ELSEVIER

Operations Research Letters 22 (1998) 111–118

operations research letters

A primal–dual interpretation of two 2-approximation algorithms for the feedback vertex set problem in undirected graphs

Fabián A. Chudak [a,1], Michel X. Goemans [b,2], Dorit S. Hochbaum [c,3], David P. Williamson [d,*]

[a] *School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY 14853, USA*
[b] *CORE, 34 Voie du Roman Pays, B-1348 Louvain-La-Neuve, Belgium*
[c] *University of California, Department of IEOR, 4135 Etcheverry Hall, Berkeley, CA 94720, USA*
[d] *IBM T.J. Watson Research Center, Room 33-219, P.O. Box 218, Yorktown Heights, NY 10598, USA*

Received 1 August 1996; revised 1 March 1998

Abstract

Recently, Becker and Geiger and Bafna, Berman and Fujito gave 2-approximation algorithms for the feedback vertex set problem in undirected graphs. We show how their algorithms can be explained in terms of the primal–dual method for approximation algorithms, which has been used to derive approximation algorithms for network design problems. In the process, we give a new integer programming formulation for the feedback vertex set problem whose integrality gap is at worst a factor of two; the well-known cycle formulation has an integrality gap of $\Theta(\log n)$, as shown by Even, Naor, Schieber and Zosin. We also give a new 2-approximation algorithm for the problem which is a simplification of the Bafna et al. algorithm. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: Approximation algorithms; Combinatorial optimization; Feedback vertex set

Given an undirected graph $G=(V,E)$ and non-negative weights w_v for the vertices $v \in V$, the minimum-weight feedback vertex set problem (FVS) is to find a minimum-weight set of vertices F that meets every cycle of G. Alternatively, the problem is to find a minimum-weight F such that $G[V-F]$ is acyclic, where $G[S]$ denotes the subgraph of G induced by the vertex set S. We say that a set of vertices F is an *fvs* if it is a feasible solution to the problem.

The minimum-weight feedback vertex set problem has long been known to be NP-hard (see [5], Problem GT7); hence, researchers have attempted to find *approximation algorithms* for the problem. An α-approximation algorithm for FVS runs in polynomial time and finds an fvs whose weight is no more than α times the weight of an optimal fvs. The value α is sometimes called the *performance guarantee* of the algorithm. The first non-trivial approximation algorithm was given

* Corresponding author.

[1] Research partially supported by NSF grants CCR-9307391 and DMI-9157199.

[2] On leave from MIT. Research supported in part by NSF contracts 9302476-CCR and 9623859-CCR, DARPA contract N00014-92-J-1799, and a Sloan fellowship.

[3] Research supported in part by ONR contract N00014-91-J-1241.

by Bar-Yehuda, Geiger, Naor and Roth [2], and has a performance guarantee of $\log n$, where n is the number of vertices. Recently, two slightly different 2-approximation algorithms were given by Bafna, Berman and Fujito [1] and Becker and Geiger [3].

The goal of this note is to present the two 2-approximation algorithms in terms of the primal–dual method for approximation algorithms. This method has been very successful in deriving approximation algorithms for network design problems (see Goemans and Williamson [7] for a survey); it has also been used to design approximation algorithms for feedback-style problems in planar graphs [6]. The method begins with an integer programming formulation of the problem under consideration. The standard primal–dual algorithm then simultaneously constructs a feasible integral solution and a feasible solution to the dual of the linear programming relaxation. If it can be shown that the value of these two solutions is within a factor of α, then an α-approximation algorithm is obtained. This approach has also been called the "dual feasible" approach (see [8]).

The *integrality gap* of an integer program is the worst-case ratio between the optimum value of the integer program and the optimum value of its linear relaxation. A consequence of the primal–dual method is a proof that the integrality gap of the integer program under consideration is bounded by α. As part of our proof, we give a new integer programming formulation for FVS, along with a proof that its integrality gap is at most two. A previous integer programming formulation of FVS (the well-known "cycle formulation") is known to have an integrality gap of $\Theta(\log n)$ [4].

Finally, we also give a new 2-approximation algorithm for FVS which is a simplification of the algorithm of Bafna et al. in that it does not involve searching for "semidisjoint" cycles.

This note is structured as follows. In Section 1, we prove a series of key inequalities which we need for the proofs of the algorithms. In Section 2, we give a new integer programming formulation of FVS. In Section 3, we show how the inequalities, the formulation, and the primal-dual method give the 2-approximation algorithm of Bafna et al. Section 4 gives our new 2-approximation algorithm. In Section 5 we show how a slight modification of previous arguments leads to the algorithm of Becker and Geiger.

1. Some inequalities

We begin by giving some inequalities we will need in proving the performance guarantees of the algorithms and in giving the new integer programming formulation. We first need some notation and terms. For a given graph $G=(V,E)$, we let τ denote the *cardinality* of the smallest fvs for G. Let $d(v)$ denote the degree of vertex v in G. Given a subset S of vertices, let $E[S]$ denote the subset of edges that have both endpoints in S. Let $G[S]$ denote the subgraph $(S,E[S])$ induced by S, and let $d_S(v)$ denote the degree of v in $G[S]$. We let $b(S)=|E[S]|-|S|+1$ (so that $b(V)=|E|-|V|+1$).

We say that an fvs F is *minimal* if for any $v\in F$, $F-v$ is not an fvs. An fvs F is *almost minimal* if there is at most one $v\in F$ such that $F-v$ is an fvs. Following [1], we say that a cycle is *semidisjoint* if it contains at most one vertex of degree greater than 2.

We can now state our theorem.

Theorem 1.1. *Let F denote any fvs of a graph $G=(V,E)$, where $E\neq\emptyset$. Then*

$$\sum_{v\in F}[d(v)-1]\geqslant b(V), \tag{1}$$

$$\sum_{v\in F}d(v)\geqslant b(V)+\tau. \tag{2}$$

If every vertex of G has degree at least two, and F_{M} is any minimal fvs, then

$$\sum_{v\in F_{\mathrm{M}}}d(v)\leqslant 2(b(V)+\tau)-2. \tag{3}$$

If every vertex has degree at least two and the graph contains no semidisjoint cycles or is itself a cycle, and F_{AM} is any almost minimal fvs, then

$$\sum_{v\in F_{\mathrm{AM}}}[d(v)-1]\leqslant 2b(V)-1. \tag{4}$$

Observe that inequalities (2) and (3) imply

$$\sum_{v\in F_{\mathrm{M}}}d(v)\leqslant 2\sum_{v\in F}d(v)-2, \tag{5}$$

while inequalities (1) and (4) imply

$$\sum_{v\in F_{AM}}[d(v)-1]\leqslant 2\sum_{v\in F}[d(v)-1]-1. \qquad (6)$$

Observe also that the condition under which inequality (4) holds is satisfied if the graph is 2-vertex-connected. Inequalities (2) and (1) will be used in giving new integer programming formulations, while inequalities (4) and (3) will be used to prove the performance guarantees of the various algorithms. Inequality (1) is stated in Lemma 3 of Bafna et al. [1], while inequality (4) is a strengthening of their Lemma 4. Inequality (5) is Theorem 4 of Becker and Geiger [3]. Our proofs here are somewhat different from those given in [1, 3].

Proof. Inequality (2) clearly follows from inequality (1). To prove inequality (1), we consider two cases. If $F=V$ then the hypothesis that $E\neq\emptyset$ ensures that the inequality holds. If $F\neq V$, observe that the removal of F from G gives an acyclic subgraph. The number of edges in this subgraph is thus less than its number of vertices, i.e. this subgraph contains at most $|V|-|F|-1$ edges. Moreover, by removing the vertices in F, we have removed at most $\sum_{v\in F}d(v)$ edges (the "at most" comes from the fact that an edge could be counted twice in the sum of the degrees). The total number of edges being $|E|$, we therefore derive that $|V|-|F|-1+\sum_{v\in F}d(v)\geqslant|E|$. Rearranging the terms gives the desired inequality.

Our proofs of the two remaining inequalities are similar in structure, and hence we use F to denote either F_M or F_{AM}, as appropriate for the particular inequality. We first observe that $\sum_{v\in V}d(v)=2|E|$. Let k be the number of connected components of $G[V-F]$ and note that the edges in $G[V-F]$ contribute exactly $2(|V|-|F|-k)$ to $\sum_{v\notin F}d(v)$. By these observations and rearranging terms, inequalities (3) and (4) can be rewritten as

$$|\delta(F_M)|\geqslant 2|F_M|+2k-2\tau, \qquad (7)$$

$$|\delta(F_{AM})|\geqslant |F_{AM}|+2k-1, \qquad (8)$$

respectively, where $\delta(S)$ is the set of edges with exactly one endpoint in S. To prove these inequalities, we consider the weighted bipartite graph H obtained by shrinking in G every connected component of $G[V-F]$ and by removing all the edges of $G[F]$. The weight of an edge is the number of edges from the respective node in F to the nodes in the respective connected component of $G[V-F]$. For the two inequalities, we need to show that the total weight of this bipartite graph H is at least $2|F_M|+2k-2\tau$, or $|F_{AM}|+2k-1$, respectively.

We first observe that for each vertex $v\in F_M$, there must be some *witness cycle* C_v of G such that $C_v\cap F_M=\{v\}$; otherwise F_M would not be minimal. Thus in H there must be at least one edge of weight at least 2 incident to every vertex of F_M; we designate one such edge for each vertex in F_M and call it a *primary* edge. The statement is also true for all vertices in $F_{AM}-x$, where x is the one vertex of F_{AM} (if any) such that $F_{AM}-x$ is an fvs. Indeed, by the definition of an almost minimal fvs, for any $w\in(F_{AM}-x)$, $F_{AM}-w$ is not an fvs implying the existence of a witness cycle C_w such that $C_w\cap F_{AM}=\{w\}$.

To prove inequality (7), let T be a maximum collection of vertex-disjoint witness cycles, and let $\tilde{F}$ denote the vertices of F_M whose witness cycles are not in T. Notice that $|T|\leqslant\tau$, so that the inequality is implied if the weight of H is at least $2|\tilde{F}|+2k$. By the properties of T, it must be the case that for every connected component of $G[V-F_M]$ adjacent to a vertex $v\in\tilde{F}$ via a primary edge, it must also be adjacent to a vertex of $F_M-\tilde{F}$ via a primary edge, since otherwise we would be able to add the witness cycle of v to T. Thus, if we remove the primary edges adjacent to the vertices of $\tilde{F}$ from H (which account for weight at least $2|\tilde{F}|$), there still must be weight two adjacent to each component of $G[V-F_M]$. Hence the weight of edges in H is at least $2|\tilde{F}|+2k$.

To prove inequality (8), we first observe that if G is a cycle then $|F_{AM}|$ and k must be 1, and hence inequality (8) is satisfied. Now, we consider the case $k=1$, where we need to prove that

$$|\delta(F_{AM})|\geqslant|F_{AM}|+1. \qquad (9)$$

The existence of at least $|F_{AM}|-1$ primary edges imply that $|\delta(F_{AM})|\geqslant 2(|F_{AM}|-1)\geqslant|F_{AM}|+1$, if $|F_{AM}|\geqslant 3$. On the other hand, if $|F_{AM}|=1$, inequality (9) follows from the fact that every vertex has degree at least 2, while for $|F_{AM}|=2$, inequality (9) follows from the existence of a primary edge and the fact that

$x \in F_{\text{AM}}$ cannot be adjacent only to the other vertex in F_{AM}.

Now, assume that $k > 1$. First decrease the weight of each primary edge by one; this accounts for $|F_{\text{AM}}| - 1$ of the weight of the edges of the graph. We claim that, incident to every connected component of $G[V - F_{\text{AM}}]$, there are either at least two (distinct) edges (independent of their weight) or one edge of weight at least 3 in H. Then there must be weight 2 remaining incident to each connected component, and the overall total weight is therefore at least $|F_{\text{AM}}| + 2k - 1$. The claim follows since if some connected component C of $G[V - F_{\text{AM}}]$ had a unique neighbor in F, say v, through an edge of weight at most 2, then since no vertex can have degree 1, the component C and v must form a semidisjoint cycle in G. □

2. A new integer programming formulation

Having shown the inequalities of the previous section, we turn to giving a new integer programming formulation of the minimum-weight feedback vertex set problem in undirected graphs. The standard cycle formulation of the problem is as follows:

$$\begin{aligned} (CYC)\ \text{Min.} & \sum_{v \in V} w_v x_v \\ \text{s.t.} & \sum_{v \in C} x_v \geqslant 1, \quad \forall C \in \mathscr{C}, \\ & x_v \in \{0,1\}, \quad v \in V, \end{aligned}$$

where $\mathscr{C}$ is the set of all node sets C of all cycles of the graph. Even, Naor, Schieber and Zosin [4] have shown that the integrality gap of this integer program is $\Omega(\log n)$; Bar-Yehuda et al. [2] had previously shown that it was $O(\log n)$.

Observe that if F is a feedback vertex set for G, then clearly $F \cap S$ is a feedback vertex set for $G[S]$. Hence we have the following corollary of inequality (1).

Corollary 2.1. *Let F be any feedback vertex set. Then for any $S \subseteq V$ such that $E[S] \neq \emptyset$,*

$$\sum_{v \in F \cap S} (d_S(v) - 1) \geqslant |E[S]| - |S| + 1 = b(S).$$

Our new integer programming formulation is as follows:

$$\begin{aligned} (IP)\ \text{Min.} & \sum_{v \in V} w_v x_v \\ \text{s.t.} & \\ & \sum_{v \in S} (d_S(v) - 1) x_v \geqslant b(S), \quad S \subseteq V\colon E[S] \neq \emptyset, \\ & x_v \in \{0,1\}, \quad v \in V. \end{aligned}$$

By Corollary 2.1, clearly any fvs is a feasible solution to the integer program. To see that any feasible integer solution x must be an fvs, suppose the contrary: namely, that there is some cycle on the set of vertices C such that $x_v = 0$ for all $v \in C$. Then consider the constraint corresponding to the set of vertices C. The left-hand side of the constraint must be 0, while since $E[C]$ contains a cycle, $|E[C]| \geqslant |C|$. Thus the right-hand side of the constraint is at least 1, contradicting the feasibility of x. This proves that (IP) is an integer programming formulation of the minimum-weight feedback vertex set problem.

3. The Bafna-Berman-Fujito algorithm

We now give a primal–dual 2-approximation algorithm using our integer programming formulation. This algorithm is essentially the algorithm of Bafna et al.

In our algorithm we construct a feasible solution to the dual of the linear programming relaxation of (IP). The linear programming relaxation is

$$\begin{aligned} (LP)\ \text{Min.} & \sum_{v \in V} w_v x_v \\ \text{s.t.} & \\ & \sum_{v \in S} (d_S(v) - 1) x_v \geqslant b(S), \quad S \subseteq V\colon E[S] \neq \emptyset, \\ & x_v \geqslant 0, \quad v \in V, \end{aligned}$$

and its dual is

$$(D)\ \text{Max.} \sum_S b(S)y_S$$

$$\text{s.t.}$$

$$\sum_{S:v\in S}(d_S(v)-1)y_S \leqslant w_v, \qquad v\in V,$$

$$y_S \geqslant 0, \qquad S\subseteq V\colon E[S]\neq\emptyset.$$

The algorithm is given in Fig. 1. The three algorithms we present have the same structure but differ in a single subroutine (called Violation); the subroutine used in the Bafna et al. algorithm is given in version A. The overall algorithm starts with $F=\emptyset$, the feasible dual solution $y=0$, and the original graph $(V',E')=(V,E)$. Given a set F, if it is not a fvs, there must exist a cycle in (V',E'). We first recursively remove degree one vertices and incident edges from V' and E'. We now choose some set S that corresponds to a violated constraint of (IP); we call S a "violated set". To choose S, we call the subroutine Violation. In [1], there are two cases for a choice of S. The subroutine first looks for a semidisjoint cycle in (V',E'). If it finds such a cycle, it lets S correspond to the vertices of the cycle, and returns S. Otherwise, it returns $S=V'$. The algorithm then increases the dual variable y_S as much as possible until some dual inequality becomes tight for some vertex in S, say for vertex v; that is, $\sum_{T:v\in T}(d_T(v)-1)y_T=w_v$. The algorithm adds vertex v to F. It then removes vertex v from V' and attached edges from E', and continues. When F is an fvs, the algorithm goes through the vertices of F in the reverse of the order in which they were added, and removes any extraneous vertices (that is, when vertex v is considered, we remove it from F if we still have a feasible fvs without v). We let F' denote the final fvs given by the algorithm.

The primal–dual structure of this algorithm (i.e., increasing duals, choosing the solution elements corresponding to the tight dual constraints, performing a final "clean-up" step in reverse) is the same as that used in a number of other algorithms for rather different problems (e.g., [9, 10, 6]).

It is not difficult to see that this algorithm is effectively equivalent to the following: Start with $F=\emptyset$ and the graph G. Look first for a semidisjoint cycle S; if none is found, set S to be the remaining vertices. Pick the vertex $v\in S$ that achieves the minimum $\varepsilon=\min_{v\in S} w_v/(d_S(v)-1)$. Add v to F, and set $w_u\leftarrow w_u-\varepsilon(d_S(u)-1)$ for all $u\in S$. Remove v from the graph, then recursively remove all degree one vertices and incident edges. Repeat until F is a fvs. Perform the "reverse delete" step to obtain F'. This algorithm is essentially the Bafna et al. algorithm. A straightforward implementation of this

```
1 $y \leftarrow 0$
2 $F \leftarrow \emptyset$
3 $l \leftarrow 0$
4 $V' \leftarrow V; E' \leftarrow E$
5 While $F$ is not an fvs for $(V,E)$
  6 $l \leftarrow l+1$
  7 Recursively remove degree one vertices and incident
    edges from $V'$ and $E'$
  8 $S \leftarrow$ Violation$(V',E')$
  9 Increase $y_S$ until $\exists v_l \in S : \sum_{T:v_l\in T}(d_T(v_l)-1)y_T = w_{v_l}$
  10 $F \leftarrow F \cup \{v_l\}$
  11 Remove $v_l$ from $V'$ and attached edges from $E'$.
12 For $j \leftarrow l$ downto 1
  13 if $F-\{v_j\}$ is an fvs then $F \leftarrow F-\{v_j\}$
14 $F' \leftarrow F$
15 Output $F'$ (and $y$)
```

A. Primal–dual version of the Bafna, Fujito, Berman algorithm:

```
Violation$(V',E')$

1 If $(V',E')$ contains a semi-disjoint cycle $C$
  2 Return the vertex set of $C$
3 Else
  4 Return $V'$
```

B. New version:

```
Violation$(V',E')$

1 Return the vertices of an endblock of $(V',E')$
```

C. Primal–dual version of the Becker–Geiger algorithm:

```
Violation$(V',E')$

1 Return $V'$
  Replace line 9 with

9 Increase $y_S$ until $\exists v_l \in S : \sum_{T:v_l\in T} d_T(v_l)y_T = w_{v_l}$
```

Fig. 1. Primal–dual algorithm.

algorithm takes O(mn) time, where m is the number of edges in the graph and n is the number of vertices.

We now prove that the algorithm is a 2-approximation algorithm. Notice that for any feasible solution y for the dual program (D), $\sum_S b(S)y_S$ is a lower bound on the value of the optimal integer solution.

Theorem 3.1. *The algorithm of* Fig. 1, *version* A, *constructs an fvs F' and a solution y feasible for* (D) *such that*

$$\sum_{v\in F'} w_v \leqslant 2\sum_S b(S)y_S - \sum_S y_S.$$

Hence the algorithm is a 2*-approximation algorithm.*

Proof. We reduce the proof of the theorem to inequality (4). By construction of the algorithm,

$$\sum_{v\in F'} w_v = \sum_{v\in F'}\sum_{S:v\in S}(d_S(v)-1)y_S$$
$$= \sum_S \left[\sum_{v\in S\cap F'}(d_S(v)-1)\right] y_S.$$

Thus, if we can show that for any $y_S > 0$

$$\sum_{v\in(S\cap F')}(d_S(v)-1) \leqslant 2b(S)-1,$$

then the theorem statement will follow. In order to apply inequality (4), it is sufficient to argue that $S\cap F'$ is a minimal fvs for the graph $G[S]$, and that $G[S]$ has the requisite properties; although the inequality allows $S\cap F'$ to be almost minimal, we will not need this fact for this version of the algorithm. By construction of the algorithm F' is a minimal fvs. Also by construction, at the point in time in which the algorithm chooses the violated set S, none of the vertices in $F'\cap S$ is currently in F; they are added at some later point in the algorithm. Therefore, because the final step of the algorithm deletes redundant vertices in the reverse of the order in which they were added, $F'-F$, for the current value of F, must be a minimal fvs for the current graph (V',E'). Thus, whether S is a semidisjoint cycle in (V',E') or $S=V'$, we have that $F'\cap S$ is a minimal fvs for $G[S]$ and inequality (4) applies. □

Let Z^*_{IP} be the optimal value of (IP) and let Z^*_D be the optimal value of (D). We obtain the following corollary, which proves that the integrality gap is no more than 2.

Corollary 3.2. $Z^*_{\mathrm{IP}}/Z^*_D \leqslant 2$.

An anonymous referee has observed that this gap is essentially tight: on the complete graph K_n with $w_v = 1$ for all $v\in V$, the solution $x_v = \frac{1}{2}$ is a feasible solution to the linear programming relaxation, but the value of a minimum-weight fvs is $n-2$.

4. A new algorithm

We can get a somewhat simpler algorithm than the one above by using a different Violation subroutine. The subroutine for the Bafna et al. algorithm looks for a semidisjoint cycle in (V',E'). If it finds one, it sets S to be the vertices of the cycle, otherwise $S=V'$. Our Violation subroutine avoids searching for these semidisjoint cycles by a different choice of S. Here, given a decomposition of (V',E') into its 2-vertex-connected components, we set S to be an endblock, so that S contains at most one cutvertex. We give the algorithm in Fig. 1, version B, and prove the following.

Theorem 4.1. *The algorithm of* Fig. 1, *version* B, *constructs an fvs F' and a solution y feasible for* (D) *such that*

$$\sum_{v\in F'} w_v \leqslant 2\sum_S b(S)y_S - \sum_S y_S.$$

Hence the algorithm is a 2*-approximation algorithm.*

Proof. As in Theorem 3.1, we can reduce the proof to showing that

$$\sum_{v\in(S\cap F')}(d_S(v)-1) \leqslant 2b(S)-1.$$

We wish to apply inequality (4). By the choice of S, $G[S]$ is 2-vertex-connected and thus inequality (4) applies, so all we need to show is that $S\cap F'$ is an almost minimal fvs for $G[S]$. As in the proof of Theorem 3.1, given the value of F at the current point in the algorithm, $F' - F$ is a minimal fvs for (V',E'). The set S is chosen to be an endblock of (V',E'). Hence, the witness cycles for vertices in $F' \cap S$ must lie entirely within $G[S]$ except possibly the witness cycle for a cutvertex x. Thus each vertex in $F' \cap S$ except possibly x is necessary for $F' \cap S$ to be an fvs for $G[S]$; this implies that $F' \cap S$ is almost minimal for $G[S]$. □

5. The Becker–Geiger algorithm

Finally, we show that the primal–dual approach can be extended to the Becker–Geiger algorithm as well. Denote by $\tau(S)$ the *cardinality* (not the weight) of the smallest feedback vertex set in $G[S]$. By inequality (2) we have the following.

Corollary 5.1. *Let F be any feedback vertex set. Then for any $S \subseteq V$ such that $E[S] \neq \emptyset$,*

$$\sum_{v\in F\cap S} d_S(v) \geqslant |E[S]| - |S| + 1 + \tau(S) = b(S) + \tau(S).$$

By reasoning similarly to that in Section 2, the following integer program is also a formulation of the feedback vertex set problem:

$$\begin{aligned}(IP')\ \text{Min.} &\sum_{v\in V} w_v x_v \\ \text{s.t.} & \\ &\sum_{v\in S} d_S(v)x_v \geqslant b(S) + \tau(S), \quad S \subseteq V\colon E[S] \neq \emptyset, \\ &x_v \in \{0,1\}, \qquad v \in V.\end{aligned}$$

The dual of its linear programming relaxation is:

$$\begin{aligned}(D')\ \text{Max.} &\sum_{S} (b(S) + \tau(S))y_S \\ \text{s.t.} &\sum_{S:v\in S} d_S(v) y_S \leqslant w_v, \qquad v \in V, \\ &y_S \geqslant 0, \qquad S \subseteq V\colon E[S] \neq \emptyset.\end{aligned}$$

We would like to point out that the objective function of the dual is not well characterized, but this will not cause any problem for the algorithm. We can run a primal–dual algorithm with this relaxation just as before; the only differences here are that the VIOLATION subroutine returns $S = V'$, and that we use a slightly different dual increase step (line 9), corresponding to the slightly different dual constraints on (D'). See Fig. 1, version C, for a description of the algorithm. The algorithm is effectively the following: start with graph G and $F = \emptyset$. Recursively remove any degree one vertices and associated edges from the graph. Pick the vertex v that achieves the minimum $\varepsilon = \min_{v\in S} w_v/d(v)$. Add v to F, and set $w_u \leftarrow w_u - \varepsilon d(u)$ for all $u \in V$. Remove v from the graph, and repeat until F is a fvs. Perform the "reverse delete" step to obtain F'. This algorithm is essentially the Becker–Geiger algorithm. A straightforward implementation of this algorithm can be performed in $O(mn)$ time.

Theorem 5.2. *The algorithm of* Fig. 1, *version* C, *constructs an fvs F' and a solution y feasible for* (D) *such that*

$$\sum_{v\in F'} w_v \leqslant 2\sum_S (b(S) + \tau(S))y_S - 2\sum_S y_S.$$

Hence the algorithm is a 2*-approximation algorithm.*

Proof. As argued in the proof of Theorem 3.1, we only need to show that

$$\sum_{v\in(S\cap F')} d_S(v) \leqslant 2(b(S) + \tau(S)) - 2.$$

This follows from inequality (3) since $S \cap F'$ is a minimal fvs for $G[S]$ (as in the proof of Theorem 3.1) and since all vertices in $G[S]$ have degree at least two by construction. □

One important difference between this algorithm and the previous two is that here we cannot evaluate the value of the dual solution constructed. