Some Complexity Results for k-Cardinality Minimum Cut Problems

Maurizio Bruglieri †
Dipartimento di Elettronica e Informazione
Politecnico of Milan
Piazza Leonardo da Vinci 32, 20133 Milan, Italy
bruglier@elet.polimi.it

Matthias Ehrgott, Horst W. Hamacher
Fachbereich Mathematik
University of Kaiserslautern
Postfach 3049, D-67653 Kaiserslautern, Germany
{ehrgott, hamacher}@mathematik.uni-kl.de

Abstract

Many polynomially solvable combinatorial optimization problems (COP) become NP$-hard$ when we require solutions to satisfy an additional cardinality constraint. This family of problems has been considered only recently.

We study a new problem of this family: the *k-cardinality minimum cut problem*. Given an undirected edge-weighted graph the *k-cardinality minimum cut problem* is to find a partition of the vertex set V in two sets V_1, V_2 such that the number of the edges between V_1 and V_2 is exactly k and the sum of the weights of these edges is minimal. A variant of this problem is the *k-cardinality minimum s-t cut problem* where s and t are fixed vertices and we have the additional request that s belongs to V_1 and t belongs to V_2. We also consider other variants where the number of edges of the cut is constrained to be either less or greater than k.

For all these problems we show complexity results in the most significant graph classes.

Keywords: *k-cardinality minimum cut, cardinality constraint combinatorial optimization, computational complexity.*

† The work of M.B. has been supported by Ph.D. MA.C.R.O. Grant of the Department of Mathematics "F. Enriquez", University of Milan.

1. Introduction

Let $G = (V, E)$ be an undirected graph with vertex set V and edge set E.

Definition 1 A *cut* is a partition of vertex set V in two sets $V_1,\ V_2$. In this way a cut edge set $C := \{\{v_1, v_2\} \in E : v_1 \in V_1, v_2 \in V_2\}$ is associated to every cut.

We agree that a cut can be determined indifferently in one of the following ways:

1. by a pair $(V_1,\ V_2)$ of vertex sets called the shores of the cut, defining a partition of V;
2. by one vertex set S, understanding $V_1 = S$ and $V_2 = S^C$;
3. by the cut edge set C. For going back to the sets V_1 and V_2 we can use the following procedure with complexity $O(n)$.

   Procedure from C to (V_1, V_2)

   i) Set $V_1 = \emptyset$ and $V_2 = \emptyset$;

   ii) For all $v \in V \setminus \{V_1 \cup V_2\}$

   set $V_1 := V_1 \cup \{v\}$;

   for all $\{v, u\} \in \delta(v) \cap C$

   set $V_2 := V_2 \cup \{u\}$.

   Here $\delta(v) := \big\{\{v_1, v_2\} \in E :\ v_1 = v \text{ or } v_2 = v\big\}$.

Definition 2 Given $s, t \in V$ an *s-t cut* is a cut (V_1, V_2) such that $s \in V_1$ and $t \in V_2$.

When $G = (V, E)$ is a directed graph the previous definitions hold, too, but we agree that cut edge set C contains only the edges directed from V_1 to V_2. We note that in this case, unlike for undirected graphs, (V_1, V_2) and (V_2, V_1) define different cuts. From now on we always consider G to be an undirected graph, if not specified otherwise.

Let $w : E \to \mathbb{N}$ be a positive integer function on the edge set of graph G and let k be a positive integer.

Definition 3 The $k-$cardinality minimum cut problem ($k-card\ cut$) is the problem to find a cut such that cut edge set C has cardinality k and a given objective function $f(C)$ is minimized.

In particular we address two classical cases,

1. the sum objective function

$$f(C) = \sum_{e \in C} w(e)$$

and

2. the bottleneck objective function

$$f(C) = \max_{e \in C} w(e).$$

Definition 4 The $k-$cardinality minimum $s-t$ cut problem ($k-card\ s-t\ cut$) is defined analogously to $k-card\ cut$ with the additional request that the cut we want to find is an $s-t$ cut.

Definition 5 The $\leq k-card\ cut$ and $\leq k-card\ s-t\ cut$ problems are defined analogously to $k-card\ cut$ and $k-card\ s-t\ cut$ only that the cardinality of C is required to be less than or equal to k. Analogously we define the $\geq k-card\ cut$ and $\geq k-card\ s-t\ cut$ problems requiring the cardinality of C to be greater than or equal to k.

The simple example below shows that k–$card\ cut$, $\leq k$–$card\ cut$ and $\geq k$–$card\ cut$ can have different optimal solutions.

