DISJOINT PATHS IN SPARSE GRAPHS*

Cédric Bentz†

Abstract

We generalize all the results obtained for maximum integer multiflow and minimum multicut problems in trees by Garg, Vazirani and Yannakakis [Primal-dual approximation algorithms for integral flow and multicut in trees, Algorithmica 18 (1997) 3–20] to graphs with a fixed cyclomatic number, while this cannot be achieved for other classical generalizations of trees. We also introduce the *k-edge-outerplanar* graphs, a class of planar graphs with arbitrary (but bounded) treewidth that generalizes the cacti, and show that the integrality gap of the maximum edge-disjoint paths problem is bounded in these graphs.

1 Introduction

In this paper, we are interested in the study of the maximum edge-disjoint paths and the minimum multicut problems in undirected graphs (no directed version is considered), as well as some of their variants. These two fundamental problems have been extensively studied, and are well-known to be $\mathcal{NP}$-hard even in very restricted classes of graphs.

Assume we are given an n-vertex m-edge undirected graph $G = (V, E)$, a *capacity function* $c : E \rightarrow \mathbb{Z}^+$ and a list $\mathcal{N}$ of pairs (source s_i, sink s'_i) of *terminal* vertices. Each pair (s_i, s'_i) defines a *net* or a *commodity*. The *maximum integer multiflow problem* (MaxIMF) consists in maximizing the number of flow units routed between the nets (each unit being routed between s_i and s'_i for some i), while enforcing the capacity constraints on the edges. When $c_e = 1$ for each $e \in E$, MaxIMF turns into the *maximum edge-disjoint paths problem* (MaxEDP). When each commodity is required to be routed along a single path, MaxIMF turns into the *maximum unsplittable flow problem* (MaxUSF).

The *minimum multicut problem* (MinMC) consists in selecting a minimum weight set of edges (the weight of edge e being $c(e)$) whose removal leaves no path between s_i and s'_i for each i. The *minimum multiterminal cut problem* (MinMTC) is a special case of MinMC in which, given a set of vertices $\mathcal{T} = \{t_1, \dots, t_{|\mathcal{T}|}\}$, the nets are (t_i, t_j) for $i \neq j$.

*A preliminary version of this paper appeared in [4].

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

For $|\mathcal{N}| = 1$, the powerful Ford-Fulkerson's theorem establishes that the value of the minimum cut is equal to the value of the maximum integral flow [20]. Unfortunately, this property does not hold for larger $|\mathcal{N}|$. However, MaxIMF and MinMC do have a fundamental relationship. Both can be expressed as integer linear programs, and the continuous relaxations of their linear programming formulations are dual. One consequence is that the value of any feasible multiflow cannot exceed the value of any multicut. This property explains why approximation results sometimes relate the value of an approximately optimal multiflow to the value of a well-suited feasible multicut, instead of relating it directly to the value of an optimal multiflow. Throughout the paper, when mentioning integrality gaps for these problems, we shall always mean integrality gaps *with respect to the classical linear programming formulations of the problems* (see [13, 24]).

A lot of work has been done on these problems. Although the basic problems are known to be $\mathcal{NP}$-hard for a long time, much effort has been done in two directions: first, identifying classes of graphs or special cases where the problems become tractable; second, obtaining good polynomial-time approximation algorithms for these problems, and, in particular, deriving good integer solutions from fractional solutions (i.e., finding solutions with a small integrality gap) or designing primal-dual schemes.

Both aspects are considered in this paper. Since we are looking for valuable cases, we begin by presenting the main known results. Given an optimization problem P and a real $\alpha > 0$, an *α-approximation algorithm for P* is a polynomial-time algorithm A that always outputs a feasible solution for P such that $\max_{I \text{ / } I \text{ is an instance of } P} \left\{ \frac{\mathrm{OPT}_I}{\mathrm{SOL}_A(I)}, \frac{\mathrm{SOL}_A(I)}{\mathrm{OPT}_I} \right\} \leq \alpha$, where OPT_I is the optimum value for the instance I of problem P and $\mathrm{SOL}_A(I)$ is the value of the solution given by A for the instance I of problem P.

Prior to the study of MaxEDP, lots of results concerned a basic $\mathcal{NP}$-complete problem, the *edge-disjoint paths problem* (EDP). Given an undirected graph and a list of nets, the problem is to decide whether it is possible to route *all* the nets along edge-disjoint paths. Obviously, whenever this decision problem is $\mathcal{NP}$-complete, MaxEDP is $\mathcal{NP}$-hard. However, solving EDP in polynomial time does not necessarily help us for dealing with MaxEDP efficiently. See [22] for an extensive survey on EDP.

On the negative side, Pfeiffer and Middendorf show that EDP remains $\mathcal{NP}$-complete even if the graph obtained by adding the edges (s_i, s'_i), $i \in \{1, \ldots, |\mathcal{N}|\}$, to the initial graph G, is planar [37] (however, if, in addition, we restrict the terminals to lie on a *bounded number of faces* of G, they prove that the problem becomes tractable). Moreover, Marx shows that EDP is $\mathcal{NP}$-complete in eulerian planar graphs with maximum degree bounded by 4 (by showing that it is $\mathcal{NP}$-complete in eulerian grids [36]), and Nishizeki et al. show that it is also $\mathcal{NP}$-complete in series-parallel graphs (i.e., in graphs with tree-width 2) [38].

On the positive side, Robertson and Seymour show that, when $|\mathcal{N}|$ is fixed, EDP is polynomial-time solvable in unrestricted graphs [44]. Moreover, extending a result of Okamura and Seymour [40], Frank shows that EDP is polynomial-time solvable in planar graphs, if all the terminals lie on the outer face and all the vertices not on the outer face have even degrees [21]. Note that the above class of graphs includes the planar graphs with all their vertices on the outer face, i.e., the *outerplanar graphs* (a subclass of the series-parallel graphs).

We turn back to MaxEDP. In their seminal paper, Garg, Vazirani and Yannakakis show that MaxEDP is polynomial-time solvable in trees [25]. However, they also show that, in trees with capacities 1 and 2, MaxIMF is $\mathcal{NP}$-hard and $\mathcal{APX}$-hard. By replacing each edge of capacity 2 by two parallel paths of length two, each containing only edges with capacity 1, this implies that MaxEDP is $\mathcal{NP}$-hard and $\mathcal{APX}$-hard in outerplanar graphs having all their *edges* lying on the outer face. (By replacing any edge (u, v) that does not lie in a cycle by a cycle (u, v, w, u) (w being a new vertex), and by adding two nets (u, w) and (v, w), we can show that MaxEDP remains $\mathcal{APX}$-hard even if the graph is also 2-edge-connected. Then, by replacing any vertex v whose removal disconnects the graph by a cycle (whose edges have "large" capacities and which has a number of vertices equal to the degree of v), and by letting the ith edge initially adjacent to v being linked to the ith vertex of the new cycle, we can show that MaxEDP remains $\mathcal{APX}$-hard even if the graph is also 2-vertex-connected.) Moreover, Even, Itai and Shamir show that, even if $|\mathcal{N}| = 2$, MaxEDP is $\mathcal{NP}$-hard in unrestricted graphs [19]. It can be noticed that, if $|\mathcal{N}|$ is fixed and the degrees of the vertices are bounded by a constant, then MaxEDP can be solved in polynomial time by calling a constant number of times the algorithm of Robertson and Seymour [44]. This is also true if we consider the problem of linking by edge-disjoint paths as many nets as possible (i.e., if we consider MaxUSF with unit capacities). However, to our best knowledge, *in planar graphs*, MaxEDP *remains open if* $|\mathcal{N}|$ *is fixed* (although the variant where one requires vertex-disjoint paths instead of edge-disjoint paths is known to be tractable [26]). Note that the case where $|\mathcal{N}| = 2$ and adding the edges (s_i, s'_i), $i \in \{1, \ldots, |\mathcal{N}|\}$, does not destroy planarity, is tractable [34].

We now look at approximability results. Some important ones are known for MaxEDP, MaxIMF and MaxUSF. In general graphs, there is an $O(\sqrt{n})$-approximation algorithm for MaxEDP and MaxUSF [13], although the stronger (and very recent) known inapproximability result is that both cannot be approximated within $(\log m)^{1/2-\epsilon}$ for every $\epsilon > 0$ [1]. For planar graphs, the approximation ratio is also $O(\sqrt{n})$, while only $\mathcal{APX}$-hardness is known. Both a greedy algorithm (denoted by *SPF*, for *Shortest Paths First*) and a rounding based algorithm achieve this ratio [3, 29]. Furthermore, it is important to note that there are families of planar graphs where the integrality gap is $\Theta(\sqrt{n})$ [25]. For more restricted classes of graphs, however, constant- or logarithmic-factor approximation algorithms are known: for MaxIMF, a 2-approximation algorithm in trees [25] and an $O(\log(|\mathcal{N}|)/\epsilon)$-approximation (resp. an $O(1/\epsilon)$-approximation) in graphs (resp. in planar graphs) where any multicut has a value at least $\epsilon \sum_{e \in E} c(e)$ [39]; for unit capacitated MaxUSF, a 3-approximation in trees of rings [18] and a 9-approximation in complete graphs [9]; for MaxEDP, an $O(1)$-approximation (resp. an $O(\log n)$-approximation) in densely embedded (resp. in high-diameter) and nearly eulerian planar graphs (including the two-dimensional mesh) [28, 30], an $O(\log^{10} n)$-approximation in graphs where any cut between any pair of vertices contains $\Omega(\log^5 n)$ edges [42], and an $O(F)$-approximation in graphs with *flow number* F (see [32] for details). Moreover, for high-capacitated networks (i.e., for graphs where all the capacities are $\Omega(\log n)$), an $O(1)$-approximation can be achieved for MaxIMF by randomized rounding techniques [41]. In expander graphs, a general result on the connectivity between pairs of vertices is given in [23]. In planar graphs where all capacities are at least two, a recent paper of Chekuri et al. proposes an $O(\log n)$-approximation algorithm for MaxIMF based on a continuous relaxation [11, 12]. When all capacities are at least four, they obtain an $O(1)$-approximation [14]. An even more recent result has been obtained by the same authors in [15]: they give an $O(\log n)$-approximation algorithm for MaxEDP in bounded tree-width graphs. Still, it can be noticed that few (good) approximation results are available, due to the noticeable difficulty to design good approximation algorithms for these problems.

Now, let us consider the MinMC problem. Garg, Vazirani and Yannakakis show that it is $\mathcal{NP}$-hard and $\mathcal{APX}$-hard even in unweighted stars, but that it can be solved in polynomial time in trees if $|\mathcal{N}|$ is fixed, and approximated within a factor of 2 otherwise [25]. Moreover, Dahlhaus et al. show that MinMTC (and thus MinMC) is $\mathcal{NP}$-hard in unrestricted graphs, even if $|\mathcal{N}| = 3$ [16]; in planar graphs, MinMTC is polynomial-time solvable if $|\mathcal{N}|$ is fixed [16, 27], and $\mathcal{NP}$-hard otherwise [16]. Nevertheless, the integrality gap for MinMC is $O(\log |\mathcal{N}|)$ in general graphs [24] and $O(1)$ in planar graphs [45], and there exist polynomial-time algorithms achieving these ratios. Furthermore, Călinescu et al. give a polynomial-time approximation scheme for MinMC in unweighted graphs of bounded tree-width and bounded degree, and show that dropping any of these three assumptions leads to $\mathcal{APX}$-hardness (instead of $\mathcal{NP}$-hardness only) [8].

In [25], Garg et al. give a primal-dual scheme showing, in particular, that the integrality gap for MaxIMF is at most 2 in trees, and exhibit an example showing that, even in planar graphs, this gap can be quite large in general. This raises the question of finding classes of graphs where this gap is small. Actually, there are several motivations to the present paper. First, trying to generalize the results of Garg et al., i.e., looking for classes of graphs that generalize the trees and where all (or a main part of) their results remain true, and trying to understand what makes these problems much easier on trees (is it a structural property? Or merely a key parameter that is small in trees?). Second, trying to identify a parameter (or some parameters) that makes MaxEDP tractable if we bound it (or them), and $\mathcal{NP}$-hard otherwise. And third, finding special cases generalizing the trees and specializing, in some sense, the example given in [25, page 17], and where the integrality gap remains bounded for MaxEDP. The first and the third motivations have been strongly inspired by the work of Garg et al., and the second motivation has revealed to be closely related to the first one.

A natural way of generalizing the trees is to consider graphs with bounded tree-width [43]. Another generalization is to consider planar graphs where the terminals lie on a fixed number of faces [37]. However, as mentioned above, Garg et al. have shown that MaxEDP remains $\mathcal{NP}$-hard and $\mathcal{APX}$-hard in outerplanar graphs, which have tree-width at most 2 [7], and in which the terminals all lie on one face (the outer one). In addition, their polynomial reduction remains valid even if we restrict ourselves to graphs having a bounded degree inside each 2-vertex-connected component.

Our first result is that *all* the results presented in [25] can be generalized, in some sense, to graphs with a fixed cyclomatic number (a tree being a graph with cyclomatic number 0). In particular, we prove that MaxEDP is polynomial-time solvable in such graphs, and that the integrality gap for MaxIMF is bounded by two times one plus the cyclomatic number. Although bounding the maximum degree and having all the terminals lying on one face do not lead to a bounded integrality gap for MaxEDP [25], our second main result is that the integrality gap for MaxEDP is bounded in k-outerplanar graphs having a bounded degree inside each 2-vertex-connected component. Such graphs obviously generalize the trees, but also specialize the example given in [25, page 17], where each degree is bounded by 3 and the graph is planar but not k-outerplanar. To prove this last result, we introduce the *k-edge-outerplanar* graphs, which form a subclass of the k-outerplanar graphs, and then we apply on a particular spanning tree the approximation algorithm given in [25]. We also consider the cacti, a class of graphs that generalize the trees of rings, and show that, in this case, we can bound the integrality gap for MaxIMF.

The paper is organized as follows. In Section 2, we give or recall some definitions and notions that will be needed in the next sections. In Section 3, we give a new approximation algorithm for MaxIMF. Then, in Section 4, we detail our results concerning graphs with a fixed cyclomatic number, showing how to generalize the work of Garg, Vazirani and Yannakakis. Finally, Section 5 deals with the integrality gap of MaxEDP in k-edge-outerplanar graphs.

2 Preliminary notations and definitions

2.1 Notations and classical definitions

A graph (or one of its components) is called 2-*vertex-connected* (resp. 2-*edge-connected*) iff for any two of its vertices there are at least two paths between them that do not share any vertices (resp. any edges). A *block* is an inclusionwise maximal 2-vertex-connected component of a graph.

Given $k \geq 1$, a k-outerplanar graph is a planar graph having an embedding with at most k *layers* of vertices, i.e., such that, after removing iteratively the vertices (and their adjacent edges) lying on the outer face at most k times, we obtain the empty graph [2]. In particular, an *outerplanar* graph (or *1-outerplanar* graph) is a planar graph containing at least one vertex and having an embedding with all its *vertices* lying on the outer face. The class of k-outerplanar graphs is very well-known to be an important class of planar graphs with bounded tree-width [7].

