To appear in Combinatorica. Preliminary version appeared in IPCO '96.

Primal-Dual Approximation Algorithms for Feedback Problems in Planar Graphs

Michel X. Goemans[*] M.I.T. *David P. Williamson*[†] IBM Watson

Abstract

Given a subset of cycles of a graph, we consider the problem of finding a minimum-weight set of vertices that meets all cycles in the subset. This problem generalizes a number of problems, including the minimum-weight feedback vertex set problem in both directed and undirected graphs, the subset feedback vertex set problem, and the graph bipartization problem, in which one must remove a minimum-weight set of vertices so that the remaining graph is bipartite. We give a $\frac{9}{4}$-approximation algorithm for the general problem in planar graphs, given that the subset of cycles obeys certain properties. This results in $\frac{9}{4}$-approximation algorithms for the aforementioned feedback and bipartization problems in planar graphs. Our algorithms use the primal-dual method for approximation algorithms as given in Goemans and Williamson [16]. We also show that our results have an interesting bearing on a conjecture of Akiyama and Watanabe [2] on the cardinality of feedback vertex sets in planar graphs.

1 The problems

We consider the following general problem: given a graph $G = (V, E)$, non-negative weights w_i on the vertices $i \in V$, and a collection $\mathcal{C}$ of cycles of G, find a minimum-cost set of vertices F such that every cycle in $\mathcal{C}$ contains some vertex of F. We call this problem the *hitting cycle* problem, since we must hit every cycle in $\mathcal{C}$. The hitting cycle problem generalizes several other problems we will study in this paper. If $\mathcal{C}$ is the set of all cycles in G, then the hitting cycle problem is equivalent to the problem of finding a minimum-weight *feedback vertex set* in a graph; that is, the problem of finding a minimum-weight set $F \subseteq V$ such that the graph $G[V - F]$ induced by $V - F$ is acyclic. The feedback vertex set problem will be abbreviated by FVS. If G is a directed graph (digraph), and $\mathcal{C}$ the set of all directed cycles in G, then we have the minimum-weight feedback vertex set problem in directed graphs (D-FVS). If we are given a set of *special* vertices and $\mathcal{C}$ is all cycles of an undirected graph G that contain some special vertex, then we have the *subset* feedback vertex set problem (S-FVS). Finally, if $\mathcal{C}$ contains all odd cycles of G, then we have the *graph bipartization* problem (BIP); that is, the problem of finding a minimum-weight subset F such that $G[V - F]$ is bipartite. All these problems are also special cases of *vertex deletion problems*: that is, find a minimum-weight (or minimum cardinality) set of vertices whose deletion gives a graph satisfying a given property.

[*]Address: Dept. of Mathematics, Room 2-382, M.I.T., Cambridge, MA 02139. Email: goemans@math.mit.edu.
Research supported in part by NSF contracts 9302476-CCR and 9623859-CCR, DARPA contract N00014-92-J-1799, a Sloan fellowship, and IBM. This research was conducted in part while the author was visiting IBM.

[†]Address: IBM T.J. Watson Research Center, Room 33-219, P.O. Box 218, Yorktown Heights, NY, 10598. Email: dpw@watson.ibm.com.

We will restrict our attention to the versions of these problems in which the input graph is planar and simple. Yannakakis [28] has given a general NP-hardness proof for almost all vertex deletion problems restricted to planar graphs; his results apply to the planar (directed, undirected or subset) feedback vertex set problem and to the planar graph bipartization problem. In addition, the planar D-FVS is NP-hard even if both the indegree and outdegree of every vertex is no more than 3 [12, p. 192].

We consider *approximation algorithms* for these problems. An α-approximation algorithm for a minimization problem runs in polynomial time and produces a solution of weight no more that α times the weight of an optimal solution. We call α the *performance guarantee* of the algorithm. In this paper, we give a $\frac{9}{4}$-approximation algorithm for a general class of planar hitting cycle problems which includes the planar feedback vertex set problem in undirected or directed graphs, the planar subset feedback vertex set problem in undirected graphs, and the planar graph bipartization problem.

Our algorithms are based on the primal-dual method for approximation algorithms. This method has proven useful over the past few years in designing algorithms for network design problems (see, for example, [15, 13, 21, 27]). The authors have written a survey of this method [16] which gives a generic algorithm and theorem for deriving approximation algorithms for the hitting set problem, of which the hitting cycle problem is a special case. The algorithm and analysis here are an application of the algorithm and theorem given in the survey.

We now review previously known work. For FVS in general undirected graphs, two slightly different 2-approximation algorithms were given recently by Becker and Geiger [6] and Bafna, Berman, and Fujito [4]; see Chudak et al. for an overview [8]. These algorithms improve on a $\log n$-approximation algorithm of Bar-Yehuda, Geiger, Naor, and Roth [5], where n is the number of vertices. They also gave a 10-approximation algorithm for the case of undirected planar graphs, which we can show to be a 5-approximation algorithm for this case. None of these algorithms apply to the feedback vertex set problem in directed graphs. Even, Naor, Schieber, and Sudan [9] show that a result of Seymour [25] can be converted to an $O(\log n \log\log n)$-approximation algorithm for general directed graphs. This observation improves on an $O(\log^2 n)$-approximation algorithm for this case due to Leighton and Rao [22]. In the case of directed planar graphs, Stamm [26] has given an $O(n \log n)$ time approximation algorithm whose performance guarantee is bounded by Δ, the maximum degree of the graph, and an $O(n^2)$ time approximation algorithm with performance guarantee no more than the number of cyclic faces in the planar embedding of the graph minus 1.

For the subset feedback vertex set problem in general undirected graphs, the first approximation algorithm is due to Klein, Rao, Agrawal and Ravi [20], who give a $O(\log^3 n)$-approximation algorithm. A very recent result due to Even, Naor, and Zosin [11] shows an 8-approximation algorithm. To the best of our knowledge, no previous approximation algorithm has been given for the special case of planar graphs.

For the graph bipartization problem, Klein et al. give a $O(\log^3 n)$-approximation algorithm. Garg, Vazirani, and Yannakakis [14] give an improved $O(\log n)$-approximation algorithm. As before, to the best of our knowledge, no previous approximation algorithm was known for the case of planar graphs.

Although our result for the undirected feedback vertex set problem on planar graphs is worse than the known approximation algorithm for general undirected graphs, it still turns out to be interesting. Our result implies that the LP relaxation of the cycle formulation of all four problems is within a factor of 9/4 of the corresponding optimum value for planar graphs. This is known to be false for general graphs (the ratio can be logarithmic in n [25, 10]). This ratio has an interesting connection to a conjecture of Akiyama and Watanabe [2] and Albertson and Berman [3] which we discuss in Section 6. Their conjecture states that any undirected planar graph on n vertices contains a feedback vertex set of size no more than $n/2$, and that any undirected planar bipartite graph contains a feedback vertex set of size $3n/8$. Our bound of $9/4$ implies the existence of a feedback vertex set of size at most $3n/4$ in planar graphs, and a feedback vertex set of size at most $9n/16$ if the graph is also bipartite. The first statement follows easily from the 4-color theorem, but we don't know of any other proof besides our own. A coloring result of Borodin [7] shows that any planar graph has a feedback vertex set of size no more than $3n/5$; however, Jensen and Toth [19, p. 6] call the proof reminiscent of the proof of the 4-color theorem, partly because it involves 450 reducible configurations.