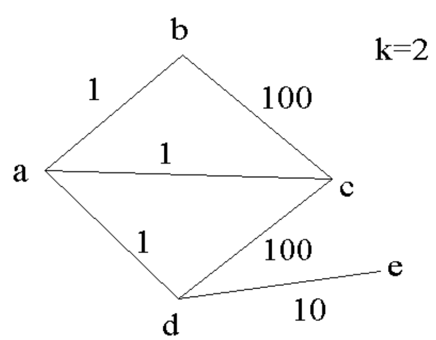

Figure 1. Illustrating different cardinality constraints.

$$S^1 = \{e\} \text{ is the solution for } \leq k-card\ cut$$

$$S^2 = \{b\} \text{ is the solution for } k-card\ cut$$

$$S^3 = \{a\} \text{ is the solution for } \geq k-card\ cut$$

The optimal values of $\leq k-card\ cut$ and of $\geq k-card\ cut$ are always less than or equal to the optimal value of $k-card\ cut$ because their feasible sets contain the feasible set of $k-card\ cut$. But between $\leq k-card\ cut$ and $\geq k-card\ cut$ there is no dominating relation: In the previous example the optimal solution of $\geq k-card\ cut$ has a smaller weight than the optimal solution of $\leq k-card\ cut$, but the opposite holds if we set $w(\{d,e\}) = 2$, for instance.
We can note that for every graph class for which *k–card cut* is easy $\leq$ *k–card cut* and $\geq$ *k–card cut* are easy, too, because they can be solved taking the best solution of $p - card\ cut$ with $p = 1, 2, \ldots, k$ and with $p = k, k+1, \ldots, |E|$, respectively.
A problem easier than the previous ones is the existence problem where we only want to decide whether there are feasible solutions. Theorem 1 of [3] establishes in general for any k-cardinality combinatorial optimization problem (k-*card COP*) the equivalence between existence and bottleneck problems in this sense: The bottleneck problem is solvable in polynomial time if and only if the the existence problem is polynomially solvable, too. As a consequence the existence and bottleneck problems are both easier than sum problems. Any of the results we prove below for uniform weights $w(e) = 1$ therefore apply to existence and bottleneck problems (with arbitrary weights). We can therefore restrict our discussion to problems with sum objective.
We note that without cardinality constraints the previous problems are easy because they become minimal cut problems and several efficient algorithms exist in the literature for solving the latter (see [6], [7] and [8]).

2. Complexity of κ-Cardinality Cut Problems

2.1 General Graphs

Theorem 1 $K-card\ cut$ and $k\text{–}card\ s\text{–}t\ cut$ are $strongly\ \mathcal{NP}-complete$ even if $w(e) = 1$ for all $e \in E$.

Proof: We prove the result for $k-card\ cut$ first. It is easy to see that the recognition version of $k\text{–}card\ cut$ belongs to $\mathcal{NP}$. For proving the strong hardness we polynomially reduce *simple max cut* to $k\text{–}card\ cut$. An instance of *simple max cut* is an undirected graph $G = (V, E)$ where we look for a cut with the maximum number of edges. We can transform this instance into instances for $k-card\ cut$ considering the same graph with weight $w(e) = 1$ for all $e \in E$ and values of k between 1 and $|E|$. A solution of $k-card\ cut$ for the maximum feasible value of k is also a solution of *simple max cut*. Finally the proof follows from *strong* $\mathcal{NP}-completeness$ of *simple max cut* (see [1], page 210).

For $k-card\ s-t\ cut$ it is easily seen that this problem belongs to $\mathcal{NP}$, too. For proving the *strong hardness* we polynomially reduce *k–card cut* to *k–card s–t cut*. Solving $k-card\ s-t\ cut$ for all pairs of vertices s, t and taking the best solution we obtain a solution for $k-card\ cut$. ∎

Corollary 1 The existence and bottleneck problems for $k-card\ cut$ are $\mathcal{NP}-complete$.

Remark 1 For some classes of graphs, for example for planar graphs, *simple max cut* belongs to $\mathcal{P}$ (see page 247 of [5]). Therefore for these graphs the proof of Theorem 1 is not valid. We discuss the problem for planar graphs later.

Proposition 1 The $\geq k-card\ cut$ and $\geq k-card\ s-t\ cut$ problems are $strongly\ \mathcal{NP}-complete$ even if $w(e) = 1$ for all $e \in E$.

Proof: We can proceed through a reduction from *simple max cut* like in the proof of Theorem 1. ∎

Proposition 2 The $\leq k\text{–}card\ cut$ problem is in $\mathcal{P}$ if $w(e) = 1$ for all $e \in E$.

Proof: Let $\overline{k}$ be the cardinality of a solution of the minimal cut problem. If $k < \overline{k}$ then $\leq k\text{–}card\ cut$ is infeasible otherwise any solution of the minimal cut problem is also a solution of $\leq k-card\ cut$. ∎

Proposition 3 Let $\overline{k}$ be the cardinality of a solution of the minimal cut problem. For all $k \geq \overline{k}$ the $\leq k-card\ cut$ problem is in $\mathcal{P}$, for all $k \leq \overline{k}$ the $\geq k-card\ cut$ problem is in $\mathcal{P}$, even if the weights are not uniform.

Proof: In these cases a solution is given by the solution of the min cut problem with cardinality $\overline{k}$. ∎

We now proceed to consider some specific classes of graphs.

2.2 Complete Graphs

Lemma 1 If $G = (V, E)$ is a simple and complete graph $k-card\ cut$ and $k-card\ s-t\ cut$ are feasible

$$\Leftrightarrow \; k = j(n-j) \;\; \text{with } j \in \left\{1, \ldots, \left\lfloor \frac{n}{2} \right\rfloor \right\}, \tag{1}$$

where $n = |V|$.

