, y) \leq w(x, y) + w(y, z)$). Given is also a set of integers $K = \{d_i\}_{i=1}^{k}$ such that $\sum_{i=1}^{k} d_i = n$.

Definition 8. For any $k \geq 3$, the METRIC RESTRICTED CYCLIC k-CUT PROBLEM computes, given an instance $I = (G, w)$ satisfying the triangle inequality and k integers d_i with $\sum_{i=1}^{k} d_i = n$, k disjoint subsets of vertices $V_i \subseteq V$ with size $|V_i| = d_i$ for $i \leq k$, minimizing the total weight of edges whose two ends are in the i and $i+1$ sets for $i = 1, \ldots, k$, where $k + 1 \equiv 1$.

Actually, the METRIC RESTRICTED CYCLIC k-CUT PROBLEM is the METRIC RESTRICTED C_k-CUT PROBLEM as indicated in Definition 3. The METRIC RESTRICTED CYCLIC 3-CUT PROBLEM is **NP**-hard because it is the 3-cut problem with the additional requirement, proved **NP**-hard in [18]. Here, we strengthen this result by proving that it is the case even if the weights are either one or two.

Theorem 8. *For any $k \geq 3$, the* METRIC RESTRICTED CYCLIC k-CUT PROBLEM *is **NP**-hard, even if $w(e) \in \{1, 2\}$.*

Proof. Let $k \geq 3$. We propose several polynomial reductions depending on the parameter k. These reductions are quite similar and are done from the bisection graph problem in complete graphs K_{2n} with weights in $\{1, 2\}$ which is know to be **NP**-hard. Recall that the bisection graph problem consists of finding a minimum cut of an unweighted graph such that the two cut sets have the same size. The bisection graph problem is **NP**-hard [7] and it is easy to see that the metric bisection graph problem on complete graphs restricted to weights 1 and 2 remains **NP**-hard. Let $I = (K_{2n} = (V, E), w)$ be a complete graph on $2n$ vertices and edges weighted by $w(e) \in \{1, 2\}$, instance of the metric bisection graph problem.

For $k = 3$. Consider the instance $I' = K_{2n+1}, w')$ with $d_1 = d_2 = n$ and $d_3 = 1$ of the METRIC METRIC RESTRICTED CYCLIC 3-CUT PROBLEM described as follows: $V(K_{2n+1}) = V \cup \{x_1\}$ and for any $u, v \in V$ $w'(u, v) = w(u, v)$, $w'(x_1, v) = 1$ for every $v \in V$.

We claim that there is a bisection of K_{2n} of value $w(V_1, V_2)$ at most B iff there is a cyclic 3-cut (with $d_1 = d_2 = n$ and $d_3 = 1$) of value at most $B + 2n$.

Clearly, if (V_1, V_2) is a bisection of K_{2n} of value at most $w(V_1, V_2) \leq B$, then $(V_1, V_2, V_3 = \{x_1\})$ is a cyclic 3-cut with value $w'(V_1, V_2, V_3) \leq B + 2n$. Conversely, let (V_1, V_2, V_3) be a cyclic 3-cut with value at most $w'(V_1, V_2, V_3) \leq B + 2n$ and such that $|V_1| = |V_2| = n$ and $|V_i| = 3$. Let us prove that we can polynomially transform it into a cyclic 3-cut (V'_1, V'_2, V'_3) with $w'(V_1, V_2, V_3) \leq w'(V'_1, V'_2, V'_3)$ and such that $V'_3 = \{x_1\}$. So, assume that $V'_3 = \{v\}$ with $v \in V$ and $x_1 \in V_1$. By setting $(V'_1 = V_1 \setminus \{x_1\} \cup \{v\}, V'_2 = V_2, V'_3 = \{x_1\})$, we get: $w'(V'_1, V'_2, V'_3) - w'(V_1, V_2, V_3) = 2n + w(v, V_2) - (n + 1 + w(v, V \setminus \{v\})) = n - 1 - w(v, V_1 \setminus \{v\}) \leq 0$. Hence, $(V'1, V'_2)$ is a bisection of K_{2n} of value $w(V'1, V'_2) = w'(V'_1, V'_2, V'_3) - 2n \leq -w'(V_1, V_2, V_3) - 2n \leq B$.

For $k = 4$. We assume that n is even; actually, it is easy to see that the bisection graph problem in complete graphs K_{4n} with weights in $\{1, 2\}$ remains **NP**-hard. Let $I = (K_{4n} = (V, E), w)$ be a complete graph on $4n$ vertices and edges weighted by $w(e) \in \{1, 2\}$, instance of this restriction. By setting $I' = (K_{4n}, w)$ and $d_i = n$ for every $i = 1, \dots, 4$, we can easily prove that $(V'_1, \dots, V'_4)$ is a restricted cyclic 4-cut of value $w(V'_1, \dots, V'_4) \leq B$ iff $(V_1 = V'_1 \cup V'_3, V_2 = V'_2 \cup V'_4)$ is a bisection of value $w(V_1, V_2) \leq B$.