It is also possible to consider *edge* counterparts of the given problems; that is, find a minimum-weight subset of edges F that meet every cycle in a given collection $\mathcal{C}$. This leads to the minimum-weight feedback edge set problem in undirected graphs, the minimum-weight feedback arc set problem in directed graphs, and the minimum-weight graph bipartization problem via edge removals. However, these problems tend to be simpler than their vertex counterparts, especially for planar graphs. The feedback problem in general undirected graphs is trivially the complement of the maximum spanning tree problem. The minimum-weight bipartization problem is complementary to the maximum-weight cut problem in planar graphs, which is polynomial-time solvable (Hadlock [18]; Orlova and Dorfmann [24]) since the problem is equivalent to a T-join problem in the dual graph. The feedback arc set problem in planar digraphs is well-known to be reducible to finding a minimum-weight dijoin in the dual graph, which can be solved in polynomial time (see, for example, Grötschel, Lovász, and Schrijver [17, p. 253,254]). Given a directed graph $G = (V, A)$, a *dijoin* $A' \subseteq A$ is a set of arcs such that $G = (V, B)$, $B = A \cup \{(v, u) | (u, v) \in A'\}$, is strongly connected. In the minimum-weight dijoin problem, we are given non-negative weights w_a for $a \in A$, and we must find the minimum-weight dijoin. Stamm [26] has given a simple 2-approximation algorithm for this problem by superposing two arborescences. It is interesting to notice that all these problems, when translated to the dual graph, lead to problems of hitting certain cutsets of the dual graph, problems which can be approximated within a ratio of 2 by the primal-dual method [15, 27, 16].

The paper is structured as follows. In Section 2 we begin with some preliminary concepts and definitions. Section 3 reviews the generic primal-dual algorithm and its analysis from Goemans and Williamson [16]. In Section 4, we show how the algorithm leads to a 3-approximation algorithm for a class of hitting cycle problems, and in Section 5 we improve the algorithm and its analysis to give a $\frac{9}{4}$-approximation algorithm. We comment on the integrality gap of the linear programming relaxation and its relation to Akiyama and Watanabe's conjecture in Section 6. The implementation of the algorithms is described in Section 7, and we conclude in Section 8.

2 Preliminaries

When we refer to a cycle C of an undirected graph $G = (V, E)$, we refer to its vertex set, even though this is somewhat ambiguous. If we would like to refer to its edge set, we will write $E(C)$.

Recall the hitting cycle problem defined in the previous section. Let G be an undirected graph, let $w_i \geq 0$ be the weight of vertex i, and let $\mathcal{C}$ be a collection of cycles of G. The hitting cycle problem is that of finding a minimum-weight set F of vertices such that F intersects every member of $\mathcal{C}$. In most cases, when we will refer to a *cycle*, we will implicitly mean a cycle of $\mathcal{C}$, unless stated otherwise.

We will restrict our attention to families $\mathcal{C}$ satisfying the following property. We abuse notation slightly here by referring to cycles C as both sets of edges and of vertices. Paths P are sets of edges; for directed graphs, the set of edges is a path for the underlying undirected graph.

Property A For any two cycles $C_1, C_2 \in \mathcal{C}$, let P_2 be a path in C_2 which is edge-disjoint from C_1 and which intersects C_1 only at the endpoints of P_2. Let P_1 be a path in C_1 between the endpoints of P_2. Then either $P_1 \cup P_2 \in \mathcal{C}$ and $(C_1 - P_1) \cup (C_2 - P_2)$ contains a cycle in $\mathcal{C}$, or $(C_1 - P_1) \cup P_2 \in \mathcal{C}$ and $(C_2 - P_2) \cup P_1$ contains a cycle in $\mathcal{C}$.

We will refer to families satisfying Property A as *uncrossable*. Our approximation algorithms will apply to any uncrossable hitting cycle problem for input graphs restricted to be planar, given that we can compute efficiently certain minimal cycles which we will define in a moment.

We claim that the problems we are interested in correspond to uncrossable families. First notice that the (multi)graph $H = E(C_1) \cup E(C_2)$ is Eulerian, i.e. every vertex has even degree, or every vertex has indegree equal to outdegree in the case of D-FVS. Also, when removing a cycle C from H, the resulting multigraph remains Eulerian (assuming C is directed in the case of D-FVS). It can therefore be decomposed into cycles. However, we have to be somewhat careful since these cycles may be trivial cycles consisting simply of duplicated edges. Taking $C = P_1 \cup P_2$, this shows that property A is satisfied for FVS unless $(C_1 - P_1) \cup (C_2 - P_2)$ only consists of trivial cycles. However, in this latter case, $C_1 - P_1 = C_2 - P_2$ and both $(C_1 - P_1) \cup P_2$ and $P_1 \cup (C_2 - P_2)$ are simple cycles, implying Property A. For D-FVS, Property A is also satisfied. Let a and b be the two endpoints of the path P_2. Then either P_2 is directed from a to b (and $C_2 - P_2$ is directed from b to a) or vice versa. Thus, either $P_1 \cup P_2$ or $(C_1 - P_1) \cup P_2$ defines a directed cycle C, and $H - E(C)$ contains a directed cycle since it is Eulerian (directed cycles cannot consist of duplicated edges). For S-FVS, there must be a special vertex on either P_1 or $C_1 - P_1$ and also on either P_2 or $C_2 - P_2$. Therefore, we can make sure that the Eulerian graph $H - E(C)$ still contains a special vertex, say v. Moreover, in a cycle decomposition of $H - E(C)$, v will only be on trivial cycles if the edges incident to v in C_1 and C_2 are identical. In this case, taking $C = P_1 \cup P_2$ (resp. $C = (C_1 - P_1) \cup P_2$) if $v \in P_1$ (resp. $v \in (C_1 - P_1)$) would give an Eulerian graph $H - E(C)$ for which v is on a non-trivial cycle. Thus, one of the two cases of Property A must hold. For BIP, we observe that $P_1 \cup P_2$ and $(C_1 - P_1) \cup P_2$ have different parities, and therefore one of them must be odd. Moreover, $H - E(C)$ is Eulerian and has an odd number of edges if C is odd, and therefore must contain an odd cycle (which cannot be trivial) in any cycle decomposition. So, once again, Property A holds.

The FVS, S-FVS and BIP also satisfy an additional property:

Property B For any cycle $C \in \mathcal{C}$ and any path P, edge-disjoint from C and intersecting C only at the endpoints of P, let C_1, C_2 be the two cycles defined by C and P. Then either C_1 or C_2 (or both) belongs to $\mathcal{C}$.

Observe that this is not the case for D-FVS since there is no guarantee that P is a directed path. Property B will be useful for implementation purposes.

Our approximation algorithms for uncrossable hitting cycle problems will depend on the embedding of the planar graph. Given a plane graph G (i.e. a planar graph with an embedding), any cycle C partitions the plane into two regions, the interior and exterior regions. We will associate to any cycle C the set $f(C)$ of faces in the interior region of C. Observe that the exterior face of the embedding of G never belongs to $f(C)$. We will say that cycle C_1 contains cycle C_2 and write $C_1 \supseteq_f C_2$ or $C_2 \subseteq_f C_1$ if $f(C_1) \supseteq f(C_2)$. Two cycles C_1 and C_2 are said to be crossing if $f(C_1)$ and $f(C_2)$ cross[1], i.e. $f(C_1) \cap f(C_2) \neq \emptyset$, $f(C_1) - f(C_2) \neq \emptyset$ and $f(C_2) - f(C_1) \neq \emptyset$. Similarly, we say that a collection of cycles form a *laminar family* if no two cycles are crossing.

[1]Observe that the exterior face is never in $f(C_1) \cup f(C_2)$, and thus the notions of crossing and intersecting are equivalent.

We say that a cycle $C \in \mathcal{C}$ is *face-minimal* if there does not exist a cycle $C' \in \mathcal{C}$, $C' \neq C$, with $f(C') \subseteq_f f(C)$. The collection of face-minimal cycles will play a central role in our approximation algorithms. The following lemma shows that face-minimal cycles form a laminar family.