Proof: Let the vertex set S be a shore of a cut. We can suppose $|S| \leq \left\lfloor \frac{n}{2} \right\rfloor$ because otherwise S^C (where the superscript C denotes the complement of a set) is so and it determines the same cut. We set $j = |S|$, therefore $|S^C| = n - j$. Since the graph is simple and complete each vertex of S is connected with every vertex of S^C. Therefore the cardinality of the cut is $j(n-j)$ and the cardinalities of all possible cuts are given by (1). ∎

Proposition 4 If G is a simple and complete graph and $w(e) = 1$ for all $e \in E$ then both $k-card\ cut$ and $k-card\ s-t\ cut$ are in P.

Proof: We have to solve the equation $j^2 - nj + k = 0$ with respect to j. If $\Delta := n^2 - 4k < 0$ or if $\tilde{j} := \frac{n-\sqrt{\Delta}}{2}$ is not integer then both cut problems are infeasible because of Lemma 1. Otherwise, an optimal solution S is given by any choice of $\tilde{j}$ vertices of V, for $k-card\ cut$, and by $\{s\}$ union any choice of $\tilde{j} - 1$ vertices of $V \setminus \{s, t\}$, for $k-card\ s-t\ cut$. ∎

Lemma 2 If G is a simple complete graph with non-uniform weights on the edges the *equicut* problem is *strongly* $\mathcal{NP}-$ *complete*.

Proof: The *equicut* problem is *strongly* $\mathcal{NP}-$ *complete* for general graphs as proved in [9]. Now we will reduce the *equicut* problem for general graph to the *equicut* problem for complete graphs. Given any graph we transform it into a complete graph adding edges with weight 0. Since the cuts of the new graph have the same total weight and the same vertices as the corresponding cuts of the original graph, an optimal equicut in the original general graph corresponds to an optimal equicut in the new (complete) graph. ∎

Proposition 5 If G is a simple complete graph with non-uniform weights on the edges *$k-$card cut* is *strongly $\mathcal{NP}-$complete*.

Proof: Reduction from equicut. Given an instance of equicut we can solve it solving the *$k-$card cut* problem with $k = \lfloor \frac{n}{2} \rfloor \lceil \frac{n}{2} \rceil$. The *strong hardness* derives from the *strong hardness* of equicut for complete graphs established by Lemma 2. ∎

2.3 Complete Bipartite Graphs

Definition 6 A graph $G = (V, E)$ is called *complete bipartite* if its vertex set V can be partitioned into two nonempty, disjoint sets L and R such that no two vertices in L and no two vertices in R are linked by an edge and every vertex in L is linked to every vertex in R by exactly one edge. If $|L| = n$ and $|R| = m$, then G is denoted by $K_{n,m}$.

Lemma 3 Given a complete bipartite graph $K_{n,m} = (V, E)$, *$k-$card cut* and *$k-$card $s-t$ cut* are feasible

$$\begin{aligned} \Leftrightarrow\ & k = in + jm - 2ij \\ & \text{with } i \in \{0, 1, \ldots, m\},\ j \in \{0, 1, \ldots, n\},\ \ i + j \leq \left\lfloor \frac{m+n}{2} \right\rfloor \end{aligned} \tag{2}$$

Proof: Let L and R be the sets introduced in Definition 6. Let the vertex set S be a shore of a cut. We can suppose

$$|S| \leq \left\lfloor \frac{|V|}{2} \right\rfloor = \left\lfloor \frac{m+n}{2} \right\rfloor$$

because otherwise S^C is so and it determines the same cut. Let $j = |S \cap L|$ and $i = |S \cap R|$, thus j is an integer between 0 and $|L| = n$, i is an integer between 0 and $|R| = m$ and

$$i + j = |S| \leq \left\lfloor \frac{m+n}{2} \right\rfloor .$$

Let us introduce the notation $\delta(A, B)$ and $\delta(A)$ for all $A,\ B \subset V$ as follows:

$$\delta(A, B) := \{\{v_1, v_2\} \in E : v_1 \in A, v_2 \in B\}$$
$$\delta(A) := \delta(A, A^C).$$

Therefore

$$\delta(S) = \delta(S \cap L) \cup \delta(S \cap R) \setminus \delta(S \cap L, S \cap R)$$

and

$$|\delta(S)| = |\delta(S \cap L)| + |\delta(S \cap R)| - 2|\delta(S \cap L, S \cap R)| =$$
$$= jm + in - 2ij\ .$$

In this way the cardinalities of all possible cuts are given by (2). ■

Remark 2 Given a k value we can have several pairs of values $i,\ j$ satisfying (2). For example for the graph $K_{3,2}$ both $i = j = 1$ and $i = 1,\ j = 0$ satisfy (2) for $k = 3$. Moreover, unlike for complete graphs there is no one to one correspondence between the cardinality k of the cut and the cardinality of the minimal shore S of the cut. The example below shows that two cuts with the same cardinality can have minimal shores with different cardinalities.

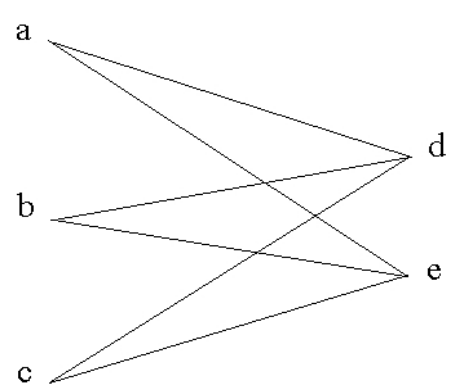

Figure 2. The complete bipartite graph $K_{3,2}$.

Referring to Figure 2., $S_1 = \{e\}$ and $S_2 = \{c, d\}$ determine both cuts with cardinality $k = 3$ but they are sets of different cardinality. Analogously, it is easy to see that two cuts with different cardinalities can have minimal shores of the same cardinality.

Proposition 6 Given a complete bipartite graph $K_{n,m} = (V, E)$ such that $w(e) = 1$ for all $e \in E$, both *$k-$card cut* and *$k-$card $s-t$ cut* are polynomially solvable.

Proof: Through formula (2) of Lemma 3 we calculate in polynomial time the cardinalities of all feasible cuts. If the given value of k is not among them then both *$k-$card cut* and *$k-$card $s-t$ cut* are infeasible. Otherwise we can go back to a pair of values of i and j satisfying (2). In this case any choice of j vertices of L and of i vertices of R is a solution for *$k-$card cut*.

For *$k-$card $s-t$ cut* we distinguish two cases:

a) Vertices s and t both belong to L. (We can reason analogously if they both belong to R).

b) Vertex s belongs to L and vertex t belongs to R. (If the opposite occurs we only have to exchange the names of s and t).

In case a), if every pair of values i, j satisfying (2) for the given value of k, has $j = 0$ or $j = n$ then the problem is infeasible. Otherwise suppose i, j satisfy (2) with j between 1 and $n - 1$. In this case an optimal solution S is given by $\{s\}$ union any choice of $j - 1$ vertices of $L \setminus \{s, t\}$ union any choice of i vertices of R. Now consider case b). Let i, j be a pair of values satisfying (2) for the given value of k. We distinguish the following three subcases:

i) $i \leq m - 1$ and $j \neq 0$,

ii) $i \leq m - 1$ and $j = 0$,

iii) $i = m$.

In case i) an optimal solution S is given by $\{s\}$ union any choice of $j - 1$ vertices of $L \setminus \{s\}$ union any choice of i vertices of $R \setminus \{t\}$. In case ii) S is given by $\{t\}$ union any choice of $i - 1$ vertices of $R \setminus \{t\}$. Finally, in case iii) S is given by R union any choice of j vertices of $L \setminus \{s\}$. ■

Remark 3 The problem to establish the complexity of *k–card cut* and *k–card s–t cut* in complete bipartite graphs with non-uniform weights is still open because unlike for general graphs a reduction of max cut to *$k-$card cut* is useless because max cut is solvable in polynomial time for complete bipartite graphs (see [13]). Moreover, unlike for complete graphs we cannot reduce the equicut problem to these problems due to Remark 2.

2.4 Trees

Lemma 4 For a graph G, $C \subset E$ is the edge set of a cut if and only if C has an even number (possibly zero) of edges in common with any cycle of G.

Proof: See [2]. ■

Remark 4 We note that if G is a directed graph Lemma 4 does not hold.

Proposition 7 When G is a tree, $k-card\ cut$ is in $\mathcal{P}$.

Proof: Since G is a tree, it has no cycle. So from Lemma 4 any choice of k edges is a $k-card\ cut$. Therefore in this case the k edges with smallest weight are a solution of $k-card\ cut$, and this solution can be determined in polynomial time. ■

Lemma 5 Let $G = (V, E)$ be a tree and $s,\ t \in V$ then every $s-t\ cut$ has an odd number of edges in common with the path between vertex s and vertex t.

Proof: Let P be the path between vertex s and vertex t (the existence and uniqueness of the path is ensured from G being a tree). Let C be the edge set of the $s-t$ cut. Let $C \cap P = \{e_1, e_2, \ldots, e_r\}$ and let r be even. We suppose e_i precedes e_{i+1} along P and $e_i = \{u_i, v_i\}$ for $i = 1, \ldots, r$. Let $s \in S$. Since $e_1 = \{u_1, v_1\}$ is the first edge of $C \cap P$ every possible vertex between s and u_1 along P belongs to S whereas $v_1 \in S^C$ implies that every possible vertex between v_1 and u_2 along P belongs to S^C and $v_2 \in S$. Generalizing, we have u_{2i-1} and $v_{2i}\ \in S$ whereas u_{2i} and $v_{2i-1}\ \in S^C$ for $i = 1, \ldots, \frac{r}{2}$. Since r is even $v_r \in S$ so that every other possible vertex that follows v_r along P belongs to S, in particular $t \in S$. Since s and $t \in S$ the edges of C do not make up an $s-t\ cut$. Thus r must be odd. ■

Theorem 2 When G is a tree, $k-card\ s-t\ cut$ belongs to class $\mathcal{P}$.

