An FPT Algorithm for Planar Multicuts with Sources and Sinks on the Outer Face

Cédric Bentz

AN FPT ALGORITHM FOR PLANAR MULTICUTS WITH SOURCES AND SINKS ON THE OUTER FACE

Cédric Bentz*

Abstract

Given a list of k source-sink pairs in an edge-weighted graph G, the *minimum multicut problem* consists in selecting a set of edges of minimum total weight in G, such that removing these edges leaves no path from each source to its corresponding sink. To the best of our knowledge, no non-trivial FPT result for special cases of this problem, which is **APX**-hard in general graphs for any fixed $k \geq 3$, is known with respect to k only. When the graph G is planar, this problem is known to be polynomial-time solvable if $k = O(1)$, but cannot be FPT with respect to k under the *Exponential Time Hypothesis*.

In this paper, we show that, if G is planar and in addition all sources and sinks lie on the outer face, then this problem does admit an FPT algorithm when parameterized by k (although it remains **APX**-hard when k is part of the input, even in stars). To do this, we provide a new characterization of optimal solutions in this case, and then use it to design a "divide-and-conquer" approach: namely, some edges that are part of any such solution actually define an optimal solution for a polynomial-time solvable multiterminal variant of the problem on some of the sources and sinks (which can be identified thanks to a reduced enumeration phase). Removing these edges from the graph cuts it into several smaller instances, which can then be solved recursively.

Keywords: Multicuts, Planar graphs, FPT algorithms.

1 Introduction

Given a list of k pairs (source s_i, sink s'_i) in an undirected edge-weighted graph G, the *minimum multicut problem* (MinMC) consists in selecting a set of edges of minimum total weight in G, in such a way that removing these edges leaves no path between s_i and s'_i for each i. As the weight $w(e)$ of each edge e is commonly assumed to be a rational number, we can actually assume without loss of generality that each $w(e)$ is an integer (by multiplying all $w(e)$'s by one sufficiently large integer).

*CNAM & CEDRIC Laboratory, 292 rue Saint Martin, 75003 Paris (France). Phone: +33 (0) 1 58 80 86 14. E-mail address: cedric.bentz@cnam.fr

A well-known special case of MinMC is the *minimum multiway cut problem*, or *minimum multiterminal cut problem* (MinMTC): in any instance of this problem, we are given a set of *terminals* $\mathcal{T} = \{t_1, \ldots, t_{|\mathcal{T}|}\}$, and the source-sink pairs in the associated MinMC instance are (t_i, t_j) for $i \neq j$.

We shall only consider undirected graphs here. When $k = 1$, MinMC (which is then equivalent to MinMTC with $|\mathcal{T}| = 2$) turns into the famous *minimum cut problem*, and can therefore be solved in polynomial time. Moreover, MinMC remains polynomial-time solvable when $k = 2$ [14]. However, MinMTC is **APX**-hard for any fixed value of $|\mathcal{T}| \geq 3$ [8]: note that, when $|\mathcal{T}| = 3$, MinMTC is actually a special case of MinMC with $k = 3$.

When k is part of the input, MinMC is tractable in chains, but **APX**-hard even in stars with weights 1 [10], and hence also in planar graphs where all sources and sinks lie on the outer face. However, when $k = O(1)$, it becomes tractable in trees, and even in graphs of bounded tree-width [1].

Some results are known about the parameterized complexity of MinMC. For instance, it is known to be FPT with respect to the solution size [4, 13], but we are not aware of any non-trivial FPT result when the parameter to be considered is k. Recall that a problem parameterized by some parameter p is FPT with respect to p if it admits an FPT algorithm with respect to p, i.e., an algorithm solving it in time $O(f(p)n^c)$, where $f(\cdot)$ is some computable function of p, n is the input size, and c is a constant independent of p [9].

Let us now turn to the case where G is planar. On the one hand, when all sources and sinks lie on the outer face, it was proved that, unlike MinMC, MinMTC can be solved in polynomial time, even when $|\mathcal{T}|$ is part of the input [5]. On the other hand, when sources and sinks can lie anywhere, it was proved in [12] that, under the *Exponential Time Hypothesis* (ETH), MinMTC cannot be FPT with respect to $|\mathcal{T}|$ in planar graphs. Hence, under the same hypothesis, MinMC cannot be FPT with respect to k in these graphs. However, when $k = O(1)$, it was proved that MinMC is polynomial-time solvable in planar graphs if all sources and sinks lie on the outer face [2], and later it was proved that MinMC remains polynomial-time solvable in planar graphs even when sources and sinks can lie anywhere [3, 6]. (It was already known for MinMTC in planar graphs when $|\mathcal{T}| = O(1)$ [8].)

In the present paper, we prove the first non-trivial FPT result concerning MinMC parameterized by k only, and at the same time settle the last case left open by the results shown in [2, 3, 6, 12]. Namely, we show that MinMC is FPT with respect to k when G is planar and all sources and sinks lie on its outer face. In order to do this, we provide a new characterization of optimal solutions for MinMC in such graphs, on which our algorithm is based.

In [2], it was proved that any MinMC instance in such a graph can be reduced to a set of MinMTC instances *in planar graphs* (i.e., where sources and sinks can lie anywhere). Actually, a limited number of configurations were enumerated, and for each configuration *one* planar MinMTC instance was solved (using a non-FPT algorithm, such as the one in [8]).

Here, we prove a stronger result: there exist optimal multicuts such that some part (i.e., some of the edges) of such a solution actually defines an optimal solution for a MinMTC instance, obtained in the same graph by removing some of the sources and sinks (or, equivalently, by keeping only some of them). (In practice, determining the sources and sinks that belong to this MinMTC instance requires some enumeration, but fortunately it can be done in FPT time.) In the MinMTC instance obtained in this way, *all* terminals lie on the outer face (and hence we can use the polynomial-time algorithm given in [5]). Moreover, removing the edges of the optimal solution for this MinMTC instance cuts the initial graph into several pieces (i.e., connected components), which can then be solved recursively as smaller MinMC instances satisfying the same assumptions as the initial one.

The proposed algorithm is thus based on a *divide-and-conquer* approach: we enumerate a limited (but larger than in [2], as we will need to "guess" slightly more information) number of configurations, and for each one we solve *a set* of planar MinMTC instances (and not *one* planar MinMTC instance anymore), in which, unlike in [2], *all* terminals lie on the outer face. This enables us to obtain a nearly linear-time algorithm when $k = O(1)$.

When considering any planar MinMC (or MinMTC) instance, loops and parallel edges are useless (loops can be removed, and edges having the same endpoints can be merged into a single edge, whose weight is the sum of the weights of the merged edges), and connected components actually define independent instances, so we shall assume without loss of generality that the input graph is connected and contains neither loops nor parallel edges, and that it is already embedded in the plane without crossings (and with all the sources/sinks or terminals lying on the outer face, if needed).

Furthermore, as in [2] (where all the necessary details are provided), we assume without loss of generality that all terminals are distinct and that the input graph is 2-vertex-connected (which can easily be achieved in linear time by doubling all edge weights, then obtaining a 2-edge-connected graph, and finally replacing any articulation vertex by a cycle). This means, in particular, that the boundary of the outer face is a simple cycle.

2 A reformulation using clusterings

The starting point of our FPT algorithm is basically the same as in [3]: in any connected graph (planar or not), removing the edges of any optimal multicut yields several connected components, each of them containing at least one source or sink. The sources/sinks belonging to a same connected component define a *cluster*. We shall call such a set of clusters a *clustering* (a clustering is thus a partition of the sources and sinks), and we shall say that the considered solution *induces* these clusters (and the connected components containing them), or equivalently this clustering.

Hence, finding an optimal multicut is equivalent to finding a set of edges of minimum total weight that isolates all the clusters of the clustering induced by this optimal multicut. By definition, this clustering is such that no cluster contains both s_i and s'_i for each i. In practice (i.e., from an algorithmic point of view), since we do not know this clustering as long as we do not know the optimal multicut itself, we need to enumerate all possible clusterings in order to ensure that the one induced by the optimal solution we are looking for will be considered as well.

The number of possible clusterings only depends on the number k of source-sink pairs: in other words, it is FPT with respect to k (we shall give more details later). This immediately implies that MinMC can be reduced in FPT time (with respect to k) to the following problem, called the *minimum multi-cluster cut problem* (MinMCC): given k' sets (or clusters) of terminals $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_{k'}$ in an edge-weighted graph G, find a set of edges of minimum total weight in G, in such a way that removing these edges leaves no path between any vertex in $\mathcal{T}_i$ and any vertex in $\mathcal{T}_j$, for any $i \neq j$. When $|\mathcal{T}_i| = 1$ for each i, MinMCC simply turns into MinMTC. Also note that MinMCC is actually a special case of MinMC, in which the source-sink pairs are (u, v), for each $i \neq j$ and each $u \in \mathcal{T}_i$ and $v \in \mathcal{T}_j$.

In the remainder of this paper, we shall focus on solving MinMCC in FPT time (with respect to k'), and this will immediately enable us to solve MinMC in FPT time (with respect to k) as well. Indeed, it is not hard to see that the above-mentioned reduction from MinMC to MinMCC is actually an FPT-reduction, as we have $k' \leq 2k$. One can even show a sharper bound on k': we have actually $k' \leq k + 1$, and this bound is *tight* (to see this, simply consider a chain with $2k$ vertices, in the order $s_1, s'_1, s_2, s'_2, \ldots, s_k, s'_k$, where the only optimal multicut induces $k + 1$ clusters). This will not have any significant impact on the asymptotic running time of our algorithm, but we give a short proof of this fact anyway, for the sake of completeness.

Assume by contradiction that, in a given instance of MinMC, there exists an optimal multicut inducing at least $k+2$ connected components. Any edge of this optimal multicut lies between two of these connected components, and there must exist an i such that one component contains s_i and the other s'_i (otherwise, this edge would be useless). Hence, when adding to the (at least) $k + 2$ connected components induced by such an optimal multicut the edges lying between the two connected components containing s_1 and s'_1 (if there exist such edges), we obtain at least $k + 1$ connected components. We do the same for s_2 and s'_2 (adding edges of the optimal multicut only if they have not already been added so far), and then for each s_i and s'_i for i from 3 to k. In the end, we have *all* the edges from the initial graph, and we have reduced by *at most* k the number of connected components. Therefore, there remain at least two connected components, which contradicts the fact that the initial graph was connected.

3 A characterization using planar duality

Let us now consider planar duality. It is well-known that, to any planar graph G embedded in the plane, one can associate a *dual* planar graph G^*. More precisely, to any face in G corresponds a vertex (called a *dual* vertex) in G^*, and any (*dual*) edge between two dual vertices corresponds to the edge (or one of the edges, if there are more than one) shared by the corresponding faces in G. This also holds for the outer face of G. Furthermore, since G is 2-vertex-connected, there exists no edge belonging to only one face.

Given an optimal solution S for a MinMCC instance in a planar graph G, we shall denote by S^* the set of dual edges corresponding to S in the dual graph G^*, and, for each i, by V_i the set of vertices of the ith connected component induced by S, and by $S_i \subseteq S$ the set of edges of G having exactly one endpoint in V_i. It is well-known that S_i^*, the set of dual edges corresponding to each S_i, is a set of (non necessarily simple) cycles in G^*.

Moreover, if we look at the embedding of G^* as a set of curves in the plane (which intersect at the dual vertices), then each S_i^* is represented by a set of closed curves, denoted by C_i^*. Each S_i^* is actually composed of one or several simple cycles $\{S_i^{*1}, S_i^{*2}, \dots\}$, i.e., each C_i^* is composed of one or several simple closed curves $\{C_i^{*1}, C_i^{*2}, \dots\}$. By the *Jordan curve theorem*, each such simple closed curve divides the plane into an interior region and an exterior region (a *region* being a set of points such that any two of these points can be linked by a curve without crossing any closed curve): intuitively, the interior region is the region of the plane that *is enclosed by* (or that *lies inside*) this closed curve. The C_i^*'s thus partition the plane into several regions. Among all these regions, there is one and only one that is unbounded (and one and only one C_i^* is *associated with it*, i.e., contains all the curves adjacent to this region). Actually, the C_i^*'s may be seen as defining the boundary of the V_i's, that form a partition of the vertex set of G, and hence the interior regions of any two distinct simple closed curves that compose them cannot overlap (except if one lies inside the other).

The following lemma summarizes well-known facts for planar MinMCC:

Lemma 1 ([3, 6, 8])**.** *Given a* MinMCC *instance in a planar graph G, any optimal solution S for this instance satisfies the following properties:*

(1) *for each $C_i^* = \{C_i^{*1}, C_i^{*2}, \dots\}$ and for any $j_1 \neq j_2$, either the interior regions of $C_i^{*j_1}$ and $C_i^{*j_2}$ are disjoint, or one lies inside the other,*

(2) *each $C_i^* = \{C_i^{*1}, C_i^{*2}, \dots\}$, except the one associated with the unbounded region, contains one simple closed curve, say C_i^{*1}, such that all the vertices of V_i lie inside C_i^{*1}, but not inside C_i^{*j} for any $j \geq 2$,*

(3) *for each C_i^*, except the one associated with the unbounded region, and for each $j \geq 2$, C_i^{*j} lies inside C_i^{*1}, and, for each $j_1 \geq 2$ and $j_2 \geq 2$ with $j_1 \neq j_2$, the interior regions of $C_i^{*j_1}$ and $C_i^{*j_2}$ are disjoint.*

Proof. Each of these three properties has been more or less explicitly proved or used for solving MinM(T)C or MinMCC in planar graphs in [3, 6, 8]. Property (1) is clear from [8], and we will justify the other two briefly.

Concerning Property (2), it was noticed in [8] for MinMTC (and it can easily be extended to MinMCC) that, except for the C_i^* associated with the unbounded region, each C_i^* must enclose a region containing the terminals in $\mathcal{T}_i$. By definition, this region lies inside either the interior region or the exterior region associated with each C_i^{*j}. Obviously, it cannot lie inside several of these interior regions: indeed, from Property (1), for any two of these interior regions, either they are disjoint (and thus it is clearly not possible), or one lies inside the other (and thus this other one is useless). Therefore, for each i, the region containing the terminals in $\mathcal{T}_i$ lies inside one of these interior regions (the one associated with C_i^{*1}) and outside all the other interior regions (or, equivalently, inside all the other exterior regions).

Concerning Property (3), it is sufficient to notice that, from Property (2), the only way for the interior region of C_i^{*1} to intersect the exterior region of C_i^{*j} for any $j \geq 2$ is to have C_i^{*j} lying inside C_i^{*1} for any $j \geq 2$. Moreover, this also implies that for any $C_i^{*j_1}$ and $C_i^{*j_2}$ with $j_1 \geq 2$, $j_2 \geq 2$ and $j_1 \neq j_2$, the interior region of one cannot lie inside the interior region of the other, and hence, from Property (1), they must be disjoint. □

For the special case considered here, we can prove the following lemma:

Lemma 2. *Given a* MinMCC *instance in a planar graph G where all the terminals lie on the outer face, any optimal solution S is such that each S_i^{*j} contains the dual vertex corresponding to the outer face of G, and, for each i, any two S_i^{*j}'s have only this vertex in common.*

Proof. We begin by proving the first part of the statement. If some S_i^{*j} did not contain the dual vertex corresponding to the outer face of G, then the interior region of C_i^{*j} would not enclose any terminal (as any terminal lies on the outer face of G), and hence it would be useless in an optimal solution.

Moreover, it is easy to see that, for each i, any two S_i^{*j}'s have at most one vertex in common, even in the case where the terminals can lie anywhere in the planar graph (and hence, in our special case, they have exactly one vertex in common). Indeed, if two S_i^{*j}'s had two or more vertices in common, then they could not belong to the boundary of a single region. □

Lemma 2 implies, in particular, that, if the input graph is planar and any terminal lies on the outer face, then *any* C_i^*, including the one associated with the unbounded region, actually consists of a *single* closed curve (that may not be simple). Let us assume without loss of generality that the closed curve associated with the unbounded region is C_1^*. From Lemma 1, we call a cluster $\mathcal{T}_i$ with $i \geq 2$ a *top cluster* if there is no $j \geq 2$ with $j \neq i$ such that C_i^{*1} lies inside C_j^{*1}. (Besides, $\mathcal{T}_1$ will be referred to as a top cluster as well.)

This notion can be interpreted in the initial graph G as well. For each i and each j, let S_i^j be the set of edges in G associated with S_i^{*j}. Removing from G the edges of any S_i^1 for $i \geq 2$ yields two connected components: one that contains the vertices in $\mathcal{T}_i$ (and possibly vertices from other clusters), and one that does not. We shall denote the former one by V_i': we have $V_i \subseteq V_i'$ for each i. Then, $\mathcal{T}_i$ with $i \geq 2$ is a top cluster if V_i' is not contained in any V_j' for $j \geq 2$ and $j \neq i$. In other words, the vertices of any V_j' with $j \geq 2$ such that $\mathcal{T}_j$ is not a top cluster are included in some V_i' with $i \geq 2$ and $i \neq j$. As such, any V_i' such that $\mathcal{T}_i$ is a top cluster and $i \geq 2$ can be viewed as a maximal inclusion-wise connected component among the V_j''s.

However, this notion is still not strong enough to state our main result. We refine it as follows. Take *any* top cluster except $\mathcal{T}_1$ (say, $\mathcal{T}_2$), and define it as a *good* top cluster. Then, we define the other good top clusters iteratively: any top cluster $\mathcal{T}_i$ such that S_i^{*1} with $i \geq 2$ has at least one edge in common with S_j^{*1} for some good top cluster $\mathcal{T}_j$ with $j \geq 2$ will also be defined as a good top cluster. (Besides, $\mathcal{T}_1$ will be defined as a good top cluster as well.)

All the previous notions are illustrated in Figure 1, where the terminals are the small black rectangles, while the other vertices are the small black circles. The clusters are numbered from 1 to 9, and any terminal is labeled by the number of the cluster it belongs to. The dual vertices and edges associated with the optimal solution drawn in Figure 1 are respectively the small grey diamonds and the grey dashed lines. The top clusters are numbered 1, 2, 5, 7 and 8, and the good top ones are numbered 1, 2 and 5 (another possible choice would be the ones numbered 1, 7 and 8). Moreover, the big grey diamond is the dual vertex associated with the outer face, and the three S_2^{*j}'s are indicated on the associated dual edges, as well as some other S_i^{*j}'s.

Our main result is the following lemma:

Lemma 3. *Assume we are given a* MinMCC *instance I in a planar graph G where any terminal lies on the outer face, and let S be an optimal solution for I. Then, $\bigcup_{i:i \text{ is a good top cluster}} S_i^1$ is an optimal solution for the* MinMTC *instance I' obtained as follows: (i) the input graph G' is the graph G without the edges in $S \setminus \left(\bigcup_{i:i \text{ is a good top cluster}} S_i^1\right)$, and with one terminal for each good top cluster, and (ii) all the terminals lie on the outer face.*

Proof. Consider the planar graph G' as defined above, and assume that the embedding of G' is computed by taking the one of G and then simply removing the edges in $S \setminus \left(\bigcup_{i:i \text{ is a good top cluster}} S_i^1\right)$. Clearly, if, for each good top cluster $\mathcal{T}_i$, we add in G' a new vertex (called a *cluster vertex*) linked by an edge (called a *cluster edge*) having a sufficiently large weight to each terminal of $\mathcal{T}_i$ (assume for now that it can be done in such a way that (ii) holds), then, by the definitions of S and G', removing the edges in $\bigcup_{i:i \text{ is a good top cluster}} S_i^1$ leaves no path between any two cluster vertices. In other words, $\bigcup_{i:i \text{ is a good top cluster}} S_i^1$ is actually a feasible solution to I'.

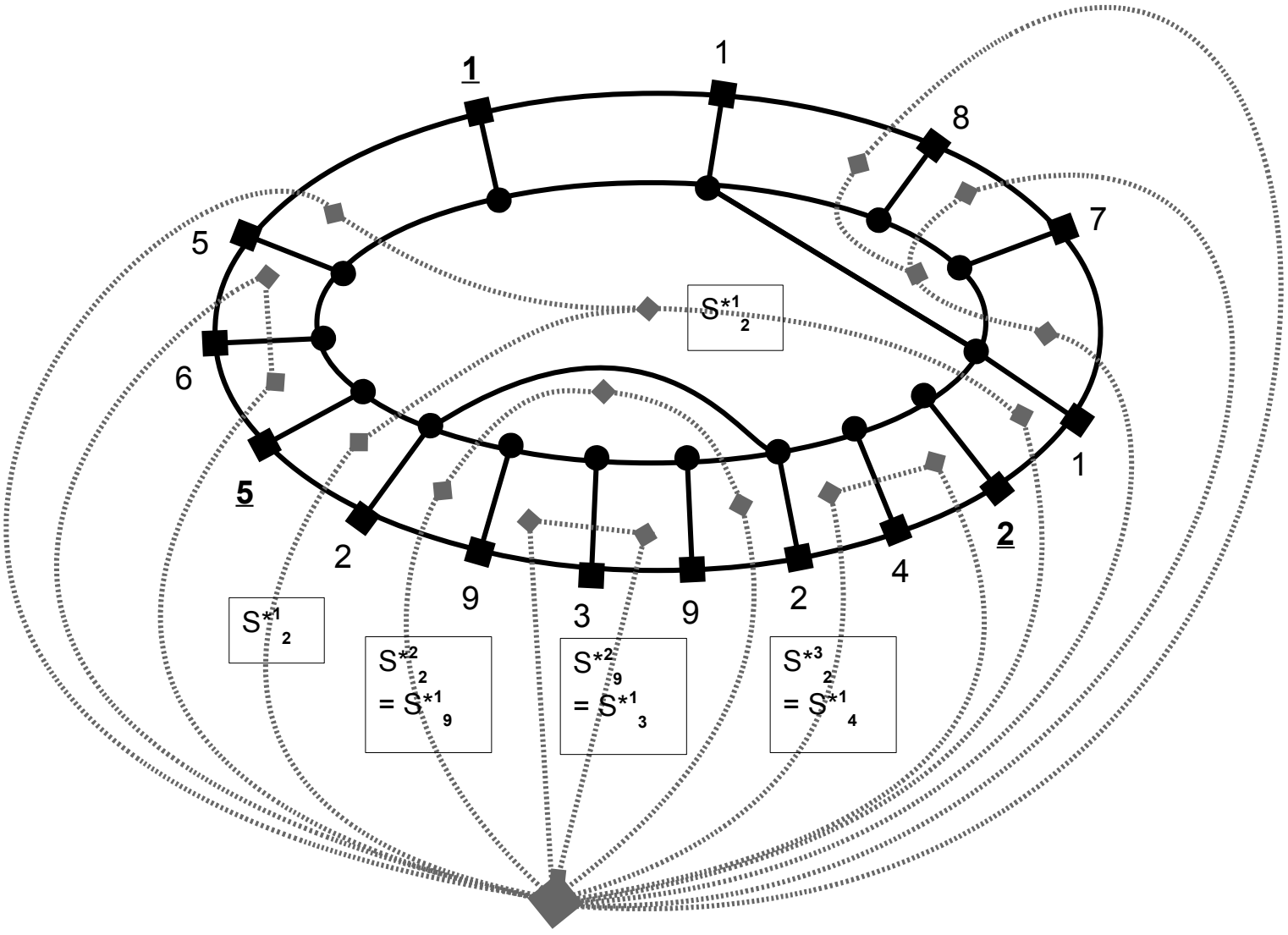

Figure 1: A planar MinMCC instance and the associated optimal solution, whose edges have weight 1 (while all the other edges have large weights).

Moreover, any edge in $\bigcup_{i:i \text{ is a good top cluster}} S_i^1$ lies between two connected components associated with two good top clusters. Indeed, on the one hand, by definition of a good top cluster, such an edge cannot belong to S_i^{*1} (and hence, from Property (3) in Lemma 1, to S_i^*) for some top cluster $\mathcal{T}_i$ which is not good. On the other hand, from Property (3) in Lemma 1 and the definition of a top cluster, a connected component associated with a non top cluster lies inside a closed curve of the form C_i^{*j} for some $j \geq 2$, where $i \geq 2$ is such that $\mathcal{T}_i$ is a top cluster. From Lemma 2, the set of edges S_i^{*j} associated with such a closed curve has one and only one vertex in common with any other S_i^{*h}, and hence, in particular, S_i^{*j} shares no edge with S_i^{*1}.

This implies that in G' there is no path from any terminal of any good top cluster to any other terminal, except for the terminals of any other good top cluster. Therefore, replacing $\bigcup_{i:i \text{ is a good top cluster}} S_i^1$ in G' by any feasible solution to I', whose total weight does not exceed the one of $\bigcup_{i:i \text{ is a good top cluster}} S_i^1$ and that contains no cluster edge, yields another feasible solution to I in G which is at least as good as S. As S is an optimal solution to I, and as any optimal solution to I' contains no cluster edge (their common weight being too large), this implies in particular that $\bigcup_{i:i \text{ is a good top cluster}} S_i^1$ is an optimal solution to I'.

It remains to prove the last part of the lemma: namely, let us prove that we can add the cluster vertices and edges in such a way that (ii) holds.

To do this, we shall proceed in a way similar to the one described in [2]: the claim that the cluster vertices can then be assumed to lie on the outer face will simply come from the fact that, unlike in [2], there is not a cluster vertex associated with each cluster, but only with each good top one. To describe our way of achieving this, we shall need some additional definitions.

Clearly, from the definition of a top cluster, all the terminals of such a cluster, *except* $\mathcal{T}_1$, are consecutive on the outer face (among all the terminals of top clusters), as otherwise C_i^{*1} would lie inside C_j^{*1} for two distinct top clusters $\mathcal{T}_i$ and $\mathcal{T}_j$ with $i \geq 2$ and $j \geq 2$, which would be a contradiction. Therefore, if we go through the outer face of the graph G clockwise (which can be done in a well-defined way, as G is 2-vertex-connected), then, for each top cluster $\mathcal{T}_i$ of S with $i \geq 2$, there is a "first" terminal of this cluster that is encountered when doing so while staying inside C_i^{*1}. In other words, each such top cluster $\mathcal{T}_i$ has a unique terminal (which we shall call the *first* terminal of $\mathcal{T}_i$) from which we can encounter every other terminal of $\mathcal{T}_i$ by going through the outer face of the graph G clockwise and without leaving the interior region of C_i^{*1}. From this first terminal, we can then define a unique ordering of the other terminals of $\mathcal{T}_i$, which is simply the order in which they are encountered while going through this outer face clockwise.

We can define the first (and last) terminal of $\mathcal{T}_1$ in a similar way, i.e., as the unique terminal of $\mathcal{T}_1$ from which we can encounter every other terminal of $\mathcal{T}_1$ by going through the outer face of the graph G clockwise and without

entering the interior region of C_i^{*1} for any good top cluster $\mathcal{T}_i$ with $i \geq 2$. Such a first terminal exists, as otherwise it would mean that, among the terminals in the good top clusters, the terminals in $\mathcal{T}_1$ are not consecutive on the outer face. In other words, it would mean that, for any choice of a first terminal of $\mathcal{T}_1$, there is a good top cluster $\mathcal{T}_i$ with $i \geq 2$ (resp. another good top cluster $\mathcal{T}_j$ with $j \geq 2$ and $j \neq i$) whose terminals are encountered while going clockwise (resp. counterclockwise) from the first terminal of $\mathcal{T}_1$ to its last one on the outer face. However, S_i^{*1} and S_j^{*1} for such i and j could not share an edge (as otherwise either the first or the last terminal of $\mathcal{T}_1$ would lie inside a closed curve belonging to S, contradicting the definition of $\mathcal{T}_1$), which would contradict the fact that $\mathcal{T}_i$ and $\mathcal{T}_j$ are good.

Hence, the point of introducing the notion of good top clusters is to extend the consecutiveness property associated with top clusters, *even* when considering the cluster $\mathcal{T}_1$. In other words, all the terminals of *any* good top cluster are consecutive on the outer face among *all* the terminals of good top clusters. The notion of first vertices is illustrated in Figure 1, where the first vertex of each of the three good top clusters (numbered 1, 2 and 5) in the optimal solution to the considered instance is indicated as follows: the number of the corresponding cluster is underlined and written in bold.

Now we can proceed almost as in [2]. For each good top cluster $\mathcal{T}_i$ of S that contains at least two terminals (otherwise, there is nothing to do), we draw a curve (called a *cluster curve*) from the first terminal of $\mathcal{T}_i$ to its last one. Thanks to the consecutiveness property associated with good top clusters, these curves can easily be drawn in such a way that no two of them intersect, and the curve corresponding to each $\mathcal{T}_i$ must be *homotopic*, with respect to the boundary of the outer face of G, to the chain μ_i that goes clockwise from the first terminal of $\mathcal{T}_i$ to its last one, and uses only vertices lying on the boundary of this outer face. As in [2], being homotopic means that it can be continuously transformed into μ_i without being blocked by the boundary of this outer face while doing so (see also [7]).

Then, we let each cluster vertex lie on the associated cluster curve, and add the cluster edges, in such a way that they do not intersect and all lie inside the region bounded by the cluster curve and μ_i. This allows us to conclude that, after adding the cluster vertices and edges as above, all the cluster vertices do lie on the outer face, which ends the proof. □

4 Crafting the algorithm

We now focus on using the results from the two previous sections to come up with an algorithm solving MinMC in time FPT with respect to k.

Let I be an instance of MinMC in a planar graph where all sources and sinks lie on the outer face, and let S be an optimal multicut for I (S exists, even if we do not know it yet explicitly).

First of all, we can "guess" the clustering associated with S by enumerating *all* the possible clusterings containing at most $k+1$ clusters, and this can be done in time FPT with respect to k (see Section 2).

Then, we have to know the structure of the clusters in S, i.e., in particular, which clusters are the top ones, and which are the good top ones (see Section 3): we can "guess" this structure by using another enumeration (which, again, can be done in time FPT with respect to k).

Moreover, recall from the proof of Lemma 3 in Section 3 that, if we go through the outer face of the input graph G clockwise, then, for each good top cluster $\mathcal{T}_i$ of S with $i \geq 2$, there is a first terminal lying inside C_i^{*1} that is encountered while doing so (a similar notion of a first terminal holds for $\mathcal{T}_1$ as well). As mentioned in this proof, we will need to know this terminal in order to ensure that, in the planar MinMTC instance that we will construct, all the terminals will lie on the outer face.

Again, for each good top cluster $\mathcal{T}_i$, we can "guess" such a terminal by enumerating all the possibilities (i.e., trying the $|\mathcal{T}_i|$ terminals of $\mathcal{T}_i$ one by one). Once we know the first terminal of each such cluster, we can construct a planar MinMTC instance as explained in the proof of Lemma 3.

By solving the above-defined planar MinMTC instance (where all the terminals lie on the outer face), we obtain from Lemma 3 a set of edges S' that we can use to replace $\bigcup_{i:i \text{ is a good top cluster}} S_i^1$ in S. However, the main drawback of Lemma 3 is that it considers a MinMTC instance defined on a graph G' that is obtained from the input graph G, but that we do not know explicitly (as it would require to already know some part of an optimal solution). We now show how we can overcome this issue, by considering a *particular* optimal solution to the MinMCC instance we wish to solve:

Corollary 1. *Assume we are given a* MinMC *instance I in a planar graph G where all the sources and sinks lie on the outer face, and consider an optimal solution S for I that induces the maximum number of clusters. Then, $\bigcup_{i:i \text{ is a good top cluster}} S_i^1$ is an optimal solution for the* MinMTC *instance I' obtained as follows: (i) the input graph is G, with one additional terminal for each good top cluster, and (ii) all the terminals lie on the outer face.*

Proof. From Lemma 3, we just have to prove that, in this case (i.e., when we consider an optimal solution S to I inducing the maximum number of connected components), we do not have to know $S \setminus \left(\bigcup_{i:i \text{ is a good top cluster}} S_i^1\right)$ explicitly in order to define I'. In other words, that any optimal solution to I' does not interact with $S \setminus \left(\bigcup_{i:i \text{ is a good top cluster}} S_i^1\right)$, i.e., does not share any edge with it. If these sets of edges did intersect, then from the proof of Lemma 3 this would yield another optimal solution for I, that would induce more connected components than S does, contradicting the choice of S. □

Since the first step of our algorithm is to enumerate all the possible clusterings containing at most $k+1$ clusters, we will in particular consider the one associated with such an optimal solution S.

By solving the above-defined planar MinMTC instance with n vertices (where all the terminals lie on the outer face), which can be done in time $O(k^3n + k^2n\log n)$ thanks to the algorithm proposed in [5], we obtain from this corollary a set of edges S' that we can use to replace $\bigcup_{i:i \text{ is a good top cluster}} S_i^1$ in S, and, furthermore, that does not intersect $S \setminus \left(\bigcup_{i:i \text{ is a good top cluster}} S_i^1\right)$. Moreover, the graph of this instance is obtained from G simply by adding cluster vertices and edges, and hence we do not have to know $S \setminus \left(\bigcup_{i:i \text{ is a good top cluster}} S_i^1\right)$ explicitly. After removing S' from G, we obtain two or more connected components (and hence knowing $S \setminus \left(\bigcup_{i:i \text{ is a good top cluster}} S_i^1\right)$ explicitly or not is irrelevant).

These components, in turn, define smaller MinMCC instances, which can then be solved recursively by using the same strategy as above (without the first step, where we guessed the clustering), in a divide-and-conquer way. Each time an instance is solved, the number of connected components increases by at least one: as S contains at most $k+1$ such components, there is a total of at most k instances to be solved.

Putting all together, we obtain the following algorithm A_1, which makes a call to another algorithm, that will be detailed after A_1:

Algorithm A_1

Input: A connected planar graph G with n vertices, and a set of k source-sink pairs $(s_1, s'_1), \ldots, (s_k, s'_k)$ lying on the outer face of G.

Output: An optimal multicut for the input graph.

- For each clustering containing $2k$ terminals and $\leq k+1$ clusters do:
  - Build the associated planar MinMCC instance, where any terminal lies on the outer face of G, and the clusters are $\mathcal{T}_1, \mathcal{T}_2, \ldots,$
  - Run Algorithm $A_2(G, \{\mathcal{T}_1, \mathcal{T}_2, \ldots\})$, and store its output.
- Output the best feasible solution found.

Observe that the input of Algorithm A_1 is a MinMC instance, while the input of Algorithm A_2 will be a MinMCC instance. If a given clustering is induced by an optimal multicut but does not contain the maximum number of clusters, then the solution computed by A_1 *for this cluster* may not be optimal (which simply means that we need to consider another clustering).

Moreover, in Algorithm A_1, each call to Algorithm A_2 is actually the first call of a series of recursive calls. In other words, Algorithm A_2 is a recursive algorithm, that can be described as follows:

Algorithm A_2

Input: A connected planar graph G with n vertices, and a set $\{\mathcal{T}_1, \mathcal{T}_2, \dots\}$ of clusters of terminals, all lying on the outer face of G.

Output: An optimal multi-cluster cut for the input graph.

- For each possible choice of good top clusters among all the clusters of the clustering $\{\mathcal{T}_1, \mathcal{T}_2, \dots\}$, and for each possible choice of first terminals for all these good top clusters, do:
  - Construct and solve the associated planar MinMTC instance, where the terminals, that are the cluster vertices associated with the good top clusters, all lie on the outer face (see Corollary 1),
  - Remove from the input graph the edges of the optimal solution computed above, obtaining several connected components $G_1, G_2, \dots$, and then store these edges in the current solution,
  - For each of these connected components G_i that contains the terminals of at least two clusters, run Algorithm $A_2(G_i, \mathcal{T}(G_i))$, where $\mathcal{T}(G_i)$ is the set of clusters whose terminals belong to G_i, and then add the associated output to the current solution.
- Output the best feasible solution found.

Thanks to the above discussion, it should be clear that Algorithm A_1 is correct, and runs in time $O(f(k)n \log n)$, for some function $f(\cdot)$ to be specified, in graphs with n vertices. This running time is actually obtained by multiplying the different factors associated with the successive steps:

1. Enumerating all the possible clusterings containing $2k$ terminals and at most $k+1$ clusters incurs a factor $O\left(\frac{(k+1)^{2k}}{(k+1)!}\right)$, as noted in [8],
2. Enumerating all the possible good top clusters among the (at most $k+1$) clusters of a given clustering, and then all their possible first terminals (among $O(k)$), incurs a factor $O(k2^{k+1}) = O(k2^k)$,
3. Solving each planar MinMTC instance with all the terminals lying on the outer face can be done in time $O(k^2(kn + n \log n))$,
4. Finally, there are at most k such instances to solve.

The overall running time is thus $O\left(k^4 2^k \frac{(k+1)^{2k}}{(k+1)!}(kn + n \log n)\right)$, i.e., it is nearly linear when $k = O(1)$. Hence, we have proved:

Theorem 1. *In planar graphs where all sources and sinks lie on the outer face,* MinMC *is FPT with respect to the number k of source-sink pairs.*

5 Extensions and open problems

In this paper, we have provided an FPT algorithm for MinMC parameterized by the number k of source-sink pairs, in the case where the input graph is planar and all the sources and sinks lie on the outer face. This algorithm actually runs in $O(n \log n)$ time when $k = O(1)$, where n is the number of vertices of the input graph. In [2], it was proved that the time for solving this problem can be improved to linear when $k = 2$, but the proof cannot be generalized to greater values of k. Therefore, this set of results leaves as open the following question: does there exist a linear-time algorithm (i.e., running in time $O(n)$ for any $k = O(1)$) in this case?

Moreover, our FPT algorithm can easily be extended to a generalization of MinMC, called *partial* MinMC (or *k-multicut problem* [11]), which asks to select a minimum-weight set of edges whose removal leaves no path from s_i to s'_i, for at least a given number of source-sink pairs (s_i, s'_i). Indeed, partial MinMC can be reduced in FPT time to MinMC, by "guessing" the subset of source-sink pairs between which there will remain no path in an optimal solution (there are $O(2^k)$ such possible subsets to enumerate).

Let us now consider MinMCC. On the one hand, it is easy to see that MinMCC is polynomial-time solvable in general graphs with two clusters (by reducing it to the minimum cut problem), but Dahlhaus et al. (that call it the *colored multiterminal cut problem*) proved in [8] that it is **NP**-hard in planar graphs, even with only four clusters (and they claimed that this remains true with only three clusters). On the other hand, when the input graph is planar and has all its terminals lying on the outer face, MinMTC (the special case of MinMCC where clusters have size 1) is polynomial-time solvable, even when $|\mathcal{T}|$ is part of the input [5], and our FPT algorithm precisely solves MinMCC parameterized by the total number of terminals in such a graph (or, equivalently, parameterized both by the number of clusters *and* by the maximum number of terminals per cluster).

However, when the number of clusters is part of the input, we do not even know the complexity of MinMCC in such a graph. Observe that this question remains open even if there are $O(1)$ terminals in each cluster.

When the number of clusters is viewed as a parameter, one may hope that our approach is able to solve the problem even when there is an arbitrary number of terminals in each cluster, as we only need to "guess" the first terminal of each good top cluster. Unfortunately, this is true only if each cluster induces exactly one connected component in any optimal solution (see Corollary 1 and the discussion preceding its statement), and hence, when such a property does not hold, this question remains open as well.

References

[1] C. Bentz. On the complexity of the multicut problem in bounded tree-width graphs and digraphs. Discrete Applied Mathematics 156 (2008) 1908–1917.

[2] C. Bentz. A simple algorithm for multicuts in planar graphs with outer terminals. Discrete Applied Mathematics 157 (2009) 1959–1964.

[3] C. Bentz. A Polynomial-Time Algorithm for Planar Multicuts with Few Source-Sink Pairs. Proceedings IPEC (2012) 109–119.

[4] N. Bousquet, J. Daligault and S. Thomassé. Multicut is FPT. Proceedings STOC (2011) 459–468.

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

[6] É. Colin de Verdière. Multicuts in Planar and Bounded-Genus Graphs with Bounded Number of Terminals. Algorithmica 78 (2017) 1206–1224.

[7] É. Colin de Verdière and J. Erickson. Tightening Nonsimple Paths and Cycles on Surfaces. SIAM Journal on Computing 39 (2010) 3784–3813.

[8] 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.

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

[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] D. Golovin, V. Nagarajan and M. Singh. Approximating the k-multicut problem. Proceedings SODA (2006) 621–630.

[12] D. Marx. A Tight Lower Bound for Planar Multiway Cut with Fixed Number of Terminals. Proceedings ICALP (2012) 677–688.

[13] D. Marx and I. Razgon. Fixed-Parameter Tractability of Multicut Parameterized by the Size of the Cutset. SIAM Journal on Computing 43 (2014) 355–388.

[14] 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.