For $k \geq 5$. Consider the instance $I' = K_{32n+(k-5)18n}, w')$ with $d_1 = d_2 = n$, $d_3 = d_k = 6n$ and $d_4 = \dots = d_{k-1} = 18n$ of the METRIC RESTRICTED CYCLIC k-CUT PROBLEM described as follows: $V(K_{32n+(k-5)18n}) = V \cup \{x_1, \dots, x_{30n+(k-5)18n}\}$ and for any $u, v \in V$ $w'(u, v) = w(u, v)$, $w'(x_i, v) = 2$ for every $i = 1, \dots, 30n + (k - 5)18n$ and $v \in V$, and finally, $w'(x_i, x_j) = 1$ for $1 \leq i < j \leq 30n + (k - 5)18n$.

We claim that there is a bisection of K_{2n} of value $w(V_1, V_2)$ at most B iff there is a cyclic k-cut (with $d_1 = d_2 = n$, $d_3 = d_k = 6n$ and $d_4 = \dots = d_{k-1} = 18n$) of value at most $B + 132n^2 + (k - 5)(18n)^2$.

Clearly, if (V_1, V_2) is a bisection of K_{2n} of value at most $w(V_1, V_2) \leq B$, then $(V_1, \dots, V_k)$ where $V_3 \cup \dots \cup V_5 = \{x_1, \dots, x_{30n+(k-5)18n}\}$ is a cyclic k-cut with value $w'(V_1, \dots, V_k) \leq B + 132n^2 + (k - 5)(18n)^2$. Conversely, let $(V_1, \dots, V_k)$ be a cyclic k-cut with value at most $w'(V_1, \dots, V_k) \leq B + 132n^2 + (k - 5)(18n)^2$ and such that $|V_1| = |V_2| = n$, $|V_3| = |V_k| = 6n$ and $|V_4| = \dots = |V_{k-1}| = 18n$. Let us prove that we can polynomially transform it into a cyclic k-cut $(V'_1, \dots, V'_k)$ with $w'(V'_1, \dots, V'_k) \leq w'(V_1, \dots, V_k)$ and such that $\cup_{i=3}^{k} V'_i = \{x_1, \dots, x_{30n+(k-5)18n}\}$.

We prove this claim in two steps using a 2-exchange procedure. First, we demonstrate that the result holds for $V_4 \cup \cdots \cup V_{k-1}$ and then we prove it for $V_3 \cup V_k$. Concerning the first step, we distinguish two cases: $k = 5$ and $k \geq 6$.

$k = 5$. So, assume that $v \in V_4 \cap V$. Then, there exists $x_i \in V_j$ with $j \in \{1, 2\}$. Consider the cyclic 5-cut $(V'_1, \ldots, V'_5)$ where from $(V_1, \ldots, V_5)$, we make a 2-exchange between v and x_i; so, $V'_j = (V_j \setminus \{x_i\}) \cup \{v\}$, $V'_4 = (V_4 \setminus \{v\}) \cup \{x_i\}$ and $V'_p = V_p$ for $p \neq j, 4$. Assume that w is the neighbor (distinct of $3 - j$) of j in G_c (so, $w = 5$ if $j = 1$ and $w = 3$ if $j = 2$). The contribution of v and x_i in the cyclic 5-cut $(V_1, \ldots, V_5)$ at least $w'(v, V_3 \cup V_5) + w'(x_i, V_{3-j} \cup V_w) \geq (2 \times 10n + 2n) + 7n = 29n$ (because on the one hand v is linked to at least $10n$ vertices of $\{x_1, \ldots, x_{30n}\}$ and at most $2n$ vertices of V and on the other hand, v is linked to $7n = |V_{3-j} \cup V_w|$ vertices) while the contribution of v and x_i in the cyclic 5-cut $(V'_1, \ldots, V'_5)$ is at most $w'(x_i, V_3 \cup V_5) + w'(v, V_{3-j} \cup V_w) \leq 10n + 4n + 2 \times 7n = 28n$. Hence, $w'(V'_1, \ldots, V'_5) - w'(V_1, \ldots, V_5) \leq 28n - 29n \leq 0$.

$k \geq 6$. First, assume that $v \in (V_4 \cup V_{k-1} \cap V$. By symmetry, suppose that $v \in V_4$. Then, there exists $x_i \in V_j$ with $j \in \{1, 2\}$. Consider the cyclic k-cut $(V'_1, \ldots, V'_k)$ where from $(V_1, \ldots, V_k)$, we make a 2-exchange between v and x_i. The contribution of v and x_i in the cyclic k-cut $(V_1, \ldots, V_k)$ is at least $w'(v, V_5 \cup V_3) \geq 2 \times 22n + 2n = 46n$ (because $|V_5 \cup V_3| = 24n$ and $|V| = 2n$; hence, v is linked to at least $22n$ vertices of $\{x_1, \ldots, x_{30n+(k-5)18n}\}$ and at most $2n$ vertices of V). On the other hand the contribution of v and x_i in the cyclic k-cut $(V'_1, \ldots, V'_k)$ is at most $w'(x_i, V_3 \cup V_5) + 2(|V_{2j-1}| + |V_w|) \leq 22n + 2 \times 2n + 2(n + 6n) = 40n$ where $w \in \{3, k\}$ is the neighbor of j different of $3 - j$ in G_c. In conclusion, $w'(V'_1, \ldots, V'_k) - w'(V_1, \ldots, V_k) \leq 40n - 46n \leq 0$.