Proof: Let P be the path between the vertex s and the vertex t. By Lemma 5 every $s-t$ cut has an odd number of edges in common with P. Let F be the set of the k smallest weight edges in G. If $|P \cap F|$ is odd then the edge set $C = F$ is a solution of $k-card\ s-t\ cut$. Else if $|P \cap F|$ is even we obtain C from F modifying F as little as possible to have $|P \cap C|$ odd. There are the following four cases:

a) $P \setminus F = \emptyset$,

b) $F \setminus P = \emptyset$,

c) $P \cap F = \emptyset$,

d) any other case.

In case a) let $e^* \in P$ and $\tilde{e} \in F^C$ be such that

$$w(e^*) = \max_{e \in P} w(e) \qquad w(\tilde{e}) = \min_{e \in F^C} w(e),$$

respectively.

An optimal solution is given by $C = F \setminus \{e^*\} \cup \{\tilde{e}\}$.

In case b) let $e^* \in F$ and $\tilde{e} \in P^C$ be such that

$$w(e^*) = \max_{e \in F} w(e) \qquad w(\tilde{e}) = \min_{e \in P^C} w(e),$$

respectively.

An optimal solution is given by $C = F \setminus \{e^*\} \cup \{\tilde{e}\}$.

In case c) let $e^* \in F$ and $\tilde{e} \in P$ be such that

$$w(e^*) = \max_{e \in F} w(e) \qquad w(\tilde{e}) = \min_{e \in P} w(e),$$

respectively.

An optimal solution is given by $C = F \setminus \{e^*\} \cup \{\tilde{e}\}$.

In case d) if $(P \cup F)^C \neq \emptyset$ let $e^* \in P \setminus F$, $\hat{e} \in (P \cup F)^C$, $\tilde{e} \in F \setminus P$ and $\overline{e} \in P \cap F$ be such that

$$w(e^*) = \min_{e \in P \setminus F} w(e) \qquad w(\hat{e}) = \min_{e \in (P \cup F)^C} w(e),$$

$$w(\tilde{e}) = \max_{e \in F \setminus P} w(e) \qquad w(\bar{e}) = \max_{e \in P \cap F} w(e)$$

respectively.

Then it easy to see that if $w(e^*) - w(\tilde{e}) \leq w(\hat{e}) - w(\bar{e})$ an optimal solution is given by $C = F \setminus \{\tilde{e}\} \cup \{e^*\}$, otherwise it is given by $C = F \setminus \{\bar{e}\} \cup \{\hat{e}\}$.

If $(P \cup F)^C = \emptyset$ an optimal solution is $C = F \setminus \{\tilde{e}\} \cup \{e^*\}$. ∎

2.5 Grid Graphs

Definition 7 A simple grid graph is a graph $G = (V, E)$ with $(h+1)(l+1)$ vertices arranged in $l + 1$ horizontal rows and $h + 1$ columns each, and edges connecting vertices in adjacent rows (columns) vertically (horizontally). The horizontal and vertical lengths of G are h and l, respectively.

Lemma 6 If $G = (V, E)$ is a simple grid graph it has a cut of cardinality m with $m = |E|$.

Proof: Let h and l be the horizontal length and vertical length of G. Let $v_{i,j}$ be the vertex at row i and column j in G for $i = 0, 1, \dots, l$, $j = 0, 1, \dots, h$. It is easy to see that the set T defined as

$$T := \{v_{i,j} \in V : 0 \le i \le l,\ 0 \le j \le h,\ i, j \text{ both even or } i, j \text{ both odd }\}$$

generates a cut with edge set equal to E. ∎

Lemma 7 If $G = (V, E)$ is a simple grid graph with horizontal length h and vertical length l, vertex set T defined in Lemma 6 has $\left\lceil \frac{(h-1)(l-1)}{2} \right\rceil$ vertices of degree 4 and $h+l$ vertices of degree 2 or 3. In particular it has 4 vertices of degree 2 if h and l are both even and it has 2 vertices of degree 2 in any other case.

Proof: The subset P of T made up by vertices of T of degree 4 is given by

$$P := \{v_{i,j} \in V : 1 \le i \le l-1,\ 1 \le j \le h-1,\ i, j \text{ both even or } i, j \text{ both odd }\}$$

The cardinality of P is

$$\left(\left\lfloor \frac{l-2}{2} \right\rfloor + 1\right)\left(\left\lfloor \frac{h-2}{2} \right\rfloor + 1\right) + \left\lceil \frac{l-2}{2} \right\rceil \left\lceil \frac{h-2}{2} \right\rceil = \left\lceil \frac{(h-1)(l-1)}{2} \right\rceil .$$

The cardinality of T is

$$\left(\left\lfloor \frac{l}{2} \right\rfloor + 1\right)\left(\left\lfloor \frac{h}{2} \right\rfloor + 1\right) + \left\lceil \frac{l}{2} \right\rceil \left\lceil \frac{h}{2} \right\rceil = \left\lceil \frac{(h+1)(l+1)}{2} \right\rceil .$$

Therefore the number of vertices of degree 2 or 3 is

$$|T| - |P| = \left\lceil \frac{(h+1)(l+1)}{2} \right\rceil - \left\lceil \frac{(h-1)(l-1)}{2} \right\rceil = h + l \ . \quad ∎$$

Theorem 3 If $G = (V, E)$ is a simple grid graph *k−card cut* and *k−card s−t cut* are feasible

$$\Leftrightarrow \quad k = 2, 3, \ldots, m-2, m \tag{3}$$

with $m = |E|$, that is they are feasible for all values of k unless $k = 1$ and $k = m-1$.

