A SIMPLE ALGORITHM FOR MULTICUTS IN PLANAR GRAPHS WITH OUTER TERMINALS

Cédric Bentz[*]

Abstract

Given an edge-weighted graph G and a list of source-sink pairs of terminal vertices of G, the minimum multicut problem consists in selecting a minimum weight set of edges of G whose removal leaves no path from the ith source to the ith sink, for each i. Few tractable special cases are known for this problem. In this paper, we give a simple polynomial-time algorithm solving it in undirected planar graphs where (I) all the terminals lie on the outer face and (II) there is a bounded number of terminals.

Keywords: Multicuts, Planar graphs, Graph algorithms.

1 Introduction

In this note, we study the polynomial-time solvability of the minimum multicut problem in planar undirected graphs.

Assume we are given an n-vertex graph $G = (V, E)$, a *weight function* $c : E \to \mathbb{Z}^+$ on the edges of G and a list $\mathcal{N}$ of pairs (source s_i, sink s'_i) of *terminal* vertices. The *minimum multicut problem* (MinMC) consists in selecting a minimum weight set of edges whose removal leaves no path from s_i to s'_i for each i. The *minimum multiterminal cut problem* (MinMTC) is a special case of MinMC in which, given a set of vertices $\mathcal{T} = \{t_1, \ldots, t_{|\mathcal{T}|}\}$, the terminal pairs are (t_i, t_j) for $i \neq j$.

For $|\mathcal{N}| = 1$, MinMC is equivalent to the *minimum cut problem*, and therefore is polynomial-time solvable both in directed and in undirected graphs [7]. For the same reason, MinMTC is polynomial-time solvable for $|\mathcal{T}| = 2$ in undirected graphs. Unfortunately, MinMTC becomes $\mathcal{NP}$-hard, and even $\mathcal{APX}$-hard, as soon as $|\mathcal{T}| = 3$ in undirected graphs [5], and as soon as $|\mathcal{T}| = 2$ in directed graphs [11]. This implies that MinMC is $\mathcal{APX}$-hard for fixed $|\mathcal{N}| \geq 3$ in undirected graphs, and for fixed $|\mathcal{N}| \geq 2$ in directed graphs. Note, however, that MinMC is tractable in undirected graphs when $|\mathcal{N}| = 2$ [19].

[*]LRI, Univ. Paris-Sud and CNRS, Orsay, F-91405.
Phone: +33 (0) 1 69 15 31 06. E-mail address: cedric.bentz@lri.fr

For an arbitrary number of source-sink pairs, MinMC is tractable in paths and in chains, but $\mathcal{APX}$-hard even in unweighted stars [10]. However, when $|\mathcal{N}|$ is fixed, it becomes tractable in trees, and even in bounded tree-width graphs [2]. Moreover, there is a polynomial-time approximation scheme (PTAS) for MinMC in unweighted undirected graphs of bounded tree-width and bounded degree, but dropping any of these three assumptions leads to $\mathcal{APX}$-hardness (instead of $\mathcal{NP}$-hardness only) [3]. There also exists an $O(\log |\mathcal{N}|)$-approximation algorithm for MinMC in general graphs [9] and an $O(1)$-approximation algorithm in planar graphs [18].

In this note, we show that MinMC is polynomial-time solvable in planar graphs where (I) all the terminals lie on the outer face (we shall say in this case that the graph has **outer** terminals) and (II) $|\mathcal{N}|$ is fixed. We also give a linear-time algorithm for the special case where $|\mathcal{N}| = 2$.

The interest of this result may follow from several reasons. First, it generalizes the result for trees given in [10] and is, to our best knowledge, the first tractability result for MinMC in graphs more general than trees (and with unbounded tree-width). Second, it should be noticed once again that MinMC remains $\mathcal{APX}$-hard if the graph is not planar (even for $|\mathcal{N}| = 3$) [5], or if $|\mathcal{N}|$ is not fixed (even in trees) [10]. This latter result implies the $\mathcal{NP}$-hardness of MinMC in planar graphs with outer terminals and arbitrary $|\mathcal{N}|$, while MinMTC is tractable in this case (this follows from a nice reduction to a tractable special case of the Steiner tree problem) [4]. Third, MinMTC, which is a special case of MinMC, is tractable in planar graphs if $|\mathcal{N}|$ is fixed (it is $\mathcal{NP}$-hard otherwise) [5, 12]. As a comparison, the complexity of MinMC in planar graphs is still open when $|\mathcal{N}|$ is fixed: the case studied in this paper is thus a natural special case of this more general (open) case. Indeed, several flow and cut problems have already been efficiently solved in planar graphs with outer terminals, although they were much harder in "general" planar graphs [4, 8, 14, 15, 16].

Furthermore, an interesting feature of our approach is that it relies on a (new) reduction from MinMC to MinMTC. Such a reduction was given in [5], but unfortunately the authors observed that it did not preserve planarity: we modify their reduction slightly and show that, in planar graphs with outer terminals, this new reduction does preserve planarity. To our best knowledge, such a planarity-preserving reduction was not known before.

For the sake of simplicity, all the (planar) graphs considered in this paper are *simple* (i.e., with no parallel edges), *loopless*, *connected* (if this is not the case, we consider each connected component independently), and already embedded in the plane without crossings. Moreover, any time we say that some (new) edge is *heavy* or that some (new) edge weight is equal to $+\infty$, we mean that the edge weight is equal to a sufficiently large integer, e.g., $1 + \sum_{e \in E} c(e)$, where E is the initial edge set.

2 Description of the algorithm

We now present our algorithm. The basic idea for it is simple: we wish to reduce, in polynomial time, a planar MinMC instance with outer terminals to a planar MinMTC instance (or, more precisely, to a set of planar MinMTC instances). Obviously, we also want the planar MinMTC instance(s) to have a fixed number of terminals (as the MinMC instance does). Then, we can use any algorithm solving MinMTC in planar graphs in polynomial time for fixed $|\mathcal{T}|$, such as the ones in [5, 12], to solve our initial instance of MinMC in polynomial time. Therefore, we begin by describing our planarity-preserving reduction.

2.1 A reduction that preserves planarity

In [5], a general reduction from MinMC to MinMTC was presented. More precisely, it was shown that any MinMC instance can be solved by solving at most $\frac{(\sqrt{2|\mathcal{N}|}+1)^{2|\mathcal{N}|}}{(\sqrt{2|\mathcal{N}|}+1)!}$ MinMTC instances. Indeed, in any optimal solution $\mathcal{S}$ for the initial MinMC instance $\mathcal{I}$, the $2|\mathcal{N}|$ terminals are clustered in $q \leq \sqrt{2|\mathcal{N}|}+1$ sets (or *clusters*), such that (a) each cluster does not contain both s_i and s'_i for each i, (b) each cluster is separated from all the other clusters by the edges of $\mathcal{S}$, and (c) for each pair of clusters, there is an i such that s_i is in one cluster and s'_i is in the other (otherwise, we can merge the two clusters and still have a valid clustering, i.e., one that satisfies (a) and (b)). Given this clustering, for each cluster, we add one new vertex (called a *cluster vertex*) linked by a new heavy edge (called a *cluster edge*) to every terminal contained in this cluster in order to transform the MinMC instance $\mathcal{I}$ into a MinMTC instance $\mathcal{I}'$ where the terminals are the cluster vertices. We have $opt(\mathcal{I}) = opt(\mathcal{I}')$ (where, given any instance $\mathcal{J}$, $opt(\mathcal{J})$ is the optimal value for $\mathcal{J}$). Indeed, $opt(\mathcal{I}') \leq opt(\mathcal{I})$ since $\mathcal{S}$ is a solution for $\mathcal{I}'$, and $opt(\mathcal{I}) \leq opt(\mathcal{I}')$ since any solution for $\mathcal{I}'$ is a solution for $\mathcal{I}$. Hence, whenever $|\mathcal{N}|$ is fixed and such instances of MinMTC are polynomial-time solvable, one can solve in polynomial time the corresponding instances of MinMC by trying all the possible clusterings and, for each one of them, solving a MinMTC instance. The main drawback of this technique is that it does not necessarily preserve planarity (as it can be easily seen).

We now define another reduction, which is in fact a refinement of the previous one: in any optimal solution of a MinMC instance, any terminal belongs to a connected component of the graph obtained by removing the edges of this solution. Therefore, the idea is to consider only clusterings associated with these connected components. In other words, for any optimal solution $\mathcal{S}$, we define the clustering associated with (the connected components of) $\mathcal{S}$ as follows: $\mathcal{T}_i$, the ith cluster of the clustering, contains the terminals lying in the ith connected component. We shall see that this particular clustering, which satisfies (a) and (b), has nice properties. However, since it does not necessarily satisfy (c), it can contain more than $\sqrt{2|\mathcal{N}|}+1$ clusters (actually, it can contain $\Omega(|\mathcal{N}|)$ clusters), and so our reduction needs to enumerate more clusterings than the general reduction described above. Before proving that our reduction preserves planarity, we make some preliminary remarks and give some definitions.

2.2 Obtaining a 2-vertex-connected graph

We consider an arbitrary instance of planar MinMC. Without loss of generality, we first assume that all the terminal vertices are distinct (if this is not the case, it is quite an easy job to make this assumption hold). Let us show that we can also assume without loss of generality that the graph of this instance is 2-vertex-connected (which will imply that the boundary of the outer face is a unique cycle). Indeed, we can transform the graph into a 2-edge-connected graph by doubling all edge weights (and thus, all weights become even), and then by replacing every edge (u, v) (with weight $c(u, v)$) by four edges $(u, w), (w, v), (u, w')$ and (w', v), where w and w' are two new vertices, of respective weights $\frac{c(u,v)}{2}, +\infty, \frac{c(u,v)}{2}$ and $+\infty$ (hence, we double the value of any solution, and so, in particular, the optimal value). Then, we replace any *articulation vertex* (i.e., any vertex whose removal disconnects the graph) by a cycle with heavy edges (let us call such a cycle an *articulation cycle*), containing one vertex for each edge that was adjacent to the initial vertex. Eventually, we make the ith edge initially adjacent to the articulation vertex adjacent to the ith vertex of the articulation cycle. In order to preserve planarity, we must do this in such a way that the order of the edges corresponds to the order that one gets, for instance, by visiting these edges clockwise (assuming that the graph is embedded in the plane without crossings). If there was a terminal t_i on the articulation vertex, then in this new graph we will make t_i lie on one of the vertices of the corresponding articulation cycle that are on the outer face (there are two such vertices for each inclusion-wise maximal 2-vertex-connected component where t_i lies in the initial graph). This transformation, which is well-defined (the articulation cycle being a cycle of length at least 4) and can be done in linear time, is illustrated in Figure 1.

We also need to number the vertices on the outer face. The numbering of a vertex v will be denoted by $n(v)$. To do this numbering, start from an arbitrary vertex of the outer face (which shall thus be the vertex numbered 1), then number the other vertices $2, 3, \ldots$, by following the order obtained by visiting the outer face clockwise. (This way, since all terminal vertices are distinct, no two of them have the same number.) To simplify the notations, given two vertices u and v lying on the outer face such that $n(u) < n(v)$, we denote by $p(u, v)$ the vertices of the chain linking u to v on the outer face (i.e., the vertices w of the outer face such that $n(u) \leq n(w) \leq n(v)$).

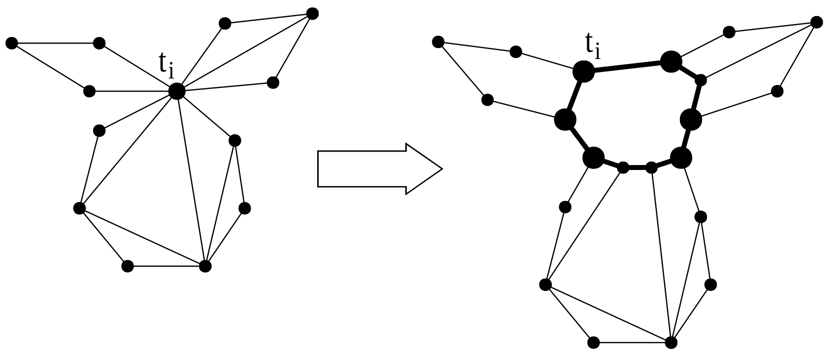

Figure 1: Transforming an articulation vertex into an articulation cycle (the big black vertices are the ones of the new cycle that lie on the outer face).

Sometimes, we shall even abuse notation and write $p(u, v)$ for this chain itself.