Lemma 2.1 Let $\mathcal{C}$ satisfy Property A and let $C_1, C_2 \in \mathcal{C}$. If C_1 is a face-minimal cycle then C_1 and C_2 do not cross.

Proof: The proof follows immediately from Property A. If the two cycles were to cross, then by choosing P_2 to be a path in C_2 which lies in the interior of C_1, the two cycles $P_1 \cup P_2$ and $(C_1 - P_1) \cup P_2$ would both be contained in C_1. This is a contradiction since at least one of them belongs to $\mathcal{C}$ and C_1 is face-minimal. ■

Whenever $\mathcal{C}$ satisfies Property B, the face-minimal cycles have a very simple structure given by the next lemma.

Lemma 2.2 Let $\mathcal{C}$ satisfy Property B. Then the face-minimal cycles are the boundaries of the interior faces which are simple cycles.

Proof: Suppose C is a face-minimal cycle of $\mathcal{C}$ which is not given by the boundary of an interior face. Then there must be a path P in the interior of C that only intersects C at its endpoints. Using property B, one of the two cycles defined by C and P must be in $\mathcal{C}$. But this cycle must be contained in C, which contradicts the face-minimality of C. ■

In particular, for families satisfying Property B, this lemma shows that the face-minimal cycles are the boundaries of *all* interior faces corresponding to cycles in $\mathcal{C}$ if the graph is 2-connected. The lemma is not true for the directed feedback vertex set problem, which does not satisfy Property B.

3 The primal-dual framework

The uncrossable hitting cycle problem is a special case of the general *hitting set problem* in which one needs to find a minimum-weight set hitting every set in a given collection of sets. More precisely, given a ground set of elements E, weights c_e for all $e \in E$, and sets $T_1, \ldots, T_p \subseteq E$, the hitting set problem is that of finding a minimum-weight $A \subseteq E$ such that $A \cap T_i \neq \emptyset$ for $i = 1, \ldots, p$. In a recent survey [16], we have developed a general methodology to derive approximation algorithms for hitting set problems based on the so-called *primal-dual method*. This was motivated by a sequence of papers [1, 15, 21, 27] developing the technique for network design problems. In the survey, we propose a generic primal-dual method for deriving approximation algorithms for hitting set problems, with a generic proof of the performance guarantee. We illustrate in [16] the technique on a variety of problems, and also claim that the method can be applied to many more problems. As we show here, the technique directly applies to any uncrossable hitting cycle problem in planar graphs.

A hitting cycle problem can be formulated by the following integer program (IP):

$$
(IP) \qquad \begin{array}{lll}
 & \text{Min} \ \sum_{i \in V} w_i x_i & \\
\text{subject to:} & & \\
 & \sum_{i \in C} x_i \geq 1 & \text{cycles } C \in \mathcal{C} \\
 & x_i \in \{0,1\} & i \in V.
\end{array}
$$

$y \leftarrow 0$
$S \leftarrow \emptyset$
$l \leftarrow 0$
While S is not feasible
  $l \leftarrow l + 1$
  $\mathcal{V} \leftarrow$ VIOLATION(S)
  Increase y_C uniformly for all $C \in \mathcal{V}$ until $\exists v_l \notin S : \sum_{C : v_l \in C} y_C = w_{v_l}$
  $S \leftarrow S \cup \{v_l\}$
For $j \leftarrow l$ downto 1
  if $S - \{v_j\}$ is feasible then $S \leftarrow S - \{v_j\}$
Output S (and y)

Figure 1: Primal-dual algorithm for uncrossable hitting cycle problems.

The primal-dual method simultaneously constructs a feasible solution to this hitting set problem, and a solution feasible for the dual of the linear programming relaxation of (IP). The dual of the LP relaxation is:

$$
(D) \qquad \begin{array}{lll}
\text{Max} & \displaystyle\sum_{C \in \mathcal{C}} y_C & \\
\text{subject to:} & & \\
& \displaystyle\sum_{C : i \in C} y_C \leq w_i & i \in V \\
& y_C \geq 0 & C \in \mathcal{C}.
\end{array}
$$

The generic primal-dual method developed in [16] is described in Figure 1. It is specified by the oracle VIOLATION(S) which given a set of vertices S outputs a specific set of cycles in $\mathcal{C}$ which are not hit by S. The algorithm begins with an empty set of vertices S and a dual solution $y = 0$. While S is not a feasible solution to the hitting cycle problem, it increases the dual variables on the cycles returned by VIOLATION(S) until one of the dual packing constraints becomes tight for some vertex $i \in V$. This vertex is added to S and the process continues. When S becomes feasible, the algorithm performs a "clean-up" step. It goes through the vertices in the reverse of the order in which they were added and removes any vertex which is not necessary for S to remain feasible.

In [16], it is proved that the performance guarantee of this algorithm can be obtained by using the following theorem. In this theorem, a *minimal augmentation* F of S means a feasible solution F containing S such that for any $v \in F - S$, $F - v$ is not feasible.

Theorem 3.1 [Goemans and Williamson [16]] The primal-dual algorithm described in Figure 1 delivers a solution of cost at most $\gamma \sum_C y_C \leq \gamma z_{OPT}$, where z_{OPT} denotes the weight of an optimum solution, if γ satisfies that for any infeasible set $S \subset V$ and any minimal augmentation F of S

$$\sum_{C \in \mathcal{V}(S)} |F \cap C| \leq \gamma |\mathcal{V}(S)|,$$

where $\mathcal{V}(S)$ denotes the collection of violated sets output by the VIOLATION oracle on input S.

Therefore, we only need to specify what the VIOLATION oracle does, compute the value of γ given by Theorem 3.1, and prove that the algorithm runs in polynomial time in order to obtain a γ-approximation algorithm. Observe that by considering $G - S$, we can assume without loss of generality that, in Theorem 3.1, $S = \emptyset$ and F is a minimal feasible solution.

One possibility is that the VIOLATION oracle returns only one cycle. This is essentially the approach used by Bar-Yehuda et al. [5] for FVS in general graphs. They gave a 10-approximation algorithm for this problem in planar graphs by simply finding a "short" cycle in the graph, but their analysis can be improved. We give below a brief sketch of their VIOLATION oracle and of the improved analysis. Given the planar graph G, we can first assume that G has no degree 1 vertex since such vertices can be deleted without affecting the cycles of G. We claim that the resulting graph has a cycle with at most 5 vertices of degree 3 or higher; moreover, this cycle can be chosen to be (part of) the boundary of a face. It is then easy to see that γ can be chosen to be 5 in Theorem 3.1. To prove the claim, observe that, if the graph is 2-connected, the claim is equivalent to the existence of a vertex of degree at most 5 in the dual graph, a well-known fact (since the sum of the degrees is at most $6|V| - 12$). If the graph is not 2-connected, we consider an endblock of the graph (i.e. a block with at most one cutvertex) and use the same argument. The only slight problem is that the resulting cycle may contain the cutvertex and this cutvertex may have degree 2 in the endblock. This however can be dealt with by using the fact that a planar graph has more than one vertex of degree at most 5. The idea of having the VIOLATION oracle return only one cycle does not seem to work for S-FVS, D-FVS or BIP.

4 A 3-approximation algorithm

In this section, we consider the VIOLATION oracle which, on input S, returns the set of face-minimal cycles of $G - S$ (with respect to $\mathcal{C}$). We will refer to this oracle as FACE-MINIMAL. We show that the corresponding value of γ is 3. In the following section, we give a refined oracle for which the corresponding γ is 9/4. These performance guarantees are tight for D-FVS, S-FVS and BIP.

In order to prove that FACE-MINIMAL has a γ value of 3, we need to show the following result (applied to the graph $G - S$).