Proof: Let us suppose $h \geq 3$ and $l \geq 1$ (or vice versa) otherwise the theorem is trivial. Let T and P be the vertex sets defined in Lemma 6 and Lemma 7. Let S be the vertex set of the cut of cardinality k we are looking for.

If $k \leq 4\left\lceil \frac{(h-1)(l-1)}{2} \right\rceil$ we set S equal to $p = \lfloor \frac{k}{4} \rfloor$ vertices of P including vertex $v_{1,1}$ and

$$\begin{aligned} S &:= S \cup \{v_{0,2}\} & &\text{if } k - 4p = 3 \\ S &:= S \cup \{v_{0,0}\} & &\text{if } k - 4p = 2 \\ S &:= S \setminus \{v_{1,1}\} \cup \{v_{0,0}, v_{0,2}\} & &\text{if } k - 4p = 1 \\ S &\text{ remains unchanged} & &\text{if } k - 4p = 0. \end{aligned}$$

If $4\left\lceil \frac{(h-1)(l-1)}{2} \right\rceil < k \leq m - d$ where

$$d = \begin{cases} 8\ , & \text{if } h,\ l \text{ both even} \\ 4\ , & \text{otherwise} \end{cases}$$

we set $S := P$, we add to S q vertices of the set

$$Q := \{v_{i,j} \in V : i = 0, l,\ 1 \leq j \leq h-1,\ i, j \text{ both even or } i, j \text{ both odd } \}$$

including vertex $v_{0,2}$, with

$$q := \left\lfloor \frac{k - 4p^*}{3} \right\rfloor \quad p^* := \left\lceil \frac{(h-1)(l-1)}{2} \right\rceil$$

and we set

$$\begin{aligned} S &:= S \cup \{v_{0,0}\} & &\text{if } k - 4p^* - 3q = 2 \\ S &:= S \setminus \{v_{0,2}\} \cup \{v_{0,0}, \tilde{v}\} & &\text{if } k - 4p^* - 3q = 1 \\ S &\text{ remains unchanged} & &\text{if } k - 4p^* - 3q = 0, \end{aligned}$$

where

$$\tilde{v} = \begin{cases} v_{0,h}\ , & \text{if } h \text{ even} \\ v_{l,0}\ , & \text{if } l \text{ even and } h \text{ odd} \\ v_{l,h}\ , & \text{if } l \text{ and } h \text{ both odd.} \end{cases}$$

If $k > m - d$ we set

$$S := T \quad \text{if } k = m$$

$$S := T \setminus \{v_{0,0}\} \quad \text{if } k = m - 2$$

$$S := T \setminus \{v_{0,2}\} \quad \text{if } k = m - 3$$

and in addition in the case h and l are both even we set

$$S := T \setminus \{v_{1,1}\} \quad \text{if } k = m - 4$$

$$S := T \setminus \{v_{0,0}, v_{0,2}\} \quad \text{if } k = m - 5$$

$$S := T \setminus \{v_{0,0}, v_{1,1}\} \quad \text{if } k = m - 6$$

$$S := T \setminus \{v_{0,2}, v_{1,1}\} \quad \text{if } k = m - 7. \blacksquare$$

2.6 Planar Graphs

Definition 8 A *planar graph* is a graph which is isomorphic to a geometric graph in the plane, i.e. it can be drawn in the plane in such a way that its edges intersect only at their endnodes.

Definition 9 An *isthmus* of a graph G is an edge whose removal increases the number of connected components of G.

Theorem 4 When $G \cup \{\{s,t\}\}$ is a planar graph without isthmus, $\leq$ *k–card s–t cut* is in $\mathcal{P}$.

Proof: If graph G contains an edge $\{s,t\}$, we let $G \cup \{\{s,t\}\}$ indicate the union of G with an additional edge $\{s,t\}$. Since the graph $G \cup \{\{s,t\}\}$ is planar it is possible to associate to it the dual graph G^* according to the following rule: inside each face F_i of the graph G we put a vertex v_i^* of the graph G^*, and to each edge e_i of G we assign that edge e_i^* of G^* that connects the vertices u_i^* and v_i^* corresponding to the faces F_i, H_i on the two sides of the edge e_i. Since G does not have any isthmus, G^* does not have any loop. Let s^* and t^* be the end vertices of the dual of edge $\{s,t\}$: We can uniquely determine (the position of) these two vertices considering the dual of another edge incident with s or with

t. It is easy to see that every $s-t$ cut of graph G corresponds to a path (in general not elementary) from s^* to t^* of $G^* \setminus \{\{s^*, t^*\}\}$ and vice versa. Moreover, the cardinality of the cut is equal to the length (i.e. the number of edges) of the path. Therefore $\leq k-card\ s-t\ cut$ is equivalent to finding a minimum path (now elementary) from vertex s^* to vertex t^* with cardinality $\leq k$, where the weights of the dual edges are equal to the weights of the original edges of G. This last problem is solvable in polynomial time through Ford's label correcting algorithm (see [4], page 136). So in this case $\leq k-card\ s-t\ cut \in \mathcal{P}$. ■

Corollary 2 If $G \cup \{\{s,t\}\}$ is a planar graph without isthmus for all $s,\ t \in V$ then $\leq k-card\ cut \in \mathcal{P}$.