Now assume $v \in V_p$ with $5 \leq p \leq k - 2$ (in this case, note that $k \geq 7$). The contribution of v and x_i in the cyclic k-cut $(V_1, \ldots, V_k)$ is at least $w'(v, V_{p-1} \cup V_{p+1}) \geq 2 \times 34n + 2n$ (because $|V_{p-1} \cup V_{p+1}| = 36n$ and $|V| = 2n$). On the other hand the contribution of v and x_i in the cyclic k-cut $(V'_1, \ldots, V'_k)$ is at most $w'(x_i, V_{p-1} \cup V_{p+1}) + 2(|V_j| + |V_w|) \leq 34n + 2 \times 2n + 2(n + 6n) = 52n$ (because x_i is linked to at least $34n$ vertices of $\{x_1, \ldots, x_{30n+(k-5)18n}\}$ and at most $2n$ vertices of V). In conclusion, $w'(V'_1, \ldots, V'_k) - w'(V_1, \ldots, V_k) \leq 34n - 52n \leq 0$.

In any cases ($k = 5$ or $k \geq 6$), by repeating this process, we get a cyclic k-cut $(V'_1, \ldots, V'_k)$ satisfying $V_4 \cup \cdots \cup V_{k-1} \subset \{x_1, \ldots, x_{30n+(k-5)18n}\}$ and $w'(V'_1, \ldots, V'_k) \leq w'(V_1, \ldots, V_k)$.

Now, assume that $v \in (V_3 \cup V_k) \cap V$ (by symmetry, suppose $v \in V_3$). Then, there exists $x_i \in V_j$ with $j \in \{1, 2\}$. As previously, consider the cyclic k-cut $(V'_1, \ldots, V'_k)$ resulting of a 2-exchange between v and x_i and let w be the neighbor different of $3 - j$ of j in G_c. The contribution of v and x_i in the cyclic k-cut $(V_1, \ldots, V_k)$ at least $w'(v, V_4) \geq 2 \times 18n = 36n$ (because from the previous case we know $V_4 \subseteq \{x_1, \ldots, x_{30n+(k-5)18n}\}$ while the contribution of v and x_i in the cyclic k-cut $(V'_1, \ldots, V'_k)$ is at most $w'(x_i, V_2 \cup V_4) + w'(v, V_{3-j} \cup V_w) \leq |V_4| + 2|V_2| + 2(|V_{3-j}| + |V_w|) = 18n + 2n + 2(n + 6n) = 34n$. Thus, $w'(V'_1, \ldots, V'_k) - w'(V_1, \ldots, V_k) \leq 34n - 36n \leq 0$.

In conclusion, from $(V_1, \ldots, V_k)$ we polynomially obtain a cyclic k-cut $(V'_1, \ldots, V'_k)$ such that $\cup_{i=3}^{k} V'_i = \{x_1, \ldots, x_{30n+(k-5)18n}\}$ and such that $w'(V'_1, \ldots, V'_k) \leq w'(V_1, \ldots, V_k)$. Hence, (V'_1, V'_2) is a bisection of K_{2n} with value $w(V'_1, V'_2) = w'(V'_1, \ldots, V'_k) - 132n^2 - (k-5)(18n)^2 \leq w'(V_1, \ldots, V_k) - 132n^2 - (k-5)(18n)^2 \leq B$.

Note that if k is unbounded, then the METRIC RESTRICTED CYCLIC k-CUT PROBLEM is **APX**-hard because this problem contains the METRIC TSP PROBLEM.

We demonstrate that for any fixed k it is possible to obtain in polynomial time an approximation of at most three times the optimal value. We start by defining a new problem which we solve optimally for a constant k, and then use its solution to approximate the METRIC RESTRICTED CYCLIC k-CUT PROBLEM.

Definition 9. The MIN-ADJACENT-STAR PROBLEM finds vertices $v_1, \ldots, v_k$ and a k-cut, such that $v_i \in V_i$, $|V_i| = d_i$, $i = 1, \ldots, k$, and

$$\sum_{i=1}^{p} \Big(d_i w(v_i, V_{i+1}) + d_{i+1} w(v_{i+1}, V_i) + d_i d_{i+1} w(v_i, v_{i+1}) \Big)$$

is minimized, where indices are (mod k).

Theorem 9. *Algorithm FindCyclicPartition (see Algorithm 1) solves the* MIN-ADJACENT-STAR PROBLEM. *It can be executed in time $O(n^{k+1})$.*

input :
1. A complete graph $K_n = (V, E)$, $|V| = n$, with weights $w(e) \geq 0$, $e \in E$.
2. Integers $d_1 \ldots, d_k$ such that $\sum_{i=1}^{k} d_i = n$.

output:
1. $v_1, \ldots, v_k \subseteq V$.
2. A partition $V_1, \ldots, V_k$ of V such that $v_i \in V_i$, $|V_i| = d_i$, $i = 1, \ldots, k$.