Now, let us define two other classes of graphs. Given two integers $k \geq 1$ and $d \geq 2$, the class of k-outerplanar graphs having a degree bounded by d inside each block will be denoted by $OPBID_{k,d}$. Given an integer $\gamma \geq 0$, the class of connected graphs $G = (V, E)$ with a cyclomatic number $\nu(G) = |E| - |V| + 1$ smaller than or equal to γ will be denoted by S_γ (these graphs being very *sparse* since $|E| \leq |V| - 1 + \gamma$). Note that each connected planar graph with at most γ internal faces is in S_γ (in particular, S_0 represents the trees), and that $\nu(G)$ is bounded in graphs G with bounded tree-width.

Given a graph G and a list of nets $\{(s_1, s'_1), \ldots, (s_{|\mathcal{N}|}, s'_{|\mathcal{N}|})\}$ on its vertices, we denote by P_i the set of elementary paths linking s_i to s'_i in G, for $i \in \{1, \ldots, |\mathcal{N}|\}$. Moreover, let $P_{\mathcal{N}} = \bigcup_{i\in\{1,\ldots,|\mathcal{N}|\}} P_i$. A *flow path* is a path carrying at least one unit of flow of any commodity. Note that all the graphs considered in this paper are *simple* (i.e., with no parallel edges), *loopless* and *connected* (if this is not the case, we consider each connected component independently).

Eventually, we need two simple notation rules. Given a multicut C and a multiflow F, we shall denote by $\|C\|$ and $\|F\|$ their respective *values*. Given a graph G and a subset R of the edge set of G, let $G \setminus R$ denote the graph obtained from G by removing all the edges in R from the edge set of G.

2.2 New notions

In this paper, we introduce the class of k-*edge-outerplanar* graphs, which has been inspired by the previously mentioned class of k-*outerplanar* graphs. Given $k \geq 1$, a k-edge-outerplanar graph is a planar graph having an embedding with at most k layers of edges, i.e., such that, after removing iteratively the edges lying on the outer face at most k times, we obtain a graph with no edge. In particular, an *edge-outerplanar* graph (or *1-edge-outerplanar* graph) is a planar graph containing at least one edge and having an embedding with all its *edges* lying on the outer face. We will detail in Section 5.1 the relationships between k-outerplanar and k-edge-outerplanar graphs. Note that the $2k \times N$ planar mesh ($N > 2k$) is both k-outerplanar and $(k+1)$-edge-outerplanar.

We also need to define the notion of *inside degrees*. Recall that the *degree* of a vertex is the number of vertices adjacent to it. Given a graph, one of its 2-vertex-connected components $2VCC$, and a vertex v of $2VCC$, the degree of v *inside* $2VCC$, denoted by $deg_{2VCC}(v)$, is the number of vertices lying in $2VCC$ that are adjacent to v. Note that a vertex can have a bounded inside degree and an unbounded degree (the converse being obviously false).

3 A simple approximation algorithm

Recall that the approximation ratio of the greedy algorithm *SPF* is $O(\min(\sqrt{m}, n^{2/3}))$ [10, 33], i.e., $O(\sqrt{n})$ in planar graphs. This simple algorithm iteratively routes the shortest available path in $P_{\mathcal{N}}$ (it was introduced in [31]). Moreover, there exist families of trees where this bound is reached. We give one family here. Start with a path $v_1, v_2, \ldots, v_{p+2}$ of length $p+1$. Then, add a path of length $p+1$ from v_i to s_i for each $i \in \{2, \ldots, p+1\}$. Eventually, let s_1 lie on v_1, let s'_1 lie on v_{p+2} and let s'_i lie on v_{i+1} for each $i \in \{2, \ldots, p+1\}$. This graph has $\Theta(p^2)$ vertices (i.e., $p = \Theta(\sqrt{n})$) and $p+1$ nets (i.e., $|\mathcal{N}| = p+1$), and the path from s_i to s'_i has length $p+2$, for each $i \in \{2, \ldots, p+1\}$. *SPF* routes (s_1, s'_1) (which has length $p+1$), while the optimal solution is to route $(s_2, s'_2), \ldots, (s_{p+1}, s'_{p+1})$. Furthermore, the graph is a tree and the new graph obtained by adding the $p+1$ edges (s_i, s'_i) is outerplanar. Note that this instance can easily be transformed into a non trivial one (i.e., such that the optimal value is neither $O(1)$ nor $\Theta(|\mathcal{N}|)$).

Hence, even for restricted classes of graphs, we have to look for better approximation algorithms. Given a connected graph G, several of our results use the same basic idea: computing a spanning tree of G in order to use the results given in [25] for trees. Since we shall use a new simple algorithm based on this idea several times, we give it here. It can be viewed as a primal-dual scheme containing three steps:

1. Compute a spanning tree T of G;
2. Use the primal-dual algorithm given in [25] which constructs an integer multiflow F_T and a multicut C_T for T such that $\|C_T\| \leq 2\|F_T\|$;
3. Build a multicut C_G for G satisfying $\|C_G\| \leq \alpha\|C_T\|$ for a fixed $\alpha > 0$.

At the end of this algorithm (let us call it *ST-GVY-WG*), we obtain an integer multiflow F_T and a multicut C_G such that $\|C_G\| \leq 2\alpha\|F_T\|$. Noticing that F_T is also feasible for G, this yields 2α-approximation algorithms for both MaxIMF and MinMC. Obviously, the first step may have to be done with some care, and the third one as well (even if our purpose is not to find the best possible α). Note that Step 3 is only relevant to prove the approximation ratio: if one is only interested in computing an approximately optimal flow, only the two first steps are needed. Also note that, to the best of our knowledge, this is the first attempt to generalize the constant bound of the maximum integer multiflow / minimum multicut theorem of Garg et al. in trees [25]. Indeed, the family of trees given at the beginning of this section has an $\Omega(\sqrt{n})$ flow number (since any path has length $\Omega(\sqrt{n})$) and the minimum multicut uses $O(\sqrt{n})$ edges, hence neither the results in [32] nor the results in [39] provide a constant bound.

An interesting feature of Algorithm *ST-GVY-WG* is that it can be used as a fast heuristic for MaxIMF in general graphs (in which case we can call this heuristic *MST-GVY*, since we do not need Step 3): we shall consider in Step 1 a *maximum spanning tree*, as in Section 4. Actually, since, on the one hand, *SPF* works quite well when there remain short paths in $P_{\mathcal{N}}$, and, on the other hand, *MST-GVY* can in fact be iterated several times, the following heuristic, parameterized by a small integer $\lambda \geq 0$, would probably be better than *MST-GVY*:

- While there remains a path of length at most λ in $P_{\mathcal{N}}$, run *SPF*;
- While there is an i such that there exists a path between s_i and s'_i:
  - Run *MST-GVY* on each connected component of the current graph (the graph with the current capacities);
  - Update the current capacities, and remove any edge whose remaining capacity is 0.

It would be interesting to test this heuristic on real-life or randomly generated instances, for different values of λ.

4 Graphs with a fixed cyclomatic number

In this section, we generalize the results of [25] from trees to graphs in S_γ. We prove that MaxEDP can be solved in polynomial time for graphs in S_γ, by solving $O((2^\gamma|\mathcal{N}|+1)^\gamma)$ instances on a set of trees. Then, given a graph G in S_γ, we show how to compute an integer multiflow F_G and a multicut C_G such that $\|C_G\| \leq 2(\gamma+1)\|F_G\|$, by using Algorithm *ST-GVY-WG*. Finally, we show that MinMC can be solved in $O(m^{2^\gamma|\mathcal{N}|})$ time for graphs in S_γ, which is polynomial in m if $|\mathcal{N}|$ is fixed. For the sake of simplicity, we do not systematically try to optimize the constants used in our analysis.

4.1 Solving MaxEDP

Garg et al. show that MaxEDP is polynomial-time solvable in trees. We use this result to design a polynomial-time algorithm solving MaxEDP in graphs of S_γ. Note that this result generalizes and unifies the only tractable cases known for MaxEDP, namely, trees and rings.

Let G be a graph in S_γ. We remove (at most) γ edges from G, so that the resulting graph is a spanning tree, by iteratively picking an edge from a block. Let these edges be $e_1, \ldots, e_\gamma$. The main idea is that, since γ is fixed, there is a bounded number of edges that has to be considered. For each one of these γ edges, we select either no path or one elementary path that crosses it, and remove this edge and all the other edges crossed by the (possibly) selected path. We have to be careful to select only compatible (i.e., edge-disjoint) paths: for instance, if we select a path p crossing e_i, we must also select p for e_j, $i \neq j$, if p crosses e_j. After we do this for the γ edges, we obtain a forest. We compute an optimal solution for MaxEDP in this forest by using the algorithm of Garg et al. [25]. Gathering the paths selected in this solution with the ones selected previously, we obtain a solution for MaxEDP in G. We repeat this procedure until each possible combination of the elementary paths crossing $e_1, \ldots, e_\gamma$ has been tried (recall that, in fact, for each of these γ edges, we also have to try the case where no path goes through it). Keeping the best of all these solutions, we obtain an optimal solution.

Our algorithm solves $O((|P_\mathcal{N}| + 1)^\gamma)$ instances of MaxEDP in a forest. Thus, γ being fixed, if $|P_\mathcal{N}|$ is polynomial in n and $|\mathcal{N}|$, our algorithm runs in polynomial time. The following lemma gives a bound on $|P_i|$ for each i[1]:

Lemma 1. *Given a graph G in S_γ and two vertices s_i and s'_i, the number of elementary paths $|P_i|$ linking s_i to s'_i in G is at most 2^γ.*

Proof. We proceed by induction on γ. For $\gamma = 0$, we have $|P_i| = 1$ (G is a tree). Assume this holds for $\gamma - 1$, $\gamma \geq 1$, and let us show it holds for γ. If $|P_i| = 1$, we are done. Otherwise, let v be the first vertex, encountered in any elementary path from s_i to s'_i, that lies in a block. We can assume w.l.o.g. that $v = s_i$ (if this is not the case, this assumption does not modify $|P_i|$). Thus, there are at least two edges, e_1 and e_2, adjacent to s_i and lying in a block. No elementary path from s_i to s'_i crosses both e_1 and e_2, hence there is an edge $e \in \{e_1, e_2\}$ such that at least half of the paths in P_i do not cross e. Moreover, if we remove e, we obtain a graph $G' \in S_{\gamma-1}$, and we can apply the induction hypothesis: there are at most $2^{\gamma-1}$ elementary paths between s_i and s'_i in G'. Hence, $|P_i| \leq 2 \cdot 2^{\gamma-1} = 2^\gamma$. Lemma 1 follows. □

Note that this result is tight: to see this, consider for instance a path of length γ with s_i and s'_i as endpoints. Then, replace each edge (u, v) of this

[1] We have not been able to determine whether this result is already known, but we give a short proof anyway for the sake of completeness.

path by a cycle (u, w, v, w', u). The obtained graph is in S_γ and satisfies $|P_i| = 2^\gamma$. Moreover, Lemma 1 implies that $|P_{\mathcal{N}}| \leq 2^\gamma |\mathcal{N}|$, and thus the algorithm given above runs in polynomial time. Hence:

Theorem 1. MaxEDP *is polynomial-time solvable for graphs in* S_γ.

Actually, MaxEDP is even FPT [17] for the pair of parameters $(|\mathcal{N}|, \gamma)$. It would be interesting to determine whether there exists an FPT algorithm for MaxEDP, if only the cyclomatic number is viewed as a parameter (clearly, our algorithm is not FPT in this case).

Moreover, recall that MaxEDP is $\mathcal{NP}$-hard even for $|\mathcal{N}| = 2$. Nevertheless, using the results in this section, one can state:

Theorem 2. *If* $|\mathcal{N}|$ *is fixed,* MaxEDP *is polynomial-time solvable in graphs whose cyclomatic number is* $O(\sqrt{\log n})$.

Proof. We use the above algorithm. Recall that we have to solve $O((2^\gamma|\mathcal{N}|+1)^\gamma)$ instances of MaxEDP in a forest, where γ is the cyclomatic number. Thus, if $\gamma = O(\sqrt{\log n})$ and $|\mathcal{N}|$ is fixed, we have to solve $O(n^{O(1)})$ instances, which is polynomial in n. $\square$

Note that Theorems 1 and 2 (and their analyses) also hold for the variant where one requires (internally) vertex-disjoint paths instead of edge-disjoint paths, since this problem is also polynomial-time solvable in trees [8].

4.2 Bounding the integrality gap for MaxIMF

MaxIMF is $\mathcal{NP}$-hard and $\mathcal{APX}$-hard for trees, and hence for graphs in S_γ. However, Garg et al. have shown that, given a tree T, one can compute in polynomial time an integer multiflow F_T and a multicut C_T such that $\|C_T\| \leq 2\|F_T\|$. In this section, we prove that, given a graph G in S_γ, one can compute in polynomial time an integer multiflow F_G and a multicut C_G such that $\|C_G\| \leq 2(\gamma+1)\|F_G\|$.

We use Algorithm *ST-GVY-WG* given in Section 3. All we have to do is to detail how to construct a spanning tree T for G (Step 1), and then, how to construct a multicut C_G such that $\|C_G\| \leq (\gamma+1)\|C_T\|$ (Step 3).

Step 1 proceeds as follows: we construct a maximum weight spanning tree of G, using a variant of Kruskal's algorithm [35]. In other words, for each $i \geq 1$, we iteratively pick an edge e_i having the minimum capacity among all the edges lying in blocks of $G \setminus \{e_1, \ldots, e_{i-1}\}$. This gives us a set of γ edges satisfying $c(e_1) \leq c(e_2) \leq \cdots \leq c(e_\gamma)$, and the graph $G \setminus \{e_1, \ldots, e_\gamma\}$ is a tree T.

In Step 2, we compute for T an integral multiflow F_T and a multicut C_T such that $\|C_T\| \leq 2\|F_T\|$. Eventually, in Step 3, we use C_T to construct a multicut C_G for G, and we let $F_G = F_T$. For each edge f_j in C_T, let $\lambda(f_j)$ be the largest i such that, before the edge e_i was removed from $G \setminus \{e_1, \ldots, e_{i-1}\}$,

f_j still lied in a block (and hence we have $c(f_j) \geq c(e_{\lambda(f_j)})$). If f_j does not lie in a block of G, let $\lambda(f_j) = 0$. Moreover, let $\lambda^* = \max_{f_j \in C_T} \lambda(f_j)$ and let $f_{j^*} \in C_T$ be such that $\lambda(f_{j^*}) = \lambda^*$. Then, let $C_G = C_T \bigcup \{e_1, \ldots, e_{\lambda^*}\}$.

First, let us prove that $\|C_G\| \leq (\gamma+1)\|C_T\|$. We have $\|C_G\| = \|C_T\| + \sum_{i=1}^{\lambda^*} c(e_i) \leq \|C_T\| + \lambda^* c(e_{\lambda^*}) \leq \|C_T\| + \lambda^* c(f_{j^*}) \leq \|C_T\| + \lambda^*\|C_T\| = (\lambda^*+1)\|C_T\|$. Note that the inequality $c(e_{\lambda^*}) \leq c(f_{j^*})$ comes from the definitions of λ^* and f_{j^*}, and the way e_{λ^*} has been chosen.