Proof: The corollary follows from Theorem 4 solving $\leq k-card\ s-t\ cut$ for all vertex pairs $s,\ t$ of V. ■

Remark 5 In planar graphs $k-card\ s-t\ cut$ is "equivalent" to $k-$cardinality minimum $s-t$ path (with possible repetitions of vertices) and the latter problem has not yet been studied in the literature, as far as we know. Therefore the computational complexity of $k-card\ s-t\ cut$ in planar graphs is still an open problem.

Considering the relation between the *max cut* problem and k–*card cut* established in Theorem 1, and the fact that max cut is polynomial for planar graphs, established by Theorem 5 of [10] which we report below, it is interesting to consider the polyhedral structure of both problems for planar graphs.

Theorem 5 Let

$$
\begin{aligned}
P_C(G) := \{x \in \mathbb{R}^{|E|} : 0 \leq x_e \leq 1 \quad \forall e \in E \\
& x(F) - x(C \setminus F) \leq |F| - 1, \\
& \forall \text{ circuit } C \subset E \text{ and } \forall F \subset C, |F| \text{ odd } \}
\end{aligned}
$$

and let $CUT(G)$ be the cut polytope of G, i.e. the convex hull of all incidence vectors of cuts of G, then

$$P_C(G) = CUT(G) \ \Leftrightarrow \ G \text{ is not contractible to } K_5,$$

where K_5 is the complete graph on 5 vertices.

This theorem shows that the max-cut problem is solvable in polynomial time for the class of graphs not contractible to K_5: The separation problem for all inequalities in the concise description of $CUT(G)$ is solvable in polynomial time since it can be reduced to the computation of n shortest paths as shown in [11]. Since, by a well-known characterization theorem (Kuratowski's theorem, see Theorem 4.5 of [12]), planar graphs are those graphs which are not contractible to K_5 or $K_{3,3}$, the previous result holds for planar graphs, too. We would like to adapt this result for $k-card\ cut$.

Let $KCUT(G,k)$ denote the convex hull of all incidence vectors of the k–$card\ cut$, i.e.

$$KCUT(G,k) := \text{ conv } \left\{ x \in \{0,1\}^{|E|} : \ x \text{ is a cut and } \sum_{e \in E} x_e = k \right\}.$$

Therefore

$$\begin{aligned} KCUT(G,k) &\subset \text{conv} \left\{ x \in \{0,1\}^{|E|} : \ x \text{ is a cut} \right\} \cap \text{conv} \left\{ x \in \{0,1\}^{|E|} : \ \sum_{e \in E} x_e = k \right\} \\ &= CUT(G) \cap \left\{ x \in [0,1]^{|E|} : \ \sum_{e \in E} x_e = k \right\} \qquad (4) \end{aligned}$$

If the opposite inclusion held, too, we could conclude

$$\begin{aligned} KCUT(G,k) = CUT(G) &\cap \left\{ x \in [0,1]^{|E|} : \ \sum_{e \in E} x_e = k \right\} \\ = P_C(G) &\cap \left\{ x \in [0,1]^{|E|} : \ \sum_{e \in E} x_e = k \right\} \end{aligned}$$

and so we also would have a compact description for the $k-card\ cut$ polytope. But unfortunately the opposite inclusion does not hold in (4) as the example below shows.

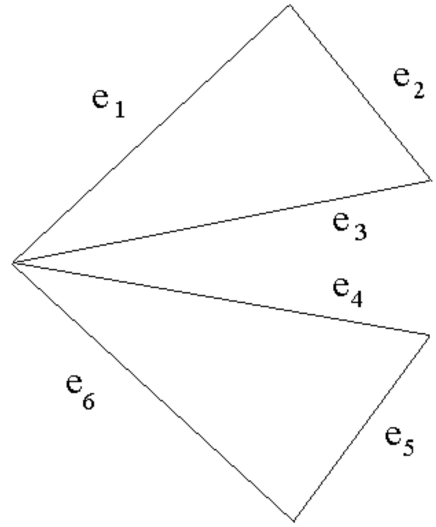

Figure 3. Planar graph G for which
$KCUT(G,3) \neq CUT(G) \cap \left\{x \in [0,1]^{|E|} : \sum_{e\in E} x_e = 3\right\}$.

For the graph G drawn in Figure 3. $KCUT(G,3) = \emptyset$ because this graph has only cuts with cardinality 2 or 4. But $CUT(G) \cap \{x \in [0,1]^{|E|} : \sum_{e\in E} x_e = k\} \neq \emptyset$ because, for example, $\tilde{x} = (1, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 0, \frac{1}{2})$ belongs to this set. Indeed, $x \in CUT(G)$ because $\tilde{x} = \frac{1}{2}x' + \frac{1}{2}x''$ where $x' = (1,0,1,1,0,1)$ and $x'' = (1,1,0,0,0,0)$ are incidence vectors of cuts of G. Moreover, $\sum_{e\in E} \tilde{x}_e = 3$. Examples of grid graphs and triangulations, for which the equality does not hold, can be easily constructed, too.

Are there other graphs for which the two polyhedra coincide? The answer is yes: trees. Due to the proof of Proposition 7, any subset of edges is a cut, so any subset of k edges is a k-card cut. Therefore for trees

$$KCUT(G,k) := \text{ conv } \left\{x \in \{0,1\}^{|E|} : \sum_{e\in E} x_e = k\right\}$$

$$= \left\{x \in [0,1]^{|E|} : \sum_{e\in E} x_e = k\right\}$$

$$= \text{ conv } \left\{x \in \{0,1\}^{|E|}\right\} \cap \left\{x \in [0,1]^{|E|} : \sum_{e \in E} x_e = k\right\}$$

$$=: CUT(G) \cap \left\{x \in [0,1]^{|E|} : \sum_{e \in E} x_e = k\right\}$$

2.7 Summarizing Tables

We summarize the complexity results obtained for *k−card cut*, $\leq$ *k−card cut* and $\geq$ *k−card cut* in the following three tables:

graph class	**existence**	**sum**
general	*strongly* $\mathcal{NP}$ − *complete*	*strongly* $\mathcal{NP}$ − *complete*
complete	$\mathcal{P}$	*strongly* $\mathcal{NP}$ − *complete*
complete bipartite	$\mathcal{P}$	?
tree	$\mathcal{P}$	$\mathcal{P}$
grid	$\mathcal{P}$	?
planar	?	?

Table 1. Results for *k−card cut*.

graph class	**existence**	**sum**
general	$\mathcal{P}$	?
complete	$\mathcal{P}$	?
complete bipartite	$\mathcal{P}$	?
tree	$\mathcal{P}$	$\mathcal{P}$
grid	$\mathcal{P}$	?
planar	$\mathcal{P}^*$	$\mathcal{P}^*$

* under the condition of Corollary 2

Table 2. Results for $\leq$ *k−card cut*.

graph class	existence	sum
general	*strongly* $\mathcal{NP}-$*complete*	*strongly* $\mathcal{NP}-$*complete*
complete	$\mathcal{P}$	?
complete bipartite	$\mathcal{P}$	?
tree	$\mathcal{P}$	$\mathcal{P}$
grid	$\mathcal{P}$	?
planar	?	?

Table 3. Results for $\geq k-card\ cut$.

In these tables the symbol "?" indicates open problems.

Bibliography

[1] M.R. Garey, D.S. Johnson
Computers and Intractability: A Guide to the Theory of NP-Completeness
W.H.Freeman & Co., 1979

[2] G.I. Orlova, Y.G. Dorfman
Finding the maximum cut in a graph
Engineering Cybernetics vol. 10, 502-506, 1972

[3] M. Ehrgott, H.W. Hamacher, F. Maffioli
Fixed cardinality combinatorial optimization problems - A survey
Report in Wirtschaftsmathematik Nr.56/1999,
Fachbereich Mathematik, Universität Kaiserslautern

[4] R.K. Ahuja, T.L. Magnanti, J.B. Orlin
Network Flows. Theory, Algorithms and Applications
Prentice-Hall, 1993

[5] M. Dell'Amico, F. Maffioli, S. Martello
Annotated Bibliographies in Combinatorial Optimization

John Wiley & Sons, 1997

[6] M. Padberg, G. Rinaldi
An efficient algorithm for the minimum capacity cut problem
Mathematical Programming, vol. 47, 19-39, 1990

[7] M. Stoer, F. Wagner
A simple min-cut algorithm
Journal of the ACM, vol. 44, 585-591, 1997

[8] D. Karger
Minimum cuts in near-linear time
downloadable at *http://theory.lcs.mit.edu/~karger*

[9] M.R. Garey, D.S. Johnson, L. Stockmeyer
Some simplified NP-complete graph problems
Theoretical Computer Science, vol. 1, 237-267, 1976

[10] F. Barahona, M. Grötschel, M. Jünger, G. Reinelt
An application of combinatorial optimization to statistical physics and circuit layout design
Operations Research, vol. 36, No. 3, 493-513, 1988

[11] F. Barahona
On some applications of the chinese postman problem
Algorithms and Combinatorics. Vol.9-Paths, Flows, and VLSI-Layout.
Springer-Verlag, 1990

[12] R.G. Busacker, T.L. Saaty
Finite Graphs and Networks: An Introduction with Application
McGraw-Hill, 1965

[13] M. Grötschel, W.R. Pulleyblank
Weakly bipartite graphs and the max-cut problem
Operations Research Letters, vol. 1, No. 1, 23-27, 1981