foreach *subset* $\{v_1, \ldots, v_k\} \subseteq V$ **do**
  $\{a_1, \ldots, a_{n-k}\} := V \setminus \{v_1, \ldots, v_k\}$.
  Compute $\tilde{x}$, an optimal solution to the following transportation problem:
  minimize $\sum_{i=1}^{k} \sum_{j=1}^{n-k} \sum_{l \in \{-1,1\}} d_i w(v_i, a_j) x_{i+l,j}$
  subject to:
  $\sum_{i=1}^{k} x_{ij} = 1, \quad j = 1, \ldots, n-k,$
  $\sum_{i=1}^{n-k} x_{ij} = d_i - 1, \quad i = 1, \ldots, k,$
  $x_{ij} \in \{0, 1\}, \quad i = 1, \ldots, k, \quad j = 1, \ldots, n-k.$
  $V_i^{\{v_1,\ldots,v_k\}} := \{v_i\} \bigcup \{a_j | 1 \leq j \leq n-k, \tilde{x}_{ij} = 1\}, \quad i = 1, \ldots, k.$
  $d^{\{v_1,\ldots,v_k\}} := \sum_{i=1}^{p} d_i w(v_i, V_{i+1}^{\{v_1,\ldots,v_k\}}) + k_{i+1} w(v_{i+1}, V_i^{\{v_1,\ldots,v_k\}}) + d_i d_{i+1} w(v_i, v_{i+1})$

end

Find $\{v_1^*, \ldots, v_k^*\} \subseteq V$ for which $d^{\{v_1^*,\ldots,v_k^*\}}$ is minimal, denote it by S^*.

return $(v_1^*, \ldots, v_k^*, V_1^{\{v_1^*,\ldots,v_k^*\}}, \ldots, V_k^{\{v_1^*,\ldots,v_k^*\}})$.

Algorithm 1: FindCyclicPartition

Proof. Let $\tilde{v}_1, \ldots, \tilde{v}_k, \tilde{V}_1, \ldots, \tilde{V}_k$ be an optimal solution to the min-adjacent-star problem. Since the algorithm checks all the subsets of V of size k it also checks the subset $\{\tilde{v}_1, \ldots, \tilde{v}_k\}$. For this subset the sum $\sum_1^p d_i d_{i+1} w(\tilde{v}_i, \tilde{v}_{i+1})$ is constant, so we need to find a partition $(V_1, \ldots, V_k)$ which minimizes $\sum_1^k [d_i w(\tilde{v}_i, V_{i+1}) + d_{i+1} w(\tilde{v_{i+1}}, V_i)]$. This is achieved by finding an optimal solution to a transportation problem (where $x_{ij} = 1$ if vertex a_j is assigned to the subset V_i). For a fixed value of k we can solve the transportation problem in time $O(n)$, using the algorithms of [27]. There are $O(n^k)$ subsets $\{v_1, \ldots, v_k\} \subseteq V$, so altogether the time complexity is $O(n^{k+1})$.

We now show that the weight of the partition found as an optimal solution for the MIN-ADJACENT-STAR PROBLEM is no more than $3\mathrm{opt}$, where opt is the value of the optimal solution of the METRIC CYCLIC k-CUT PROBLEM. Denote by apx the value of the partition constructed by Algorithm 1.

Theorem 10. *Algorithm FindCyclicPartition is a* 3*-approximation for the* METRIC RESTRICTED CYCLIC k-CUT PROBLEM *when* k *is constant.*

Proof. Let $(v_1, \ldots, v_k, V_1, \ldots, V_k)$ be the output of Algorithm FindCyclicPartition and let $O_1, \ldots, O_k$ be an optimal solution of the METRIC RESTRICTED CYCLIC k-CUT PROBLEM; obviously, $\forall i \leq k$, $|O_i| = d_i$ by hypothesis. We will prove that

$$\mathrm{apx} = \sum_{i=1}^{k} w(V_i, V_{i+1}) \leq 3 \sum_{i=1}^{k} w(O_i, O_{i+1}) = 3\mathrm{opt}.$$

By construction, we have:

$$\begin{aligned}
\mathrm{apx} &= \sum_{i=1}^{k} w(V_i, V_{i+1}) \\
&= \sum_{i=1}^{k} \sum_{\substack{u_i \in V_i \\ u_j \in V_{i+1}}} w(u_i, u_j) \\
&\leq \sum_{i=1}^{k} \sum_{\substack{u_i \in V_i \\ u_j \in V_{i+1}}} \Big(w(v_i, v_{i+1}) + w(u_i, v_{i+1}) + w(v_i, u_{i+1}) \Big) \\
&= \sum_{i=1}^{k} \Big(d_i w(v_i, V_{i+1}) + d_{i+1} w(v_{i+1}, V_i) + d_i d_{i+1} w(v_i, v_{i+1}) \Big) \\
&\equiv S^*.
\end{aligned}$$

On the other hand, according to Theorem 9, FindCyclicPartition solves the MIN-ADJACENT-STAR PROBLEM, so that for every $(u_1, \ldots, u_k)$ such that $u_1 \in O_1, \ldots, u_k \in O_k$,

$$\begin{aligned}
S^* &= \sum_{i=1}^{k} \Big(d_i w(v_i, V_{i+1}) + d_{i+1} w(v_{i+1}, V_i) + d_i d_{i+1} w(v_i, v_{i+1}) \Big) \\
&\leq \sum_{i=1}^{k} \Big(d_i w(u_i, O_{i+1}) + d_{i+1} w(u_{i+1}, V_i) + d_i d_{i+1} w(u_i, u_{i+1}) \Big).
\end{aligned}$$