Theorem 4.1 Let G be a planar graph and let $\mathcal{M}$ be the collection of face-minimal cycles corresponding to an uncrossable family $\mathcal{C}$. Consider any minimal solution F. Then

$$\sum_{C \in \mathcal{M}} |F \cap C| \leq 3|\mathcal{M}|.$$

Since F is a minimal solution, we know that for every $v \in F$, $F - v$ is not feasible, implying the existence of a cycle $C_v \in \mathcal{C}$ such that $C_v \cap F = \{v\}$. We call such a cycle C_v a *witness cycle* (for v). A *family* of witness cycles is a collection of witness cycles $C_v \in \mathcal{C}$, one for each $v \in F$.

Lemma 4.2 There exists a laminar family of witness cycles $C_v \in \mathcal{C}$, $v \in F$.

Proof: Consider any family of witness cycles and assume the existence of two witness cycles C_u and C_v that cross for $u, v \in F$. By assumption $F \cap C_u = \{u\}$ and $F \cap C_v = \{v\}$. The assumption implies that u and v have degree 2 in $H = E(C_v) \cup E(C_u)$ and that no other vertices of H are in F. Since the cycles cross there is some path P_u of C_u in the interior of C_v which intersects C_v only at its endpoints. By Property A, C_u and C_v can be replaced by two cycles such that one is in $\mathcal{C}$, call it C', and the other contains a cycle say C'' in $\mathcal{C}$. Say that C_v is replaced by C'; by property

A, it will contain strictly fewer faces than C_v. Since F is feasible, both C' and C'' must be hit by F. However, since u and v have degree 2, it must be the case that C' and C'' each have exactly one of u and v and are witness cycles for u and v.

In order to show the existence of a laminar family of witness cycles, we need to prove that the crossing pairs of cycles being replaced can be selected in such a way that the replacing process terminates with a laminar family. We begin by arbitrarily choosing two cycles C_a and C_b that cross, and replacing them with two cycles C' and C'' as above. Either C' or C'' will contain a strict subset of the faces of C_a; suppose it is C'. If C' is crossed by any cycles, we continue the process by replacing this pair as above; otherwise, we stop and mark the witness cycle C'. This process must terminate since each time we replace a pair, one cycle in the pair contains a smaller subset of faces of the cycle C_a than a cycle in the previous pair. Once we have marked a cycle, we begin again by choosing two crossing witness cycles and continue as before, except that we never choose any marked cycle. The important observation to make is that as we replace a crossing pair C_a and C_b as explained in the first part of the proof, if a cycle C does not cross either C_a or C_b, then C still does not cross the new witness cycles C' and C'' for a and b. This follows from the fact that $f(C)$ must either be contained entirely in one of the faces of $H = E(C_a) \cup E(C_b)$ or must contain all the interior faces of H or is disjoint from the interior faces of H. Therefore, once we mark a witness cycle, it will never cross any other witness cycle during the course of the replacement process. Therefore, this uncrossing process terminates with a laminar family of witness cycles. ■

A laminar family $\mathcal{F} = \{C_v \in \mathcal{C} : v \in F\}$ of witness cycles can be represented by a tree or more precisely by a forest by considering the partial order imposed by $\subseteq_f$. To simplify the exposition, we can add a root node $\mathbf{r}$ which is connected to all maximal sets in the family, and thus obtain a tree $\mathbf{T}$. Notice that any vertex in $\mathbf{T}$ is either $\mathbf{r}$ or corresponds to a cycle C_v for $v \in F$. Thus for each vertex $v \in F$ we will correspond a vertex $\mathbf{v} \in \mathbf{T}$.

The crucial (and only) properties of $\mathcal{M}$ we will be using are the following:

1. No element of $\mathcal{M}$ crosses any element of $\mathcal{F}$. This follows from Lemma 2.1.

2. Every element of $\mathcal{F}$ (and therefore the cycles corresponding to the leaves of $\mathbf{T}$) contains at least one element of $\mathcal{M}$.

For the analysis, and because of these two properties, we assign every element of $\mathcal{M}$ to some node in the tree $\mathbf{T}$: cycle $C \in \mathcal{M}$ is assigned to the vertex of $\mathbf{T}$ corresponding to the smallest set in $\mathcal{F}$ (inclusion-wise) which contains it. For $\mathbf{v} \in \mathbf{T}$, let $\mathcal{M}_{\mathbf{v}}$ denote the set of cycles of $\mathcal{M}$ assigned to node $\mathbf{v}$ of $\mathbf{T}$. Observe that $\mathcal{M}_{\mathbf{r}}$ may be non-empty, and that some $\mathcal{M}_{\mathbf{v}}$ may be empty. However, because of property 2, $\mathcal{M}_{\mathbf{v}}$ is non-empty for every leaf $\mathbf{v}$ of $\mathbf{T}$.

In order to prove Theorem 4.1, we first derive an upper bound on $\sum_{C \in \mathcal{M}_{\mathbf{v}}} |F \cap C|$ for every $\mathbf{v} \in \mathbf{T}$. Fix $\mathbf{v} \in \mathbf{T}$, and let $F_{\mathbf{v}}$ denote the subset of vertices of F corresponding to $\mathbf{v}$ (unless $\mathbf{v} = \mathbf{r}$) and the children (if any) of $\mathbf{v}$ in $\mathbf{T}$. Observe that $F \cap C = F_{\mathbf{v}} \cap C$ for any $C \in \mathcal{M}_{\mathbf{v}}$. Thus, $\sum_{C \in \mathcal{M}_{\mathbf{v}}} |F \cap C| = \sum_{C \in \mathcal{M}_{\mathbf{v}}} |F_{\mathbf{v}} \cap C|$. By definition of $F_{\mathbf{v}}$, its cardinality is equal to the degree $deg(\mathbf{v})$ of node $\mathbf{v}$ in $\mathbf{T}$. In order to get an upper bound on $\sum_{C \in \mathcal{M}_{\mathbf{v}}} |F_{\mathbf{v}} \cap C|$, we construct a bipartite graph B. B has a vertex for every $u \in F_{\mathbf{v}}$ and for every $C \in \mathcal{M}_{\mathbf{v}}$, and an edge between u and C iff $u \in C$. Therefore, $\sum_{C \in \mathcal{M}_{\mathbf{v}}} |F_{\mathbf{v}} \cap C|$ is precisely the number of edges of B. Observe that B is planar, since a planar embedding of B can be obtained from the embedding of G by placing the vertex corresponding to $C \in \mathcal{M}_{\mathbf{v}}$ in the interior of C. But the number of edges of a simple bipartite planar graph is at most twice the number of vertices minus four, unless the graph consists simply of a single vertex or of two vertices with one edge. Notice that B can only be a single vertex if $\mathbf{v} = \mathbf{r}$. Also, B can be an edge on two vertices; this can occur only if $\mathbf{v}$ is a leaf of $\mathbf{T}$ or $\mathbf{v} = \mathbf{r}$.