2.3 Proof of the algorithm

In this section, we prove our main result. We begin with a useful lemma:

Lemma 1. *Given a planar* MinMC *instance with outer terminals, any clustering of the terminals associated with an optimal solution of this instance is such that, for any pair of clusters $\mathcal{T}_i$ and $\mathcal{T}_j$ with $i \neq j$, for any pair of terminals v_i and v'_i of $\mathcal{T}_i$ (with $n(v_i) < n(v'_i)$) and for any pair of terminals v_j and v'_j of $\mathcal{T}_j$ (with $n(v_j) < n(v'_j)$), we have:*

$$p(v_i, v'_i) \cap p(v_j, v'_j) = \emptyset \text{ or } p(v_i, v'_i) \subset p(v_j, v'_j) \text{ or } p(v_j, v'_j) \subset p(v_i, v'_i). \quad (1)$$

Proof. Take any optimal solution for this MinMC instance, and define v_i, v'_i, v_j and v'_j as above. First, we have $v_i \neq v_j$ (for instance), so we cannot have $p(v_i, v'_i) = p(v_j, v'_j)$. Moreover, if we had $p(v_i, v'_i) \cap p(v_j, v'_j) \neq \emptyset$, $p(v_i, v'_i) \not\subseteq p(v_j, v'_j)$ and $p(v_j, v'_j) \not\subseteq p(v_i, v'_i)$, then this would imply w.l.o.g. that the four vertices v_i, v'_i, v_j, v'_j are such that $n(v_i) < n(v_j) < n(v'_i) < n(v'_j)$. However, since the solution we consider is optimal, and since v_i and v'_i (resp. v_j and v'_j) belong to the same cluster (of the clustering associated with this solution), they belong to the same connected component once the edges of the solution have been removed. Hence, there does exist a chain between v_i and v'_i and a chain between v_j and v'_j. By assumption, these two chains must be vertex-disjoint (otherwise, the four vertices would be in the same connected component), but, by the planarity of the graph and the fact that the terminals lie on the outer face, this is not possible (see [17] for a complete proof of this fact). Lemma 1 follows. $\square$

This lemma shows that the clusterings we are interested in correspond, in fact, to *laminar families* [13, Chapter 2.2] of the set of vertices lying on the outer face: actually, the set to be considered for each cluster $\mathcal{T}_i$ are the vertices of $p(u_i, v_i)$, where $n(u_i) = \min_{w \in \mathcal{T}_i} n(w)$ and $n(v_i) = \max_{w \in \mathcal{T}_i} n(w)$. It is known that any laminar family can be represented by a tree: here, the terminals are the leaves, each other node corresponds to a cluster, and the father of a node, associated with a set of terminals, is the smallest cluster that strictly contains all these terminals.

It is then easy to show that, when adding cluster vertices and edges to any clustering satisfying Property (1), we obtain a planar MinMTC instance with a bounded number of terminals (one for each cluster). Indeed, it suffices to show that, given such a clustering, we can add the cluster vertex and edges corresponding to one of the "minimal" clusters $\mathcal{T}_i$ (and then remove $\mathcal{T}_i$ from the list of the clusters) in such a way that the graph remains planar and the terminals lying in all the other clusters are still on the outer face: by induction, this will yield the desired MinMTC instance. Here, "minimal" means that $p(u_i, v_i)$ contains only terminals in $\mathcal{T}_i$, where $n(u_i) = \min_{w \in \mathcal{T}_i} n(w)$ and $n(v_i) = \max_{w \in \mathcal{T}_i} n(w)$, and such a cluster always exists when (1) holds. One way of doing this is to draw a curve from u_i to v_i (assuming $|\mathcal{T}_i| \geq 2$, since otherwise there is nothing to do) lying on the outer face (let us call it a *cluster curve*), in such a way that this curve is homotopic to $p(u_i, v_i)$ with respect to the outer face (i.e., it can be continuously transformed into $p(u_i, v_i)$ without being blocked by the boundary of the outer face when doing so). Then, choose any point on this curve, and make the cluster vertex lie on it. Finally, link all the terminals in $\mathcal{T}_i$ to this point (vertex) by curves (edges), such that no two curves are crossing and all the curves are contained in the plane region bounded by $p(u_i, v_i)$ and the cluster curve (this can be easily done). The obtained graph is as required. This may be seen as actually constructing the tree associated with the laminar family corresponding to the clustering.

It is worth noticing that the planar MinMTC instance we obtain when adding cluster vertices and edges does *not* necessarily have outer terminals: hence, we cannot use the algorithm given in [4]. Also notice that Property (1) is necessary (for a clustering to be associated with an optimal solution of a MinMC instance), but it is clearly not sufficient.

To summarize our approach, we only need to enumerate all the clusterings satisfying (a) and containing at most $2|\mathcal{N}|$ clusters (since the graph obtained by removing the edges of any optimal multicut cannot have more than $2|\mathcal{N}|$ connected components), and, for each one of them, to check if Property (1) holds (since otherwise, by Lemma 1, this clustering cannot be associated with an optimal solution). If it does hold, we compute the best solution associated with this clustering by reducing the initial MinMC instance to a planar MinMTC instance.

Therefore, our algorithm for MinMC is as follows:

- For each clustering satisfying (a) and containing at most $2|\mathcal{N}|$ clusters:
  - If the current clustering does not satisfy Property (1), reject it;
  - Otherwise, transform the instance into a planar MinMTC instance, and solve this new instance by using your favorite algorithm (e.g., one of the two given in [5, 12]). Keep the solution obtained only if it is better than the best one found until now;
- At the end of the algorithm, output the best computed solution.

The running time of the step where we check whether Property (1) holds being small compared to the one needed to solve the MinMTC instance (more details are given in the next section), the whole running time is bounded by the number of potential clusterings (a constant, since $|\mathcal{N}|$ is fixed) times the running time of the algorithm used to solve the MinMTC instance (which is polynomial [5, 12]). Therefore, we have proved:

Theorem 1. MinMC *is polynomial-time solvable in planar graphs with outer terminals, if the number of terminals is fixed.*

It can be noticed, in particular, that this includes the cacti, and even the outerplanar graphs, with a fixed number of terminals.

2.4 Some remarks on the time complexity

In this section, we discuss some implementation details for the algorithm described in the previous section.

First, while checking whether (a) holds can easily be done in $O(|\mathcal{N}|)$ time, checking whether a given clustering satisfies Property (1) can be done in $O(|\mathcal{N}|^4)$ time (it means that this time is dominated by the running time of Hartvigsen's algorithm, which is $O(|\mathcal{N}|4^{|\mathcal{N}|}n^{2|\mathcal{N}|-4}\log n)$, and which is better than the one of Dahlhaus et al.'s algorithm). Indeed, we have to consider every pair of clusters $\mathcal{T}_i$ and $\mathcal{T}_j$ (there are $O(|\mathcal{N}|)$ clusters, and thus $O(|\mathcal{N}|^2)$ pairs to be considered) and every combination of pairs of vertices v_i, v_i' of $\mathcal{T}_i$ and of pairs of vertices v_j, v_j' of $\mathcal{T}_j$. A naive implementation leads to a running time of $O(|\mathcal{N}|^6)$. However, it is not difficult to show that, given two clusters $\mathcal{T}_i$ and $\mathcal{T}_j$, it is sufficient to consider the pairs $v_i, v_i' \in \mathcal{T}_i$ and $v_j, v_j' \in \mathcal{T}_j$ such that there is no w satisfying (i) $w \in \mathcal{T}_i$ and $n(v_i) < n(w) < n(v_i')$ or (ii) $w \in \mathcal{T}_j$ and $n(v_j) < n(w) < n(v_j')$, since, if we had $n(v_i) < n(v_j) < n(v_i') < n(v_j')$ and if such a w existed, this would yield (iii) $w \in \mathcal{T}_i$ and either $n(v_i) < n(v_j) < n(w) < n(v_j')$ or $n(w) < n(v_j) < n(v_i') < n(v_j')$, or (iv) $w \in \mathcal{T}_j$ and either $n(v_i) < n(w) < n(v_i') < n(v_j')$ or

$n(v_i) < n(v_j) < n(v'_i) < n(w)$: as a consequence, we could replace one of the four terminals v_i, v'_i, v_j, v'_j by w.

So, actually, only $O(|\mathcal{N}|)$ pairs of terminals have to be considered in $\mathcal{T}_i$ (the pairs of terminals v_i, v'_i with $n(v_i) < n(v'_i)$ such that there is no $w \in \mathcal{T}_i$ satisfying $n(v_i) < n(w) < n(v'_i)$) and only $O(|\mathcal{N}|)$ pairs of terminals have to be considered in $\mathcal{T}_j$. Finally, given $v_i, v'_i \in \mathcal{T}_i$ and $v_j, v'_j \in \mathcal{T}_j$ (with $n(v_i) < n(v'_i)$ and $n(v_j) < n(v'_j)$), deciding whether v_i, v'_i, v_j, v'_j satisfy Property (1) can be done in constant time by checking whether $n(v_i) < n(v'_i) < n(v_j) < n(v'_j)$, or whether $n(v_j) < n(v'_j) < n(v_i) < n(v'_i)$, or whether $n(v_i) < n(v_j) < n(v'_j) < n(v'_i)$, or whether $n(v_j) < n(v_i) < n(v'_i) < n(v'_j)$. This implies that $O(|\mathcal{N}|^2 \cdot |\mathcal{N}| \cdot |\mathcal{N}|) = O(|\mathcal{N}|^4)$ comparisons are needed to decide whether Property (1) holds or not.

As already observed, this enables us to conclude that the running time is dominated by the time needed to enumerate all the clusterings times the complexity of the algorithm used to solve the MinMTC instances. Hartvigsen's algorithm is theoretically the most efficient, but in fact the time complexities of the two algorithms differ only by small factors, and both are huge (although polynomial) even for reasonable values of k. It would be worth improving them.

It should be noticed that we assume here that the enumeration of all the valid clusterings is done as described in Section 2.3, i.e., by enumerating all the clusterings containing at most $2|\mathcal{N}|$ clusters, and then checking, for each one of them, whether Property (1) holds. However, it may be possible that specific algorithms for enumerating laminar families of a ground set (based, for instance, on the relationship between laminar families and trees) result in better running times for this enumeration step. In any case, as we already pointed out, the bottleneck in the complexity of our algorithm is the time needed to solve the MinMTC instances.

Finally, the running time of our algorithm can be improved to linear if there are only two terminal pairs (recall that the minimum cut problem is linear-time solvable in planar graphs if the source and sink lie on the outer face [4], and can be solved in $O(n \log n)$ time otherwise [5]). In this configuration, removing the edges of any optimal solution leaves two or three connected components, so we have to try both possibilities. Indeed, let us prove that it cannot leave four components. Assume it can. Then, for $i \in \{1, 2\}$, let V_i and V'_i denote the set of vertices of the connected component containing s_i and s'_i respectively. For each i, we know that any edge of this solution adjacent to a vertex of V_i has to be adjacent to a vertex of V'_i (and vice-versa), since otherwise this edge would be useless, and so would not belong to an optimal solution. Therefore, no edge adjacent to a vertex in $V_1 \cup V'_1$ is adjacent to a vertex in $V_2 \cup V'_2$: this is not possible, because we assumed the initial graph to be connected.

In the case of two components, the associated clustering can be either $\{\{s_1, s'_2\}, \{s'_1, s_2\}\}$ or $\{\{s_1, s_2\}, \{s'_1, s'_2\}\}$: denote it by $\{\{a, b\}, \{c, d\}\}$. The four terminals lie on the outer face in the order a, b, c, d, since they must satisfy Property (1). Hence, the instance can be reduced to a planar minimum cut instance with outer source and sink (by linking a new source to a and b and a new sink to c and d), and solved in linear time.

In the case of three connected components, the associated clustering can be $\{\{s_1\}, \{s_2\}, \{s'_1, s'_2\}\}$, $\{\{s'_1\}, \{s'_2\}, \{s_1, s_2\}\}$, $\{\{s_1\}, \{s'_2\}, \{s'_1, s_2\}\}$ or $\{\{s'_1\}, \{s_2\}, \{s_1, s'_2\}\}$. Denote this clustering by $\{\{a\}, \{b\}, \{c, d\}\}$. Without loss of generality, the four terminals lie on the outer face in the order a, b, c, d or a, c, b, d. If they lie in the order a, b, c, d, the instance can be reduced to a planar MinMTC instance with three terminals, all lying on the outer face (by linking c and d to a new terminal), and solved in linear time [4]. (Alternatively, it can also be solved in linear time as a planar minimum cut instance with outer source and sink, separating a, b from c, d.) Otherwise, an optimal multicut consists of two cuts: one separating a from b, c, d, the other separating b from a, c, d. These two cuts are necessarily disjoint, since otherwise they would partition the graph into four connected components, which, as was already pointed out, is not possible. Hence, the instance can be solved by solving two planar minimum cut instances with outer source and sink: the first instance is obtained by linking b, c, d to a new sink (a being the source), the second one by linking a, c, d to a new sink (b being the source). The optimal multicut is then the union of these two simple cuts, and can thus be obtained in linear time.

3 Open problems

A lot of problems related to ours are still open. For instance, we leave as open the complexity of MinMC in planar graphs when $|\mathcal{N}|$ is fixed. We also leave as open the problem of designing an FPT [6] algorithm for the problem considered in this paper (our algorithm is clearly not FPT, since the ones given in [5, 12] are not; however, it would be if one of them was).

Another question to consider is whether there exists a polynomial-time approximation scheme (PTAS) for planar MinMTC: the problem is known to be $\mathcal{NP}$-hard [5], but not APX-hard. However, although MinMTC is tractable in k-outerplanar graphs (since, according to [5], it can be solved in polynomial time in bounded tree-width graphs), it seems that the general framework for designing PTASs developed in [1] and based on a decomposition of the planar graph into a set of k-outerplanar graphs (one gets optimal solutions for the k-outerplanar graphs, and then combines them into a single solution) cannot be easily applied to MinMTC, since gluing the pieces together does not always yield a multiterminal cut for the whole graph.

Acknowledgments

The author thanks the anonymous referees for their useful remarks and comments.

References

[1] B.S. Baker. Approximation algorithms for NP-complete problems on planar graphs. J. ACM 41 (1994) 153–180.

[2] C. Bentz. On the complexity of the multicut problem in bounded tree-width graphs and digraphs. Disc. Appl. Math. 156 (2008) 1908–1917.

[3] G. Călinescu, C.G. Fernandes and B. Reed. Multicuts in unweighted graphs and digraphs with bounded degree and bounded tree-width. Journal of Algorithms 48 (2003) 333–359.

[4] D.Z. Chen and X. Wu. Efficient algorithms for k-terminal cuts on planar graphs. Algorithmica 38 (2004) 299–316.

[5] E. Dahlhaus, D.S. Johnson, C.H. Papadimitriou, P.D. Seymour and M. Yannakakis. The complexity of multiterminal cuts. SIAM Journal on Computing 23 (1994) 864–894.

[6] R.G. Downey and M.R. Fellows. Parameterized Complexity (1999). Springer-Verlag. New York.

[7] L.R. Ford and D.R. Fulkerson. Maximal flow through a network. Canadian Journal of Mathematics 8 (1956) 339–404.

[8] A. Frank. Edge-disjoint paths in planar graphs. J. Comb. Theory, Series B 39 (1985) 164–178.

[9] N. Garg, V.V. Vazirani and M. Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. SIAM Journal on Computing 25 (1996) 235–251.

[10] N. Garg, V.V. Vazirani and M. Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees. Algorithmica 18 (1997) 3–20.

[11] N. Garg, V.V. Vazirani and M. Yannakakis. Multiway cuts in node weighted graphs. Journal of Algorithms 50 (2004) 49–61.

[12] D. Hartvigsen. The planar multiterminal cut problem. Discrete Applied Mathematics 85 (1998) 203–222.

[13] B. Korte and J. Vygen. Combinatorial Optimization. Volume 21, Algorithms and Combinatorics (2000). Springer-Verlag.

[14] R. Möhring, D. Wagner and F. Wagner. VLSI network design: a survey. M.O. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser (editors), Handbooks in Operations Research/Management Science, Volume on Networks (1995). Elsevier.

[15] H. Okamura and P.D. Seymour. Multicommodity flows in planar graphs. Journal of Combinatorial Theory, Series B 31 (1981) 75–81.

[16] J.S. Provan and R.C. Burk. Two-connected augmentation problems in planar graphs. Journal of Algorithms 32 (1999) 87–107.

[17] N. Robertson and P.D. Seymour. Graph minors VI: disjoint paths across a disc. J. Combinatorial Theory, Series B 41 (1986) 115–138.

[18] É. Tardos and V.V. Vazirani. Improved bounds for the max-flow min-multicut ratio for planar and $K_{r,r}$-free graphs. Inform. Process. Lett. 47 (1993) 77–80.

[19] M. Yannakakis, P. Kanellakis, S. Cosmadakis and C. Papadimitriou. Cutting and partitioning a graph after a fixed pattern. Proceedings ICALP, Lecture Notes in Computer Science 154 (1983) 712–722.