Summing over all $(u_1, \ldots, u_k)$ such that $u_1 \in O_1, \ldots, u_k \in O_k$, since we have $\prod_{j=1}^{k} d_j$ equalities as above leaving the left side of each inequality as is meaning S^* we have that:

$$S^* \prod_{j=1}^{k} d_j \leq \sum_{i=1}^{k} \Big((\prod_{j=1}^{k} d_i) w(O_i, O_{i+1}) + (\prod_{j=1}^{k} d_i) w(O_{i+1}, O_i) + (\prod_{j=1}^{k} d_i) w(O_i, O_{i+1}) \Big)$$

$$= (\prod_{j=1}^{k} d_i) \sum_{i=1}^{k} \Big(w(O_i, O_{i+1}) + w(O_{i+1}, O_i) + w(O_i, O_{i+1}) \Big)$$
$$= (\prod_{j=1}^{k} d_i) 3 \sum_{i=1}^{k} w(O_i, O_{i+1})$$
$$= 3(\prod_{i=1}^{k} d_i) opt$$

Hence $S^* \leq 3\text{opt}$, giving $\text{apx} \leq S^* \leq 3\text{opt}$.

5.2 Approximation algorithms for the METRIC RESTRICTED G_c-CUT PROBLEM when k is constant

At subsection 5.1, we have proposed an approximation algorithm for the METRIC RESTRICTED CYCLIC k-CUT PROBLEM. Now, we will solve the general case of the METRIC RESTRICTED G_c-CUT PROBLEM when G_c is an arbitrary cluster graph with a constant number of vertices.

Definition 10. The MIN-ADJACENT-G_c PROBLEM finds vertices $v_1, \dots, v_k$ and a G_c-cut, such that $v_i \in V_i$, $|V_i| = d_i$, $i = 1, \dots, k$, and

$$\sum_{(i,j) \in E_c} \Big(d_i w(v_i, V_j) + d_j w(v_j, V_i) + d_i d_j w(v_i, v_j) \Big)$$

is minimized.

The idea of the algorithm is similar to the previous one and is described below.

Theorem 11. *Algorithm FindGcPartition (see Algorithm 2) solves the* MIN-ADJACENT-G_c PROBLEM *in time $O(n^{k+1})$.*

Proof. Let $\tilde{v}_1, \dots, \tilde{v}_k, \tilde{V}_1, \dots, \tilde{V}_k$ be an optimal solution to the MIN-ADJACENT-G_c PROBLEM. Since the algorithm checks all the subsets of V of size k, it also checks the subset $\{\tilde{v}_1, \dots, \tilde{v}_k\}$. For this subset the $\sum_{(i,j) \in E_c} d_i d_j w(\tilde{v}_i, \tilde{v}_j)$ is constant, so we need to find a partition $(V_1, \dots, V_k)$ which minimizes $\sum_{(i,j) \in E_c} d_i w(\tilde{v}_i, V_j)$. This is achieved by solving the transportation problem (where $x_{ij} = 1$ if and only if a_j assigned to the subset V_i).

For a fixed value of k we can solve the transportation problem in linear time in n, using the algorithms in [27]. There are $O(n^k)$ subsets $(v_1, \dots, v_k)$, so altogether the time complexity is $O(n^{k+1})$.

Theorem 12. *Algorithm FindCyclicPartition is a 3-approximation for the* METRIC RESTRICTED G_c-CUT PROBLEM *when k is constant.*

Proof. Let $(v_1, \dots, v_k, V_1, \dots, V_k)$ be the output of Algorithm FindCyclicPartition and let $O_1, \dots, O_k$ be an optimal solution of the METRIC RESTRICTED G_c-CUT PROBLEM; obviously, $\forall i \leq k$, $|O_i| = d_i$ by hypothesis. Let $(v_1, \dots, v_k, V_1, \dots, V_k)$ be the outputted solution and let $O_1, \dots, O_k$ be an optimal solution of the METRIC RESTRICTED G_c-CUT PROBLEM. Assume that $\forall i \leq k$, $|O_i| = d_i^*$ and consider the step of Algorithm 2 where $(d_1^*, \dots, d_k^*)$ is given in input. We have: Let $(v_1, \dots, v_k, V_1, \dots, V_k)$ be the outputted solution and let $O_1, \dots, O_k$ be an optimal solution of the METRIC G_c-CUT PROBLEM. Assume that $\forall i \leq k$, $|O_i| = d_i^*$ and consider the step of Algorithm 2 where $(d_1^*, \dots, d_k^*)$ is given in input. We have:

input :
1. A complete graph $K_n = (V, E)$, $|V| = n$, with weights $w(e)$ $e \in E$.
2. A cluster graph $G_c(V_c, E_c)$, $|V_c| = k$.
3. Integers $d_1, \ldots, d_k$ such that $\sum_{i=1}^{k} d_i = n$.
output:
1. $v_1, \ldots, v_k \subseteq V$.
2. A G_c-cut $V_1, \ldots, V_k$ such that $v_i \in V_i$, $i = 1, \ldots, k$.
For $i, j \in V_c$ let $\alpha_{ij} = \begin{cases} 1 & (i,j) \in E_c \\ 0 & \text{otherwise} \end{cases}$
foreach $\{v_1, \ldots, v_k\} \subset V$ **do**
    $\{a_1, \ldots, a_{n-k}\} := V \setminus \{v_1, \ldots, v_k\}$.
    Compute $\tilde{x}$, an optimal solution to the following transportation problem:
    minimize $\sum_{i=1}^{k} \sum_{j=1}^{k} \sum_{l=1}^{n-k} \alpha_{ij} d_i w(v_i, a_l) x_{lj}$
    subject to:
    $\sum_{i=1}^{k} x_{ij} = 1,\ j = 1, \ldots, n-k,$
    $\sum_{i=1}^{n-k} x_{ij} = k_i - 1,\ i = 1, \ldots, k,$
    $x_{ij} \in \{0, 1\},\ i = 1, \ldots, k,\ j = 1, \ldots, n-k.$
end
Let $V_i^{\{v_1,\ldots,v_k\}} := \{v_i\} \cup \{a_j | 1 \le j \le n-k,\ \ \tilde{x}_{ij} = 1\}$, $i = 1, \ldots, k$.
$d^{\{v_1,\ldots,v_k\}} := \sum_{(i,j)\in E_c} \left[d_i w(v_i, V_j^{\{v_1,\ldots,v_k\}}) + d_i d_j w(v_i, v_j) \right]$.
$\{v_1^*, \ldots, v_k^*\} := \arg\min\{d^{\{v_1,\ldots,v_k\}} \, | \{v_1^*, \ldots, v_k^*\} \in V$.
return $(v_1^*, \ldots, v_k^*, V_1^{\{v_1^*,\ldots,v_k^*\}}, \ldots, V_k^{\{v_1^*,\ldots,v_k^*\}})$.

Algorithm 2: FindGcPartition

$$
\begin{aligned}
\mathrm{apx} &\leq \sum_{(i,j)\in E_c} w(V_i, V_j) \\
&= \sum_{(i,j)\in E_c} \sum_{\substack{u_i \in V_i \\ u_j \in V_j}} w(u_i, u_j) \\
&\leq \sum_{(i,j)\in E_c} \sum_{\substack{u_i \in V_i \\ u_j \in V_j}} \Big(w(v_i, v_j) + w(u_i, v_j) + w(v_i, u_j) \Big) \\
&= \sum_{(i,j)\in E_c} \Big(d_i^* w(v_i, V_j) + d_j^* w(v_j, V_i) + d_i^* d_j^* w(v_i, v_j) \Big) = S^*.
\end{aligned}
$$

On the other hand, according to Theorem 11, FindGcPartition solves the MIN-ADJACENT-G_c PROBLEM so that for every $(u_1, \ldots, u_k)$ such that $u_1 \in O_1, \ldots, u_k \in O_k$,

$$
\begin{aligned}
S^* &= \sum_{(i,j)\in E_c} \Big(d_i^* w(v_i, V_j) + d_j^* w(v_j, V_i) + d_i^* d_j^* w(v_i, v_j) \Big) \\
&\leq \sum_{(i,j)\in E_c} \Big(d_i^* w(u_i, O_j) + d_j^* w(u_j, O_i) + d_i^* d_j^* w(u_i, u_j) \Big).
\end{aligned}
$$

Summing over all $(u_1, \ldots, u_k)$ such that $u_1 \in O_1, \ldots, u_k \in O_k$:

$$
\begin{aligned}
S^* \prod_{i=1}^{k} d_i^* &\leq \sum_{(i,j)\in E_c} \left[(\prod_{i=1}^{k} d_i^*) w(O_i, O_j) + (\prod_{i=1}^{k} d_i^*) w(O_j, O_i) + (\prod_{i=1}^{k} d_i^*) w(O_i, O_j) \right] \\
&= \prod_{i=1}^{k} d_i^* \sum_{(i,j)\in E_c} [w(O_i, O_j) + w(O_j, O_i) + w(O_i, O_j)] \\
&= 3 \prod_{i=1}^{k} d_i^* \sum_{(i,j)\in E_c} w(O_i, O_j) \\
&= 3 (\prod_{i=1}^{k} d_i^*) \mathrm{opt}
\end{aligned}
$$

Hence, $S^* \leq 3\mathrm{opt}$, leading to the conclusion that $\mathrm{apx} \leq S^* \leq 3\mathrm{opt}$.

5.3 Approximation algorithms for the metric G_c-cut problem

Here, we will solve the case where w is metric, $G_c = (V_c, E_c)$ is a general graph but $|V_c|$ is constant and without constraint on cluster sizes. Let $G = (V, E)$ be a complete undirected graph, with $V = \{v_1 \ldots v_n\}$, and edge weights $w(v_i, v_j) \geq 0$ that satisfy the triangle inequality.

Theorem 13. *There is a 3-approximation for the* METRIC G_c-CUT PROBLEM *when k is constant.*

Proof. Let $|V| = n, |V_c| = k$. Enumerate all $D = (d_1, \ldots, d_k)$, $\sum_{i=1}^{k} d_i = n$ we have an $O(n^k)$ such ordered sets. For each such ordered set solve the METRIC RESTRICTED G_c-CUT PROBLEM by using the algorithm from 5.2 algorithm 2, and get a 3 approximation as in 5.2. Choose the smallest G_c-cut among all the G_c-cuts. Since the optimal solution yields a specific D, the result follows.

5.4 Approximation of the G_c-cut problem with positive weights

In Section 3, we saw that the G_c-CUT PROBLEM is not approximable at all in general graphs (see for instance Corollaries 2 or 3) due to the weight 0 for some edges. Here, we propose an approximation ratio for the G_c-CUT PROBLEM when $w(e) > 0$ for every $e \in E$ which works even when the number k of the vertices of G_c depends on the instance (we only assume $k \leq n$). Let $w_{\min} = \min_{e \in E} w(e)$, $w_{\max} = \max_{e \in E} w(e)$, and $\alpha = \frac{w_{\max}}{w_{\min}}$.

Theorem 14. *The G_c-CUT PROBLEM with positive weights is α-approximable in linear time, where $\alpha = \frac{w_{\max}}{w_{\min}}$, even if $|V_c|$ is not fixed.*

Proof. Let $I = (K_n, w)$ with $w(e) > 0$ for every $e \in E$ be an instance of the G_c-cut problem. Let $m'_{i_0} = n - k + 1$ where $i_0 = \arg\min_{i \in V_c} \deg_{G_c}(i)$, and $m'_j = 1$ for $j \in \{1, \dots, k\}$, $j \neq i_0$. Arbitrarily assign m'_i vertices of K_n to V_i for $i = 1, \dots, k$ and let $(V_1, \dots, V_k)$ be the resulting G_c-cut with value apx. Let opt be the value of an optimal solution $(V_1^*, \dots, V_k^*)$ of the G_c-CUT PROBLEM on (K_n, w). We will prove that $\mathrm{apx} \leq \alpha \cdot \mathrm{opt}$. Let $\mathrm{opt}_u = \sum_{(i,j) \in E_c} m'_i m'_j$.

By construction, since $w(e) \leq w_{\max}$ we get:

$$\mathrm{apx} \leq w_{\max} \cdot \mathrm{opt}_u \tag{1}$$

Let $m_i = |V_i^*|$ for $i = 1, \dots, k$. We mainly prove that:

$$\mathrm{opt}_u \leq \sum_{(i,j) \in E_c} m_i m_j \tag{2}$$

To see this, we will show that opt_u is the value of an optimal G_c-cut on (K_n, w') where $w'(e) = 1$ for every $e \in E$. Hence, since $\sum_{(i,j) \in E_c} m_i m_j$ is the value of a particular G_c-cut on (K_n, w'), inequality (2) will follows.

Property 1. Let $(V_1^*, \dots, V_k^*)$ be an optimal G_c-cut on (K_n, w') where $m_i^* = |V_i^*|$ for $i = 1, \dots, k$. The following properties hold:

(i) If $e = (i,j) \in E_c$, then $\min(m_i^*, m_j^*) = 1$.
(ii) If $e = (i,j) \notin E_c$, then one can assume that $\min(m_i^*, m_j^*) = 1$.

Proof. For (i). Let $e = (i,j) \in E_c$ and $m = m_i^* + m_j^*$; denote $D_i = \sum_{r \in N_{G_c}(i) \setminus \{j\}} m_r^*$ and $D_j = \sum_{r \in N_{G_c}(j) \setminus \{i\}} m_r^*$. Assume $D_i \geq D_j$. Then, the contribution of the portion of the optimal solution where one cluster is either V_i^* or V_j^* can be written as $D_i \cdot m_i^* + m_j^* \cdot D_j + m_i^* \cdot m_j^*$ (because $\forall e \in E$, $w'(e) = 1$), or equivalently (using $m_j^* = m - m_i^*$), $f(m_i) = -(m_i^*)^2 + m_i^* \cdot (m + D_i - D_j) + m \cdot D_j$. This expression is a decreasing parabola and it reaches its minimum value for $m_i^* = 1$ or $m_i^* = m - 1$ because D_i, D_j and m are constant (actually, when m_i^* decreases by one unit, m_j^* increases by one unit).

For (ii). Let $(i,j) \notin E_c$ and $m = m_i^* + m_j^*$ and suppose $m_i^* > 1$, $m_j^* > 1$. Assume $\deg_{G_c}(i) \leq \deg_{G_c}(j)$ where we recall that $\deg_{G_c}(i)$ is the degree of vertex i in G_c. The contribution of clusters V_i^*, V_j^* in the optimal solution is $\deg_{G_c}(i) \cdot m_i^* + \deg_{G_c}(j) \cdot m_j^*$ because by (i) we know that $m_w^* = 1$ for $w \in N_{G_c}(i) \cup N_{G_c}(j)$. Substituting $m_j^* = m - m_i^*$ and rearranging the above expression we obtain $(\deg_{G_c}(i) - \deg_{G_c}(j)) \cdot m_i^* + \deg_{G_c}(j) \cdot m$. This expression is strictly decreasing with m_i^* as its argument when $\deg_{G_c}(i) < \deg_{G_c}(j)$, and constant when $\deg_{G_c}(i) = \deg_{G_c}(j)$. In conclusion, we can always assume that $m_j^* = 1$.

Using Property 1, we deduce that $m^*_{i_0} = n - k + 1$ and $m^*_j = 1$ for $j \in \{1, \ldots, k\}$, $j \neq i_0$, that is exactly $m'_i = m^*_i$ for $i \in \{1, \ldots, k\}$. Now, combining inequalities (1) and (2), we obtain:

$$\text{apx} \leq w_{\max} \cdot \sum_{(i,j) \in E_c} m_i m_j. \tag{3}$$

On the other hand, by construction we have:

$$\text{opt} \geq w_{\min} \cdot \sum_{(i,j) \in E_c} m_i m_j. \tag{4}$$

Using inequalities (3) and (4), the result follows.

6 Conclusion

In this paper, we have studied the complexity and the approximation of the G_c CUT PROBLEM. Some results are given, but many open problems exist. What is the exact complexity of the G_c CUT PROBLEM on lines or rings (ie., induced paths or cycles)? Is the METRIC Gc-CUT PROBLEM admit a PTAS or is **APX**-complete? Another interesting direction for further research is to study the maximum G_c cut problem.

References

1. C. F. Ball, P.V. Kraus, and D. A. Mlynski, “Fuzzy partitioning applied to VLSI-partitioning and placement,” *IEEE International Symposium on Circuits and Systems, ISCAS’94* **1** (1994) 177-180.
2. D. Bein, W. Bein, Z. Meng, L. Morales, C.O. Shields, Jr., and I.H. Sudborough, “Clustering and the biclique partition problem,” *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)* (2008).
3. M. Burlet, and O. Goldschmidt, “A new and improved algorithm for the 3-cut problem,” *Operations Res. Lett.* **21** (1997) 225-227.
4. E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis, “The Complexity of Multiterminal Cuts,” *SIAM J. Comput.* **23** (1994) 864-894.
5. H. Fleischner, E. Mujuni, D. Paulusma and S. Szeider, “Covering graphs with few complete bipartite subgraphs,” *Theoretical Computer Science* **410** (2009) 2045-2053.
6. O. Goldschmidt and D. S. Hochbaum, “A polynomial algorithm for the k-cut problem for fixed k,” *Mathematics of Operations Research* **19** (1994) 24-37.
7. M. R. Garey and D. S. Johnson “Computers and Intractability: A Guide to the Theory of NP-Completeness *Freeman, San Francisco* (1979).
8. A.V. Goldberg, and R.E. Tarjan, “A new approach to the maximum-flow problem,” *J. ACM* **35** (1988) 921-940.
9. H. Saran and V.V. Vazirani, “Finding k-cuts within twice the optimal,” *SIAM J. Computing* **24** (1995) 101-108.
10. S. Kapoor, “On minimum 3-cuts and approximating k-cuts using cut trees,” in: *Proceedings of the 5th International IPCO Conference on Integer Programming and Combinatorial Optimization, Springer-Verlag, London, UK,* 1996 132-146.
11. G. Karypis, V. Kumar, “Multilevel k-way hypergraph partitioning,” *VLSI Design* **11** (2000) 285-300.
12. D.R. Karger, and C. Stein, “A new approach to the minimum cut problem,” *J. ACM* **43** (1996) 601-640.
13. R. Klimmek, and F. Wagner, “A simple hypergraph min cut algorithm.” Internal Report B 96-02 Bericht FU Berlin, Fachbereich Mathematik und, Informatik, 1995.
14. Y. Kamidoi, S. Wakabayashi, and N. Yoshida, “A divide-and-conquer approach to the minimum k-way cut problem,” *Algorithmica* **32** (2002) 262-276.

15. Y. Kamidoi, N. Yoshida, and H. Nagamochi, “A deterministic algorithm for finding all minimum k-way cuts,” *SIAM J. Comput.* **36** (2006) 1329-1341.
16. W.-K. Mak, and D.F. Wong, “A fast hypergraph min-cut algorithm for circuit partitioning,” *Integration, VLSI J.* **30** (2000) 1-11.
17. B. Martin and D. Paulusma, “The computational complexity of disconnected cut and 2K2-partition,” *Proceedings of the 17th International Conference on Principles and Practice of Constraint Programming (CP 2011), Perugia, Italy, September 12-16, 2011*, LNCS **6876** (2011) 561-575.
18. N.Guttmann-Beck and R. Hassin, “Approximation algorithms for minimum K-Cut,” *Algorithmica* **21** (2000) 198-207.
19. H. Nagamochi, and T. Ibaraki, “Computing edge connectivity in multigraphs and capacitated graphs”, *SIAM J. Discrete Math.* **5** (1992) 54-66.
20. H. Nagamochi, S. Katayama, and T. Ibaraki, “A faster algorithm for computing minimum 5-way and 6-way cuts in graphs”, COCOON 99, LNCS, **1627** (1999) 164-173.
21. A. Schrijver,*Combinatorial optimization, polyhedra and efficiency*, Springer Volume A (2002).
22. J. Shi, and J. Malik, “Normalized cuts and image processing,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22(8)** (2000) 888-905.
23. D. B. Shmoys. "[Approximation algorithms for] Cut problems and their application to divide-and-conquer". In: Approximation Algorithms for NP-hard Problems, (D.S. Hochbaum, ed.) PWS, 1997, 192-235., *J. ACM* **44** (1997) 585-591.
24. M. Stoer, and F. Wagner, A simple min-cut algorithm, *J. ACM* **44** (1997) 585-591.
25. T. Ito, M. Kaminski, D. Paulusma and D. M. Thilikos, “Parameterizing cut sets in a graph by the number of their components” *Lecture Notes in Computer Science* **5878** (2009) 605-615.
26. M. Thorup, "Minimum k-way cuts via deterministic greedy tree packing", in: *Proceedings of the 40th Annual ACM Symposium on Theory of Computing* (STOC 2008), 2008, 159-166.
27. T.Tokuyama and J. Nakano “Geometric algorithms for the minimum cost assignment problem,” *Random Structures & Algorithms* **6** (1995) 393-406.
28. M. Xiao, “Finding minimum 3-way cuts in hypergraphs,” *Information Processing Letters* **110** (2010) 554-558.