We have therefore derived that

$$\sum_{C \in \mathcal{M}_{\mathbf{v}}} |F_{\mathbf{v}} \cap C| \leq 2|\mathcal{M}_{\mathbf{v}}| + 2|F_{\mathbf{v}}| - 4 = 2|\mathcal{M}_{\mathbf{v}}| + 2deg(\mathbf{v}) - 4, \tag{1}$$

unless $\mathbf{v}$ is a leaf of $\mathbf{T}$ in which case

$$\sum_{C \in \mathcal{M}_{\mathbf{v}}} |F_{\mathbf{v}} \cap C| \leq 2|\mathcal{M}_{\mathbf{v}}| + 2deg(\mathbf{v}) - 3,$$

or v corresponds to $\mathbf{r}$ in which case

$$\sum_{C \in \mathcal{M}_{\mathbf{r}}} |F_{\mathbf{v}} \cap C| \leq 2|\mathcal{M}_{\mathbf{r}}| + 2deg(\mathbf{r}) - 2.$$

Summing over all $\mathbf{v} \in \mathbf{T}$, we derive that

$$\sum_{C \in \mathcal{M}} |F \cap C| \leq 2|\mathcal{M}| + 2\sum_{\mathbf{v} \in \mathbf{T}} deg(\mathbf{v}) - 4|\mathbf{T}| + l + 2,$$

where l denotes the number of leaves of $\mathbf{T}$. Since $\mathbf{T}$ is a tree, $\sum deg(\mathbf{v})$ is equal to twice the number of nodes of the tree minus two. This implies that

$$\sum_{C \in \mathcal{M}} |F \cap C| \leq 2|\mathcal{M}| - 2 + l.$$

Moreover, because of property 2, the number l of leaves is upper bounded by $|\mathcal{M}|$. This therefore shows that

$$\sum_{C \in \mathcal{M}} |T \cap C| \leq 3|\mathcal{M}| - 2,$$

proving Theorem 4.1.

For FVS, the worst instance we are aware of for our primal-dual algorithm with the oracle Face-Minimal achieves a performance ratio of 2. However, for the other problems, namely D-FVS, S-FVS and BIP, the performance guarantee of 3 is tight. Instances achieving this ratio are given in Figure 2; the same figure applies to all three problems. There are k white vertices and they have a weight of 3, and the other (black) vertices have a weight of $1 + \epsilon$. In the case of S-FVS, the special vertices are denoted by (black) squares, while for D-FVS the orientation of the arcs along two of the faces are explicitly given on the figure (the orientation of the other arcs are such that the shaded faces define directed cycles). The face-minimal cycles are the boundaries of the shaded faces, and the algorithm will select all white vertices in the solution for a total weight of $3k$. However, in all three cases, the black squares constitute a feasible solution of weight $(k+2)(1+\epsilon)$, giving the desired bound as k gets large and ϵ tends to 0. The analysis of our algorithm in fact indicates that bad examples arise only when there are two cycles in $\mathcal{M}$ with several points in common. The improved Violation oracle we develop in the next section deals precisely with such cases.

5 A 9/4-approximation algorithm

We first need some preliminaries. Two (face)-disjoint[2] cycles C_1 and C_2 partition the plane into one or several regions; excluding the interiors of C_1 and C_2, each remaining region corresponds

[2]that is, $f(C_1) \cap f(C_2) = \emptyset$.

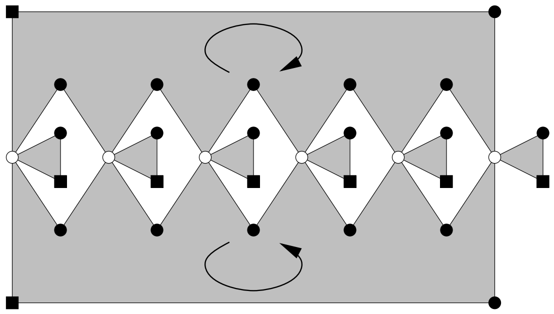

Figure 2: A bad example for the 3-approximation algorithm applied to BIP, to D-FVS, or to S-FVS.

to a connected component of the dual graph after having removed $f(C_1) \cup f(C_2)$. One of these regions contains the exterior face, and we refer to the others as the *pockets* between C_1 and C_2. The boundary of any pocket is defined by two vertices common to C_1 and C_2, say u and v, and consists of two paths between u and v, one from C_1 and one from C_2. If there exist k non-empty pockets between C_1 and C_2 then C_1 and C_2 must have at least $k+1$ vertices in common. We say that two disjoint cycles C_1 and C_2 *surround* a cycle C_3 if $f(C_3)$ is contained in one of the pockets between C_1 and C_2.

Our improved algorithm is based on the following oracle which returns a subset $\mathcal{V}$ of the family $\mathcal{M}$ of face-minimal cycles. If $\mathcal{M}$ does not contain two cycles which surround a third one then the oracle returns $\mathcal{M}$. Otherwise, the oracle outputs a non-empty subset $\mathcal{V}$ of $\mathcal{M}$ such that (i) there do not exist two cycles C_1 and C_2 in $\mathcal{V}$ which surround a third cycle of $\mathcal{V}$, and (ii) $\mathcal{V}$ consists of all cycles of $\mathcal{M}$ in one of the pockets between two cycles C_1 and C_2 of $\mathcal{M}$. This is always possible since the oracle can simply recursively select the non-empty set of cycles of one of the pockets between two cycles C_1 and C_2 until the remaining collection satisfies (i).

Theorem 5.1 Let G be a planar graph and let $\mathcal{V}$ be as defined in the paragraph above. Consider any minimal feasible solution F. Then

$$\sum_{C \in \mathcal{V}} |F \cap C| \leq \frac{9}{4}|\mathcal{V}|.$$

The structure of the proof is similar to the one in the previous section, the main difference being the proof of a sharper version of inequality (1). First, let us assume that $\mathcal{M}$ does not contain two cycles with more than one point in common. In this case, $\mathcal{V} = \mathcal{M}$. Then the bipartite graph B constructed in the proof of Theorem 4.1 does not have any cycle of length 4; indeed such a cycle would imply the existence of two cycles C_1 and C_2 in $\mathcal{M}_{\mathbf{v}}$ and two vertices x and y belonging both to C_1 and C_2. We need a sharper bound on the number of edges of such graphs.

Lemma 5.2 Let B be a simple bipartite planar with bipartition (V_1, V_2) where $a = |V_1|$ and $b = |V_2|$, and with e edges. Assume that B has no cycle of length 4 and is 2-edge-connected. Then

$$e \leq \frac{5}{4}b + \frac{9}{4}(a-2).$$

Proof: Let f be the number of faces, and let n_i denote the number of edges (or vertices) on face i. By assumption, $n_i \geq 6$. By summing this inequality over all faces, we obtain that $2e = \sum_i n_i \geq 6f$, i.e. $f \leq \frac{1}{3}e$. Together with Euler's formula stating that $f - e + a + b = 2$, this implies that

$$e \leq \frac{3}{2}(a + b - 2). \tag{2}$$

We now claim that $b \leq 3a - 6$. This follows from the following construction. Replace any vertex $u \in V_2$ and its incident edges by one edge joining two of the neighbors of u (there are at least 2 neighbors since the graph is 2-edge-connected). It is obvious that the resulting graph is planar. Moreover, it is simple; if not there would be two vertices x and y in V_1 both connected to two vertices u and v in V_2 implying the existence of a cycle of length 4. The number of edges of the resulting graph is b (since any vertex of V_2 was replaced by an edge) and the number of vertices is a. Moreover, $a \geq 3$ (since the original graph is 2-connected and cycles have length at least 6). Euler's formula now implies the claim.

The claim together with equation (2) implies that

$$e \leq \frac{5}{4}b + \frac{1}{4}b + \frac{3}{2}a - 3 \leq \frac{5}{4}b + \frac{3}{4}a - \frac{3}{2} + \frac{3}{2}a - 3 = \frac{5}{4}b + \frac{9}{4}a - \frac{9}{2}.$$

■

The bound in Lemma 5.2 is tight for any $a \geq 3$. Indeed, take a maximal planar graph on the vertex V_1 with $|V_1| = a$, and insert a vertex of V_2 on every edge. One easily checks that the resulting graph satisfies the assumptions and that the number of edges is precisely given by the bound in the lemma.

The next lemma deals with the case when the bipartite graph is not 2-edge-connected.

Lemma 5.3 Under the same assumptions as in Lemma 5.2 except that the bipartite planar graph B may not be 2-edge-connected, we have that

$$e \leq \frac{5}{4}(b - 2) + \frac{9}{4}a,$$

unless B has only one vertex.

Proof: If B has no edge then the condition $e = 0 \leq \frac{5}{4}(b-2) + \frac{9}{4}a$ is equivalent to $9a + 5b \geq 10$ which holds if B has at least 2 vertices.

If B is a non-empty forest then $e \leq a+b-1 \leq \frac{5}{4}(b-2)+\frac{9}{4}a$, the latter inequality being equivalent to $6 \leq 5a + b$ which holds because B has at least one vertex on each side of the bipartition.

If B is not a forest it has a non-trivial block and we can apply Lemma 5.2 to each such block. We claim that the stronger bound of $\frac{9}{4}a + \frac{5}{4}b - \frac{9}{2}$ of Lemma 5.2 holds. B can be built by successively adding blocks and/or edges to a first non-trivial block (and maintaining connectivity). The claim holds for the first non-trivial block. When we add another non-trivial block, the number of edges increases by at most $\frac{9}{4}\Delta a + \frac{5}{4}\Delta b - \frac{9}{4}$, where Δa (resp. Δb) represents the increase in the number of vertices in V_1 (resp. V_2); we have used the fact that the new block shares only one vertex with the rest. If we add an edge, this one edge can be upper bounded by $\frac{9}{4}\Delta a + \frac{5}{4}\Delta b - \frac{1}{4}$ since either Δa or Δb is one. Summing all these contributions, we derive the claim. ■

We can now continue the proof of Theorem 5.1 for the case in which $\mathcal{M}$ does not contain two cycles with more than one point in common. From Lemma 5.3 and the discussion of the previous section, we derive that

$$\sum_{C \in \mathcal{M}_{\mathbf{v}}} |F_{\mathbf{v}} \cap C| \leq \frac{9}{4}|\mathcal{M}_{\mathbf{v}}| + \frac{5}{4}(deg(\mathbf{v}) - 2), \tag{3}$$

unless the bipartite graph B for node $\mathbf{v}$ has only one vertex. This can only happen for the root node $\mathbf{r}$, and only if $deg(\mathbf{r}) = 1$ and $\mathcal{M}_{\mathbf{r}} = \emptyset$, in which case we have to increase the RHS by $\frac{5}{4}$. We will refer to this case as the pathological case. Summing over all nodes of $\mathbf{T}$, we derive that

$$\begin{aligned}\sum_{C \in \mathcal{M}} |F \cap C| &\leq \frac{9}{4}|\mathcal{M}| + \frac{5}{4}\sum_{\mathbf{v}}(deg(\mathbf{v}) - 2) \\ &= \frac{9}{4}|\mathcal{M}| - \frac{5}{2}, \end{aligned} \tag{4}$$

except in the pathological case for which

$$\sum_{C \in \mathcal{M}} |F \cap C| \leq \frac{9}{4}|\mathcal{M}| - \frac{5}{4}. \tag{5}$$

Now consider the slightly more general case in which $\mathcal{V} = \mathcal{M}$. This now includes the situation in which there are cycles in $\mathcal{M}$ with more than one vertex in common, but these cycles do not surround other cycles of $\mathcal{M}$. Consider any minimal solution F. Consider two cycles C_1 and C_2 with more than one vertex in common. If F does not contain more than one of these common vertices and if this happens to be true for all such pairs of cycles C_1, C_2, then the proof above is unchanged since the bipartite graph B still does not have any cycle of length 4. Assume now that there are two cycles $C_1, C_2 \in \mathcal{M}$ with more than one vertex in common belonging to F. We first claim that there must be exactly two vertices of F in $C_1 \cap C_2$. Indeed, if there were three (or more), one of their witness cycles would be contained in one of the interior pockets, contradicting the emptyness of these pockets. Furthermore, observe that no witness cycle can contain C_1 but not C_2: such a witness cycle would pass through all vertices in $C_1 \cap C_2$ which contradicts the fact that a witness cycle has only one vertex of F. We now modify $\mathcal{M}$ by replacing C_1 and C_2 by the cycle C whose interior consists of $f(C_1)$, $f(C_2)$ and the interior pockets between C_1 and C_2. This cycle may not be a cycle in $\mathcal{C}$, but we will not need this. The important fact is that the new family $\mathcal{M}'$ still obeys properties 1 and 2, and therefore we can still apply the previously established results. While replacing $\mathcal{M}$ by $\mathcal{M}'$, we have decreased $|\mathcal{M}|$ by 1, and have decreased $\sum_{C \in \mathcal{M}} |F \cap C|$ by exactly 2. By induction, we can therefore prove inequality (4), or (5) in the pathological case.

Finally, let us consider the general case in which $\mathcal{M} \neq \mathcal{V}$, and let u and v be the two vertices on C_1 and C_2 which define $\mathcal{V}$. We claim that the family $\mathcal{V}$ almost satisfies properties 1 and 2; the only difference is that the witness cycles of u and/or v (if they belong to F) may not contain a cycle of $\mathcal{V}$. For the purpose of the analysis (we cannot do this algorithmically), assume we add to $\mathcal{V}$ (at most two) cycles of $\mathcal{M}$ to guarantee that this enlarged family satisfies properties 1 and 2. Therefore we can now use (4) and/or (5). But in enlarging $\mathcal{V}$, we have increased its size by $t \leq |F \cap \{a, b\}|$ and increased $\sum_{C \in \mathcal{V}} |F \cap C|$ also by t. Thus, rewriting (4), we derive that

$$(\sum_{C \in \mathcal{V}} |F \cap C|) + t \leq \frac{9}{4}(|\mathcal{V}| + t) - \frac{5}{2},$$

or

$$\sum_{C \in \mathcal{V}} |F \cap C| \leq \frac{9}{4}|\mathcal{V}| + \frac{5}{4}t - \frac{5}{2} \leq \frac{9}{4}|\mathcal{V}|,$$

since $t \leq 2$. In the pathological case, however, we need to be a bit more careful. We claim that in the pathological case, $t \leq 1$. Indeed, if t was 2, the witness cycles of u and v would both be maximal sets in the laminar family of witness cycles, and thus the root node $\mathbf{r}$ would have degree at least 2, contradicting the definition of the pathological case. But, the same derivation then shows that in the pathological case we also have:

$$\sum_{C \in \mathcal{V}} |F \cap C| \leq \frac{9}{4}|\mathcal{V}|.$$

This concludes the proof of Theorem 5.1.

The performance guarantee of 9/4 is tight for D-FVS, S-FVS and BIP, but again we are not aware of an instance with a performance worse than 2 for FVS. As before, we have a single class of instances applying to all three problems; we show a sample instance in Figure 3. This time, there are $3k-1$ white vertices and they each have a weight of 3, and the other (black) vertices have a weight of $1+\epsilon$. As in the previous figure, the special vertices for S-FVS are denoted by (black) squares, while for D-FVS the orientation of the arcs along $k+2$ of the faces are explicitly given on the figure and the orientation of the other arcs are such that the shaded faces define directed cycles. The face-minimal cycles are the boundaries of the shaded faces, and since no pockets are formed by any pairs of face-minimal cycles, the violation oracle will return all the face-minimal cycles. The algorithm will then select all white vertices in the solution for a total weight of $9k-3$. However, in all three cases, the black squares constitute a feasible solution of weight $(4k+1)(1+\epsilon)$, giving the desired bound as k gets large and ϵ tends to 0.

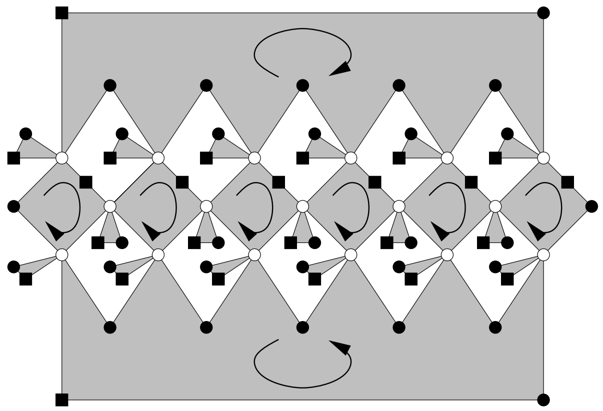

Figure 3: A bad example for the 9/4-approximation algorithm applied to BIP, to D-FVS, or to S-FVS.

6 Worst-case duality gaps

In this section, we discuss the worst-case ratio between the value of the problem considered and the optimum value of the linear programming relaxation of (IP) (or the value of its dual (D)), the worst-case being taken over all non-negatively weighted planar instances. The results of the previous section immediately imply that this worst-case ratio ρ is at most 9/4 for any uncrossable hitting cycle problem.

Before considering the worst-case ratio for hitting cycle problems in more detail, we investigate the vertex cover problem. In the vertex cover problem, one would like to find a minimum-weight set of vertices S such that for every edge at least one of its endpoints is in S. A classical linear programming relaxation of this problem is given below:

$$
(LP) \qquad \begin{array}{lll} & \text{Min} \quad \sum_{i \in V} w_i x_i & \\ \text{subject to:} & & \\ & x_i + x_j \geq 1 & (i,j) \in E \\ & x_i \geq 0 & i \in V. \end{array}
$$

It is well-known that the ratio between the value of the vertex cover problem and the value of (LP) is upper bounded by 2, and this can be approached arbitrarily closely by general graphs. However, we show below that the worst-case ratio is exactly 3/2 for planar instances by using the 4-color theorem.

Theorem 6.1 For planar graphs, $\rho_{VC} = \frac{3}{2}$.

Proof: For K_4 with unit weights, the minimum vertex cover has size 3, but the LP value is 2 and this is obtained by setting all x_i's to 0.5. This shows that $\rho_{VC} \geq \frac{3}{2}$.

To prove the other inequality, we use the 4-color theorem and a result about the structure of the extreme points of (LP). It is known that at the extreme points of (LP), $x_i \in \{0, \frac{1}{2}, 1\}$ for all i [23]. Given a four-coloring of the graph and an optimal extreme point of (LP), we find the color class $\mathcal{X}$ which maximizes $\sum_{i \in \mathcal{X}: x_i = 1/2} w_i$. Consider then the integral solution

$$
x_i^* = \begin{cases} 1 & \text{if } x_i = 1 \text{ or } x_i = \frac{1}{2}, i \notin \mathcal{X} \\ 0 & \text{if } x_i = 0 \text{ or } x_i = \frac{1}{2}, i \in \mathcal{X} \end{cases}
$$

By construction $\sum_i w_i x_i^* \leq \frac{3}{2} \sum_i w_i x_i$. Furthermore, x^* corresponds to a vertex cover since for any edge (i,j) with $x_i = x_j = \frac{1}{2}$, both i and j cannot be in $\mathcal{X}$. ■

A proof of this result not based on the 4-color theorem would be very nice. Indeed, since the solution $x_i = 0.5$ for all i is always feasible for the linear programming relaxation, the above theorem implies the existence of a vertex cover of size at most $3n/4$ (or an independent set of size at least $n/4$), which follows immediately from the 4-color theorem, but no other proof of this result is known.

The K_4 instance for the vertex cover problem leads to bad instances for many hitting cycle problems. Consider FVS, for example. If we replace in K_4 every edge by a triangle (introducing one new vertex) and if we keep all weights to be equal to 1, then the optimum solution still has value 3, and a feasible solution to the linear programming relaxation of the hitting cycle formulation (IP) can be obtained by setting the original vertices to have $x_i = 0.5$ and the new vertices to have $x_i = 0$. This shows that the worst-case ratio ρ_{FVS} for FVS on planar instances is at least $\frac{3}{2}$. The same construction shows that that $\rho_{BIP} \geq 3/2$ and $\rho_{D-FVS} \geq 3/2$ for BIP and D-FVS both in the planar case.

We can get a larger lower bound on ρ_{FVS} by considering an appropriately weighted instance of $K_{2,p}$. Let the two vertices on one side of the bipartition each have weight $p-1$, and the p vertices on the other side each have weight 1. An optimal solution to this instance has weight $p-1$. A feasible solution to the linear programming relaxation of (IP) has $x_i = 0.5$ for the p vertices and $x_i = 0$ for the two vertices. This proves that $\rho_{FVS} \geq 2 - \frac{2}{p}$, which tends to 2 as p becomes large.

Conjecture 6.2 $\rho_{D-FVS} = \frac{3}{2}$, $\rho_{BIP} = \frac{3}{2}$, and $\rho_{FVS} = 2$.

These ratios have an interesting connection with a conjecture of Akiyama and Watanabe [2] and Albertson and Berman [3]. They conjectured that a planar graph has a feedback vertex set of size at most $n/2$. Since the solution with $x_i = 1/3$ is feasible for the LP relaxation of (IP), this implies the existence of a feedback vertex set of size at most $\rho_{FVS} n/3$. A coloring result of Borodin [7] shows that any planar graph has a feedback vertex set of size no more than $3n/5$; however, Jensen and Toth [19, p. 6] call the proof reminiscent of the proof of the 4-color theorem, partly because it involves 450 reducible configurations. We think it would be interesting to derive results along these lines that do not invoke the four-color theorem or similar theorems. Akiyama and Watanabe also conjectured that in bipartite planar graphs, there exists a feedback vertex set of size at most $3n/8$. Since $x_i = 1/4$ is feasible for the LP relaxation if the graph is bipartite, this implies the existence of a feedback set of size $\rho_{FVS} n/4$. For BIP, the conjecture that $\rho_{BIP} = 3/2$ would imply the existence of at most $n/2$ vertices whose removal makes the graph bipartite. This follows easily from the 4-color theorem (removing the two smallest color classes), but once again we are not aware of any proof of this statement not based on the 4-color theorem. We should point out that in the worst case one cannot remove less than half the vertices for either FVS or BIP (consider K_4 or multiple copies of K_4). For D-FVS on simple planar digraphs, the same reasoning would imply the existence of a feedback vertex set of size at most $n/2$, which would follow clearly from Akiyama and Watanabe's or Albertson and Berman's conjecture. It seems possible in fact that $n/3$ vertices are enough for simple digraphs.

7 Implementation

We first sketch how our 3-approximation algorithms can be implemented in $O(n^2)$ time, where $n = |V|$. For all problems considered, the FACE-MINIMAL oracle can easily be implemented in linear time as follows. For the three undirected problems (FVS, S-FVS and BIP), we can first decide whether the boundary of any face is a cycle of $\mathcal{C}$ in time proportional to the length of this cycle. Over all faces, this gives a linear running time to compute the set $\mathcal{M}$ of face-minimal cycles in $\mathcal{C}$ (since the total length of all faces is equal to twice the number of edges, which is at most $3n - 6$). To implement the FACE-MINIMAL oracle in the case of D-FVS, we consider the planar dual G^* of the graph G. It is not difficult to see that the face-minimal cycles correspond to sources and sinks in a DAG formed by contracting the strongly connected components of G^*. The planar dual, its strongly connected components and the sources and sinks can easily be found in linear time, and as a result we can implement FACE-MINIMAL in linear time also for D-FVS. Notice that the FACE-MINIMAL oracle can also be used to implement the "clean-up" phase (line 10 of Figure 1): a set S is feasible if the oracle does not return any cycle. As we build $\mathcal{M}$ for any of these problems, we can also compute for each vertex v the quantity $r(v) = |\{C \in \mathcal{M} : v \in C\}|$ which represents the rate of growth of the left-hand-side of the dual constraint corresponding to v. This is useful in order to select the next vertex to add to S. Indeed, if we keep track of $a(v) = \sum_{C: v \in C} y_C$ for each vertex v then the next vertex selected by the algorithm is the one minimizing $\epsilon = \min_v (w_v - a(v))/r(v)$. We can then update $a(v)$ by setting $a(v) \leftarrow a(v) + \epsilon \cdot r(v)$. As we add a vertex to S (and remove it from the graph), we can easily update the planar graph in linear time as well. Since both loops of Figure 1 are executed $O(n)$ times, this gives a total running time of $O(n^2)$.

To get the 9/4-approximation algorithm, we have to describe how to find the appropriate subset of cycles returned by FACE-MINIMAL. We claim that this subset can be found in $O(n^2)$ time, leading to an overall running time of $O(n^3)$. We focus on the undirected problems, even though D-FVS can be treated similarly by first computing the strongly connected components of G^*. By abuse of notation, let $\mathcal{M}$ denote the vertices of G^* that correspond to face-minimal cycles in $\mathcal{C}$, and let

o denote the vertex of G^* corresponding to the outer face. We use the following characterization: two cycles in $\mathcal{M}$ corresponding to vertices $u, v \in G^*$ induce a non-empty pocket if and only if $G^* - \{u, v\}$ has a connected component containing a vertex of $\mathcal{M}$ and not containing o. For a given $u \in \mathcal{M}$, we can thus consider $G^* - \{u\}$, compute its block structure and find the cutvertex $v \in \mathcal{M}$ furthest away from the component containing o such that $G^* - \{u\} - \{v\}$ has a connected component containing a vertex in $\mathcal{M}$ and not o. This can be done in linear time. If we then select among all vertices $u \in \mathcal{M}$ the one that induces such a connected component of smallest size, we are then guaranteed that the vertices in $\mathcal{M}$ in that connected component form a suitable choice for $\mathcal{V}$. Finding $\mathcal{V}$ thus takes $O(n^2)$ time.

8 Conclusion

The most pressing question left open by this work is whether one can derive an 2-approximation algorithm for FVS in planar graphs using the primal-dual technique on the cycle formulation. Such a result would immediately imply that planar graphs have feedback vertex sets of size at most $2n/3$, which we think would be interesting since alternate proofs invoke the four color theorem or similar results. To prove such a result, one would "simply" need to find some subset of cycles $\mathcal{N}$ such that for any minimal fvs F, $\sum_{C \in \mathcal{N}} |F \cap C| \leq 2|\mathcal{N}|$. Note that in order to prove a bound on the size of a feedback vertex set, the subset would not necessarily have to be polynomial-time computable.

An additional open question is whether the time complexity of our algorithms can be made linear or near-linear.

Acknowledgements

We thank Seffi Naor for pointing out reference [26], and Jon Kleinberg for helpful discussions.

References

[1] A. Agrawal, P. Klein, and R. Ravi. When trees collide: An approximation algorithm for the generalized Steiner problem on networks. *SIAM Journal on Computing*, 24:440–456, 1995.

[2] Akiyama and Watanabe. Research problem. *Graphs and Combinatorics*, 3:201–202, 1986.

[3] M. Albertson and D. Berman. A conjecture on planar graphs. In J. Bondy and U. Murty, editors, *Graph Theory and Related Topics*. Academic Press, 1979.

[4] V. Bafna, P. Berman, and T. Fujito. Constant ratio approximation of the weighted feedback vertex set problem for undirected graphs. In J. Staples, P. Eades, N. Katoh, and A. Moffat, editors, *ISAAC '95 Algorithms and Computation*, number 1004 in Lecture Notes in Computer Science, pages 142–151, 1995.

[5] R. Bar-Yehuda, D. Geiger, J. Naor, and R. M. Roth. Approximation algorithms for the vertex feedback set problem with applications to constraint satisfaction and Bayesian inference. In *Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 344–354, 1994.

[6] A. Becker and D. Geiger. Approximation algorithms for the loop cutset problem. In *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, pages 60–68, 1994.

[7] O. Borodin. On acyclic colorings of planar graphs. *Discrete Mathematics*, 25:211–236, 1979.

[8] F. A. Chudak, M. X. Goemans, D. S. Hochbaum, and D. P. Williamson. A primal-dual interpretation of recent 2-approximation algorithms for the feedback vertex set problem in undirected graphs. Unpublished, 1996.

[9] G. Even, J. Naor, B. Schieber, and M. Sudan. Approximating minimum feedback sets and multi-cuts in directed graphs. In E. Balas and J. Clausen, editors, *Integer Programming and Combinatorial Optimization*, volume 920 of *Lecture Notes in Computer Science*, pages 14–28. Springer-Verlag, 1995.

[10] G. Even, J. Naor, B. Schieber, and L. Zosin. Approximating minimum subset feedback sets in undirected graphs with applications. In *Proceedings of the Fourth Israel Symposium on Theory of Computing and Systems*, pages 78–88, 1996.

[11] G. Even, J. Naor, and L. Zosin. An 8-approximation algorithm for the subset feedback vertex set problem. To appear in FOCS '96, 1996.

[12] M. R. Garey and D. S. Johnson. *Computers and Intractability.* W.H. Freeman and Company, New York, 1979.

[13] N. Garg, V. Vazirani, and M. Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees, with applications to matching and set cover. In *Proceedings of the 20th International Colloquium on Automata, Languages and Programming*, 1993. To appear in *Algorithmica* under the title "Primal-dual approximation algorithms for integral flow and multicut in trees".

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

[15] M. X. Goemans and D. P. Williamson. A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, 24:296–317, 1995.

[16] M. X. Goemans and D. P. Williamson. The primal-dual method for approximation algorithms and its application to network design problems. In D. S. Hochbaum, editor, *Approximation Algorithms for NP-hard Problems*, chapter 4, pages 144–191. PWS, Boston, 1997.

[17] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer-Verlag, Berlin, 1988.

[18] F. Hadlock. Finding a maximum cut of a planar graph in polynomial time. *SIAM Journal on Computing*, 4:221–225, 1975.

[19] T. R. Jensen and B. Toft. *Graph Coloring Problems.* John Wiley and Sons, New York, 1995.

[20] P. Klein, S. Rao, A. Agrawal, and R. Ravi. An approximate max-flow min-cut relation for undirected multicommodity flow, with applications. *Combinatorica*, 15:187–202, 1995.

[21] P. Klein and R. Ravi. When cycles collapse: A general approximation technique for constrained two-connectivity problems. In *Proceedings of the Third MPS Conference on Integer Programming and Combinatorial Optimization*, pages 39–55, 1993. Also appears as Brown University Technical Report CS-92-30. To appear in *Algorithmica.*

[22] T. Leighton and S. Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *Proceedings of the 29th Annual Symposium on Foundations of Computer Science*, pages 422–431, 1988.

[23] G. L. Nemhauser and L. E. Trotter Jr. Vertex packing: Structural properties and algorithms. *Mathematical Programming*, 8:232–248, 1975.

[24] G. I. Orlova and Y. G. Dorfman. Finding the maximal cut in a graph. *Engineering Cybernetics*, pages 502–506, 1972.

[25] P. D. Seymour. Packing directed circuits fractionally. *Combinatorica*, 15:281–288, 1995.

[26] H. Stamm. On feedback problems in planar digraphs. In R. Möhring, editor, *Graph-Theoretic Concepts in Computer Science*, number 484 in Lecture Notes in Computer Science, pages 79–89. Springer-Verlag, 1990.

[27] D. P. Williamson, M. X. Goemans, M. Mihail, and V. V. Vazirani. A primal-dual approximation algorithm for generalized Steiner network problems. *Combinatorica*, 15:435–454, 1995.

[28] M. Yannakakis. Node and edge-deletion NP-complete problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing*, pages 253–264, May 1978.