Second, let us show that C_G is indeed a multicut for G. In fact, all we have to prove is that, given an edge belonging to $\{e_{\lambda^*+1}, \ldots, e_\gamma\}$, there is no need to pick it in C_G, i.e., there exists a path in $T \setminus C_T$ linking its two endpoints. Let (a, b) be an edge belonging to $\{e_{\lambda^*+1}, \ldots, e_\gamma\}$. There exists a path between a and b in T, since T is a spanning tree of G. Thus, if there exists no path from a to b in $T \setminus C_T$, then necessarily the path from a to b in T contains an edge f belonging to C_T. This implies that, just before (a, b) was removed, f was lying in a block. Since $(a, b) \in \{e_{\lambda^*+1}, \ldots, e_\gamma\}$, we have a contradiction. We have $\lambda^* \leq \gamma$, and hence:

Theorem 3. *The gap between the optima of* MinMC *and* MaxIMF *is bounded by* $2(\gamma+1)$ *for the graphs in* S_γ*. Moreover, solutions for* MinMC *and* MaxIMF *achieving this ratio can be computed in polynomial time.*

Corollary 1. *The integrality gap of* MaxIMF *is bounded by* $2(\gamma+1)$ *for the graphs in* S_γ*. Moreover, a solution for* MaxIMF *achieving this ratio can be computed in polynomial time.*

Note that Theorem 3 applies to MaxUSF as well, since the solution computed by our method is feasible for MaxUSF. Also note that, in the analysis of Theorem 3, explicitly knowing that the spanning tree constructed in Step 1 is actually maximum weighted is not necessary (and knowing how it is constructed is sufficient). Moreover, we do not know whether the bound $2(\gamma+1)$ in this theorem is tight or not (obviously, this is the case for $\gamma = 0$ [25]). Figure 1 shows an example where a weaker bound holds.

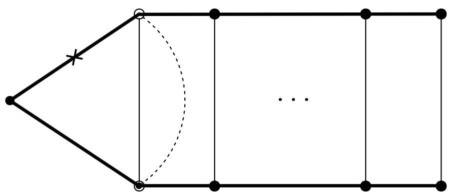


Figure 1: An example for Theorem 3 with one net (in dashed line). The edges in bold lines (i.e., the edges forming the spanning tree) have capacity $N+1$ for some $N > 0$, while all the other edges have capacity N. So, $\|F_T\| = N+1$, $\|C_T\| = N+1$ (the edge denoted by $\nrightarrow$) and $\|C_G\| = \gamma N + (N+1)$.

4.3 Solving MinMC

In this section, we detail results concerning MinMC. Recall that, in trees, Garg et al. show how to compute a multicut within twice the optimum (and even within twice the value of an integral multiflow). If we consider a graph G in S_γ and assume that all its edges have capacities bounded by a small integer β, we can construct a spanning tree T as in Section 4.1, compute a multicut C_T and an integer multiflow F_T, and build a multicut C_G for G by picking all the edges in C_T and the γ edges removed from G to obtain T. Obviously, C_G satisfies $\|C_G\| \leq \|C_T\| + \gamma\beta \leq 2\|F_T\| + \gamma\beta = (2+o(1))\|F_T\|$. This gives another generalization of the approximation result obtained in [25] for trees, which is different from the one given in Section 4.2, but which only applies to graphs with small capacities.

Moreover, Garg et al. have shown that, in trees, MinMC can be solved in polynomial time if $|\mathcal{N}|$ is fixed. The idea is that a multicut contains at most $|\mathcal{N}|$ edges, since there is one path from s_i to s'_i, for $i \in \{1, \dots, |\mathcal{N}|\}$: thus, MinMC can be solved in $O(m^{|\mathcal{N}|})$. For a graph G in S_γ, from Lemma 1, there are at most 2^γ paths between s_i and s'_i, for $i \in \{1, \dots, |\mathcal{N}|\}$. Hence, MinMC can be solved in $O(m^{2^\gamma|\mathcal{N}|})$ for the graphs in S_γ, which is polynomial in m if $|\mathcal{N}|$ is fixed.

Actually, a stronger result has been proved in [5], by using a completely different approach: MinMC is polynomial-time solvable in bounded tree-width graphs if $|\mathcal{N}|$ is fixed.

5 Integrality gap in k-edge-outerplanar graphs

In this section, we study the case of k-edge-outerplanar graphs. We first show that k'-outerplanar graphs having a degree bounded by d inside each block are closely related to these graphs.

5.1 Relationship between k-outerplanar graphs and k-edge-outerplanar graphs

The main result of this section is given in Theorem 4:

Theorem 4. *Any k-outerplanar graph such that the degree of each vertex is bounded by $d \geq 2$ inside each block is $(\lceil \frac{d}{2} \rceil + (k-1)\lfloor \frac{d}{2} \rfloor)$-edge-outerplanar. Moreover, any k-edge-outerplanar graph is k-outerplanar.*

Proof. The second part of Theorem 4 is obvious. We prove the first part by induction. Let G be a graph in $OPBID_{k,d}$, $k \geq 2$. For the proof, we can consider each block of G independently. Let B be a block of G with $|B| \geq 2$, i.e., an inclusionwise maximal 2-vertex-connected component of G containing at least two vertices. Each vertex of B lying on the outer face of G is adjacent to exactly two edges of B lying on the outer face. For each such vertex, we remove the corresponding two edges. We repeat this until each vertex of B lying on the outer face of G has at most one neighbor among the vertices lying in B. At each iteration, for each vertex v lying on the outer face and still having at least two neighbors among the vertices in B, we remove two edges adjacent to v, so we have to do it at most $\frac{d}{2}$ times if d is even. If d is odd, then we stop when the residual $deg_B(v)$ is at most one, so we have to do it at most $\frac{d-1}{2}$ times, i.e., at most $\lfloor \frac{d}{2} \rfloor$ times. After that, we obtain a component in $OPBID_{k-1,d}$. Eventually, for a graph in $OPBID_{1,d}$, we use the same technique. If d is even, then the analysis is similar. If d is odd, then we have to make the residual $deg_B(v)$ of each vertex v equal to 0, so we have to remove edges on the outer face $\lceil \frac{d}{2} \rceil$ times. (We also remove any edge that does not lie in a block.) Finally, any graph in $OPBID_{k,d}$ is $((k-1)\lfloor \frac{d}{2} \rfloor + \lceil \frac{d}{2} \rceil)$-edge-outerplanar. □

This theorem shows that, in order to be k-edge-outerplanar for some fixed k, it is sufficient for a graph to be in $OPBID_{k',d}$ for some fixed k' and d. However, it is not a necessary condition (every *Halin* graph, i.e., every planar graph with no vertex of degree 2 and whose edges are the disjoint union of a tree and a cycle connecting the leaves of this tree, is 2-outerplanar *and* 2-edge-outerplanar), and being only k'-outerplanar for some fixed k' is not sufficient in general to be k-edge-outerplanar for some fixed k (for any $p > 2$, the complete bipartite graph $K_{2,p}$ is $\lceil \frac{p}{2} \rceil$-edge-outerplanar and 2-outerplanar, the first layer having 4 vertices and the second one $p-2$).

Moreover, Figure 2 shows that the bound of Theorem 4 is tight. In Section 5.2, we consider k-edge-outerplanar graphs. Theorem 4 shows that our results will apply, in particular, to the graphs in $OPBID_{k',d}$.

5.2 Bounding the gap for MaxEDP

Recall that MaxEDP is $\mathcal{NP}$-hard and $\mathcal{APX}$-hard in edge-outerplanar graphs [25]. The main result of this section is that we can bound by a constant the integrality gap for MaxEDP in k-edge-outerplanar graphs. Before proving it, let us recall that Chekuri et al. have recently proved in [15] that the integrality gap for MaxEDP is $O(\log n)$ in bounded tree-width graphs. (It should be noticed that the work described in the present paper, which was already presented in a preliminary conference version in [4], was carried out before this result was announced.) However, when "specialized" to k-outerplanar graphs, they observed that their algorithm does not seem to yield a ratio better than $O(\log n)$. Here, we show:

Theorem 5. *The integrality gap for* MaxEDP *is bounded by* $4k$ *in* k*-edge-outerplanar graphs. Moreover, a solution for* MaxEDP *achieving this ratio can be computed in polynomial time.*

Proof. We use Algorithm *ST-GVY-WG*, given in Section 3. Let us describe Steps 1, 2 and 3. Given a k-edge-outerplanar (connected) graph $G = (V, E)$,

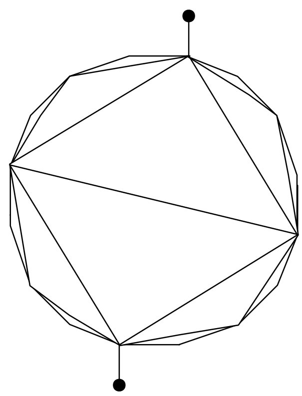


Figure 2: G_1, the skeleton of a family of tight graphs for Theorem 4 (d odd). (Any edge of G_1 lying on the outer face could be replaced by a path.) Each graph G_i, $i \geq 2$, is actually obtained from G_1 by replacing each edge by a copy of G_{i-1}, the big two vertices corresponding to the endpoints of this edge. Given $k > 0$, the graph G_k is both k-outerplanar and $((k-1)\lfloor \frac{d}{2} \rfloor + \lceil \frac{d}{2} \rceil)$-edge-outerplanar ($d = 7$ here).

Step 1 proceeds as follows: (i) for each layer L among the k layers of edges of G, (ii) for each internal face Φ, if there exist edges lying both on L and on the border of Φ, remove exactly one such edge. After part (ii) ends for the i^{th} layer ($i \leq k-1$), we obtain a $(k-i)$-edge-outerplanar connected graph. Hence, at the end of Step 1 (i.e., when part (i) ends), we obtain a spanning tree T of G. Then, F_T and C_T are obtained in Step 2. Eventually, we use C_T to construct C_G in Step 3.

For each edge in C_T, C_G will contain at most $2k$ edges, and hence $\|C_G\| \leq 2k\|C_T\| \leq 4k\|F_T\|$. The removal of any edge $(u, v) \in C_T$ separates the vertices of T (and hence the vertices of G) in exactly two connected components, V_u and $V_v = V \setminus V_u$. Let $\delta_G(u, v)$ be the set of edges between V_u and V_v in G, and let the *circuit boundary* of a block of G be the cycle delimiting this block (the circuit boundary of G is defined as the disjoint union of the circuit boundaries of all its blocks). We need the following lemma, showing that $|\delta_G(u, v)| \leq 2k$:

Lemma 2. *Given V_u and V_v in a k-edge-outerplanar graph G, $\delta_G(u, v)$ contains at most 2 edges on each one of the k layers. Moreover, it contains exactly 2 edges on the k^{th} layer iff they are on the circuit boundary of G.*

Proof. We proceed by induction on k. For $k = 1$, G is an edge-outerplanar graph. If (u, v) does not lie in a cycle of G, then $\delta_G(u, v)$ contains only (u, v) and we are done. Otherwise, by the way we construct T, there is a cycle of G containing both (u, v) and an edge not in T: $\delta_G(u, v)$ contains these 2 edges. This completes the case $k = 1$.

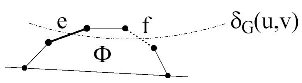
Assume now that Lemma 2 holds for $(k-1)$-edge-outerplanar graphs, $k \geq 2$. Let G be a k-edge-outerplanar (connected) graph, and let us prove it holds for G. So, let us consider V_u and V_v in G: recall that the edge (u, v) is an edge of the spanning tree obtained for G by applying Step 1 as described above. During this process, after applying part (i) only once (i.e., for only one layer, the one containing the edges of the outer face), we obtain a $(k-1)$-edge-outerplanar connected graph G' to which we can apply the induction hypothesis: in particular, at most 2 edges of the $(k-1)^{th}$ layer of G' are in $\delta_{G'}(u, v)$. Moreover, we have $\delta_{G'}(u, v) \subseteq \delta_G(u, v)$. We have to distinguish between three cases:

- If there is no edge on the $(k-1)^{th}$ layer of G' that belongs to $\delta_{G'}(u, v)$, then, obviously, for each edge e on the circuit boundary of G, there is a path linking its two endpoints and using only edges lying on the $(k-1)^{th}$ layer of G', i.e., using no edge in $\delta_{G'}(u, v)$. This path uses no edge from $\delta_G(u, v) \setminus \delta_{G'}(u, v)$ either, because such edges do not belong to G'. Hence, the endpoints of e both belong to V_u or V_v, so e does not belong to $\delta_G(u, v)$ (i.e., $\delta_{G'}(u, v) = \delta_G(u, v)$).

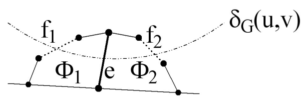
- If there is one edge (say, e) on the $(k-1)^{th}$ layer of G' that belongs to $\delta_{G'}(u, v)$, then, by assumption, e is not on the circuit boundary of G'. This means that e is a *bridge* of G' (an edge whose removal disconnects the graph), so $\delta_{G'}(u, v) = \{e\} = \{u, v\}$. Let us first assume that e lies in a block of G. If e is on the circuit boundary of G, then, by construction, there is an internal face Φ of G (adjacent to the outer face of G) whose border contains both e and an edge f not in G': hence, one endpoint of f is in V_u and the other is in V_v, so $\delta_G(u, v) = \{e, f\}$ (see Figure 3(a)). Otherwise, e belongs to the border of two internal faces of G, Φ_1 and Φ_2. By construction, Φ_1 (resp. Φ_2) is adjacent to the outer face and its border contains one edge f_1 (resp. f_2) not in G'. For each $i \in \{1, 2\}$, one endpoint of f_i is in V_u and the other is in V_v, so $\delta_G(u, v) = \{e, f_1, f_2\}$ (see Figure 3(b)). Note that if e does not lie in a block of G, then e is a bridge of G, so no edge from the circuit boundary of G belongs to $\delta_G(u, v)$ (i.e., $\delta_{G'}(u, v) = \delta_G(u, v) = \{e\} = \{u, v\}$).

- If there are two edges e_1 and e_2 on the $(k-1)^{th}$ layer of G' belonging to $\delta_{G'}(u, v)$, then, by assumption, they belong to the circuit boundary of G'. Hence, e_1 (resp. e_2) belongs to the border of an internal face Φ_1 (resp. Φ_2) of G, adjacent to the outer face of G and containing one edge f_1 (resp. f_2) not in G'. Edges f_1 and f_2 are distinct iff Φ_1 and Φ_2 are distinct (see Figures 3(c) and 3(d)). Hence, if f_1 and f_2 are distinct, then we have $\delta_G(u, v) = \delta_{G'}(u, v) \cup \{f_1, f_2\}$; otherwise, we have $\delta_G(u, v) = \delta_{G'}(u, v)$.

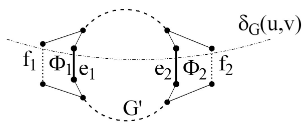
The proof of Lemma 2 is now complete. □



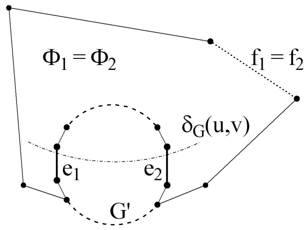
(a) e is on the circuit boundary of G.



(b) e is not on the circuit boundary of G.



(c) Φ_1 and Φ_2 are distinct.



(d) Φ_1 and Φ_2 are not distinct.

Figure 3: Illustrating the 4 main cases of Lemma 2.

We apply Lemma 2 for each edge in C_T, and this immediately implies $\|C_G\| \leq 2k\|C_T\| \leq 4k\|F_T\|$, as claimed. Note that the reason for processing Step 1 carefully is that, if T is not constructed as indicated, we will not be able to bound $|\delta_G(u,v)|$. For example, in Figure 1, assume that all the edges are valued by one and that the spanning tree constructed in Step 1 is the one in bold lines. Then, $\|C_T\| = 1$ although $\delta_G(u,v)$ contains all the edges in thin lines, so $\|C_G\|$ is unbounded. □

The last remark we shall make about the analysis of our algorithm is that it is tight: indeed, there exist instances where the cut C_G and the flow F_T computed by *ST-GVY-WG* are such that $\|C_G\| = 4k\|F_T\|$. Let us give one such family of instances here. We construct a graph G by using an *odd* number of copies of the complete bipartite graph $K_{2,d}$ (d even). For the ith copy of $K_{2,d}$, we denote by v_i and t_i its two vertices with degree d (the d other vertices having degree 2). We merge all the v_i's into a single vertex, v (and so we have $v_1 = v_2 = \cdots = v_i = \cdots = v$). Moreover, we define all the nets (t_i, t_j), $i < j$ (hence, we have $\mathcal{T} = \{t_1, \ldots, t_{|\mathcal{T}|}\}$). This graph is $\frac{d}{2}$-edge-outerplanar (since d is even). Furthermore, for any spanning tree T, $\|C_T\| = |\mathcal{T}| - 1$ and $\|F_T\| = \frac{|\mathcal{T}|-1}{2}$ (since $|\mathcal{T}|$ is odd). Finally, in each of the $|\mathcal{T}| - 1$ first copies of $K_{2,d}$, one edge belongs to C_T and d edges belong to C_G. Therefore, we have $\|C_G\| = d(|\mathcal{T}| - 1) = 2d\|F_T\|$, which yields the desired result. Figure 4 shows an example where $d = 4$ and $|\mathcal{T}| = 3$.

However, this does not imply that the ratio given in the statement of Theorem 5 is tight. Indeed, in the above example, we have *Opt*(MaxEDP)=6.

Note that, in graphs where all the edges have the same capacity, Theorem 5 applies to MaxIMF and MaxUSF as well (since at most one flow path is associated with each net and only edge-disjoint flow paths are used). More

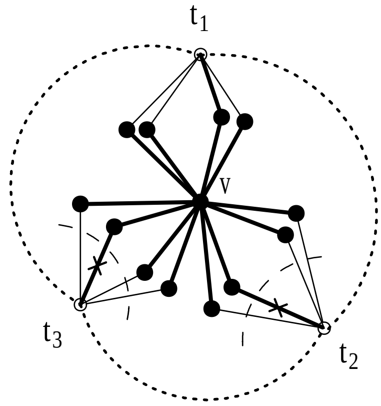


Figure 4: A 2-edge-outerplanar graph, with three terminals and $d = 4$. The three nets are drawn in dotted lines, and the edges of the spanning tree T are in bold lines. Here, we have $\|F_T\| = 1$, $\|C_T\| = 2$ (edges drawn as $\times$) and $\|C_G\| = 8$ (edges intersecting the dashed lines).

generally, we have the following corollary:

Corollary 2. *The integrality gap for* MaxIMF *and* MaxUSF *is bounded by* $4\beta k$ *in* k*-edge-outerplanar graphs* $G = (V, E)$ *satisfying* $\max_{e\in E} c(e) \leq \beta \min_{e\in E} c(e)$. *Moreover, solutions achieving this ratio can be computed in polynomial time.*

Finally, we would like to point out that, unfortunately, our approach fails (and seems hard to adapt) if the condition $\max_{e\in E} c(e) \leq \beta \min_{e\in E} c(e)$ does not hold. Indeed, if we consider the example given in Figure 1, and if we assume that all the edges lying on the outer face are weighted by an integer $N > 0$ and all the other edges by 1, our approach would yield an integer multiflow F_T and two multicuts C_T and C_G such that $\|F_T\| = 1$, $\|C_T\| = 1$ and $\|C_G\| = 2N + 1$ (by selecting all the edges in $\delta_G(u, v)$). However, it should be noticed that the conditions $c(e) \geq \beta$ (considered in [11, 12, 14, 41]) and $c(e) \leq \beta$ (considered in this section, and, to our best knowledge, in no previous work) for each $e \in E$ and for some integer $\beta > 0$, are different in nature. The first one does not allow to define the basic problem MaxEDP (which, as we already observed, captures the essential hardness of MaxIMF), but it makes the integrality gap shrink to a constant or logarithmic factor (this is obviously not the case for the second one).

5.3 MaxIMF and MinMC in edge-outerplanar graphs

In this section, we consider the class of graphs where the degree of each vertex is bounded by two (i.e., is equal to 0 or 2) inside each block. Note that this is exactly the class of graphs where two arbitrary (and not necessarily inclusionwise maximal) 2-vertex-connected components share at most one vertex, i.e., the class of graphs where each block is restricted to be a ring: hence, this is the class of edge-outerplanar graphs (or cacti). Obviously, such graphs generalize the trees of rings: a tree of rings is a graph obtained from a tree by replacing each vertex by a ring, two rings sharing a vertex if and only if the corresponding vertices of the tree are adjacent. Another definition is that a tree of rings is a 2-edge-connected edge-outerplanar graph.

The polynomial reduction given in [25] shows that MaxEDP (and thus MaxIMF) is $\mathcal{NP}$-hard and $\mathcal{APX}$-hard in edge-outerplanar graphs. Moreover, Erlebach shows that this also holds in trees of rings and gives a 3-approximation algorithm for unit capacitated MaxUSF in these graphs [18]. We now show how to obtain 4-approximation algorithms for both MaxIMF and MinMC in edge-outerplanar graphs. The idea is to use the algorithm given in Section 3. Given an edge-outerplanar (connected) graph G, we denote by R_i, $i \in \{1, \ldots, \rho\}$, its i^{th} cycle (ring). Then, for each i, we remove the edge e_i in R_i having the smallest capacity among all the edges in R_i. This way, we obtain a maximum weight spanning tree T of G, and we can compute an integer multiflow F_T and a multicut C_T for T such that $\|C_T\| \leq 2\|F_T\|$ by using the algorithm given in [25]. Eventually, we construct a multicut C_G for G: for each cycle R_i, we select the edge e_i in C_G if and only if there is another edge of R_i in C_T. Moreover, we add in C_G all the edges of C_T. We have $\|C_G\| = \|C_T\| + \sum_{e_i \text{ / there is an edge of } R_i \text{ in } C_T} c(e_i) \leq 2\|C_T\| \leq 4\|F_T\|$. It is easily seen that C_G is indeed a multicut for G, since, for each edge $e_i = (a_i, b_i)$ not selected in C_G, there exists a path from a_i to b_i in T (i.e., a path in R_i that does not cross e_i). This implies:

Theorem 6. *In edge-outerplanar graphs, the integrality gap for* MaxIMF *(resp.* MinMC*) is at most* 4*. Moreover, a solution for* MaxIMF *(resp.* MinMC*) achieving this ratio can be computed in polynomial time.*

Note that Theorem 6 also holds for MaxUSF. Moreover, this theorem shows that the integrality gap for MaxIMF shrinks to a factor of 4 when the maximum inside degree is at most 2, while it can be as large as $\sqrt{n}$ when the maximum degree is 3 [25, p. 17].

Finally, the family of instances given in Figure 4 proves that our analysis is tight, since there exist instances where $\|C_G\|$ is equal to $4\|F_T\|$ (by setting $d = 2$). Nevertheless, this does not necessarily imply that the integrality gaps given for MaxIMF and MinMC in Theorem 6 are tight.

6 Conclusion

In this paper, we have generalized all the results obtained for trees by Garg, Vazirani and Yannakakis to graphs with a fixed cyclomatic number.

In particular, this implies that, in these graphs, MaxEDP is polynomial-time solvable and MaxIMF has an integrality gap bounded by two times one plus the cyclomatic number. It is worth mentioning that our algorithmic approaches are simple and directly rely on algorithms for trees, so any improvement for these algorithms (improved running times, parallelization, online versions, etc.) can immediately be used for ours. Moreover, we have shown that other classical generalizations do not lead to results such as ours. We have also introduced a new class of planar graphs, the k-edge-outerplanar graphs. We have proved that the integrality gap for MaxEDP is bounded in these graphs and have shown how they are related to k-outerplanar graphs. Furthermore, we have shown that the integrality gap for MaxIMF is bounded by 4 in edge-outerplanar graphs (or cacti), a class of graphs that generalizes the trees of rings.

However, there are still interesting open problems for which no significant progress has been made: can we improve the $O(\sqrt{n})$ approximation ratio for MaxEDP in planar graphs, or can an inapproximability result stronger than $\mathcal{APX}$-hardness be proved for this problem? And what about the general graphs? Turning back to our results, one may also explore further the fixed-parameter tractability of MaxEDP [17]. Furthermore, is the integrality gap for MaxEDP or MaxIMF bounded by a constant in k-outerplanar graphs, or even in bounded tree-width graphs? Finally, the last open problem we would like to mention concern the k-edge-outerplanar graphs. Given a planar graph, Bienstock and Monma have shown that a k-outerplanar embedding for which k is minimal can be found in polynomial time [6]. It would be interesting to find, if such exists, a similar result for k-edge-outerplanar graphs.

Acknowledgements

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

References

[1] M. Andrews, J. Chuzhoy, S. Khanna and L. Zhang. Hardness of the undirected edge-disjoint paths problem with congestion. Proceedings FOCS 05 (2005).

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

[3] A. Baveja and A. Srinivasan. Approximation algorithms for disjoint paths and related routing and packing problems. Math. Oper. Res. 25 (2000) 255–280.

[4] C. Bentz. Edge disjoint paths and max integral multiflow/min multicut theorems in planar graphs. Proceedings 7th International Colloquium on Graph Theory (ICGT '05), Electronic Notes in Discrete Mathematics 22 (2005) 55–60.

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

[6] D. Bienstock and C. Monma. On the complexity of embedding planar graphs to minimize certain distance measures. Algorithmica 5 (1990) 93–109.

[7] H.L. Bodlaender. Planar graphs with bounded treewidth. Technical Report RUU-CS-88-14, Utrecht University, The Netherlands (1988).

[8] 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.

[9] P. Carmi, T. Erlebach and Y. Okamoto. Greedy edge-disjoint paths in complete graphs. Proceedings 29th International Workshop on Graph Theoretic Concepts in Computer Science. LNCS 2880 (2003) 143–155.

[10] C. Chekuri and S. Khanna. Edge disjoint paths revisited. Proceedings SODA 03 (2003).

[11] C. Chekuri, S. Khanna and B. Shepherd. Edge-disjoint paths in planar graphs. Proceedings 45th IEEE FOCS (2004).

[12] C. Chekuri, S. Khanna and B. Shepherd. Multicommodity flow, well-linked terminals, and routing problems. Proceedings STOC 05 (2005).

[13] C. Chekuri, S. Khanna and F.B. Shepherd. An $O(\sqrt{n})$-approximation and integrality gap for disjoint paths and UFP. Theory of Computing 2 (2006) 137–146.

[14] C. Chekuri, S. Khanna and F.B. Shepherd. Edge-disjoint paths in planar graphs with constant congestion. Proceedings STOC 06 (2006).

[15] C. Chekuri, S. Khanna and F.B. Shepherd. A note on multiflows and treewidth. To appear in Algorithmica (2007).

[16] 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.

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

[18] T. Erlebach. Approximation algorithms and complexity results for path problems in trees of rings. Proceedings 26th International Symposium on Mathematical Foundations of Computer Science. LNCS 2136 (2001) 351–362.

[19] S. Even, A. Itai and A. Shamir. On the complexity of timetable and multicommodity flow problems. SIAM J. Comput. 5 (1976) 691–703.

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

[21] A. Frank. Edge-disjoint paths in planar graphs. Journal of Combinatorial Theory, Series B 39 (1985) 164–178.

[22] A. Frank. Packing paths, circuits and cuts – a survey. In B. Korte, L. Lovász, H. J. Prömel and A. Schrijver. *Paths, Flows and VLSI-Layout.* Algorithms and combinatorics 9 (1990) 47–100. Springer-Verlag. Berlin.

[23] A. Frieze. Edge-disjoint paths in expander graphs. SIAM Journal on Computing 30 (2000) 1790–1801.

[24] 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.

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

[26] A.M.H. Gerards and B. Shepherd. Preselecting homotopies for the weighted disjoint paths problem. Manuscript (1993). Available at: http://www.math.mcgill.ca/∼bshepherd/PS/homotopy.ps

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

[28] J. Kleinberg and É. Tardos. Disjoint paths in densely embedded graphs. Proceedings 36th IEEE FOCS (1995) 52–61.

[29] J. Kleinberg. Approximation algorithms for disjoint paths problems. PhD thesis, MIT, Cambridge, MA (1996).

[30] J. Kleinberg and É. Tardos. Approximations for the disjoint paths problem in high-diameter planar networks. Journal of Computer and System Sciences 57 (1998) 61–73.

[31] S. Kolliopoulos and C. Stein. Approximating disjoint-paths using greedy algorithms and packing integer programs. Proceedings IPCO 98 (1998).

[32] P. Kolman and C. Scheideler. Improved bounds for the unsplittable flow problem. Proceedings SODA 02 (2002), 184–193.

[33] P. Kolman. A note on the greedy algorithm for the unsplittable flow problem. Information Processing Letters 88 (2003) 101–105.

[34] E. Korach and M. Penn. A fast algorithm for maximum integral two-commodity flow in planar graphs. Discrete Applied Mathematics 47 (1993) 77–83.

[35] J.B. Kruskal. On the shortest spanning subtree of a graph and traveling salesman problem. Proc. Amer. Math. Soc. 7 (1956) 48–50.

[36] D. Marx. Eulerian disjoint paths problem in grid graphs is NP-complete. Discrete Applied Mathematics 143 (2004) 336–341.

[37] M. Middendorf and F. Pfeiffer. On the complexity of the disjoint paths problem. Combinatorica 13 (1993) 97–107.

[38] T. Nishizeki, J. Vygen and X. Zhou. The edge-disjoint paths problem is NP-complete for series-parallel graphs. Discrete Applied Mathematics 115 (2001) 177–186.

[39] K. Obata. Approximate max-integral-flow/min-multicut theorems. Proceedings STOC 04 (2004).

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

[41] P. Raghavan. Probabilistic construction of deterministic algorithms: approximating packing integer programs. Journal of Computer and System Sciences 37 (1988) 130–143.

[42] S. Rao and S. Zhou. Edge disjoint paths in moderately connected graphs. Proceedings ICALP 06 (2006), 202–213.

[43] N. Robertson and P.D. Seymour. Graph minors II: algorithmic aspects of tree-width. Journal of Algorithms 7 (1986) 309–322.

[44] N. Robertson and P.D. Seymour. Graphs minors XIII: the disjoint paths problem. J. Combin. Theory, Ser. B, 63 (1995) 65–110.

[45] 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.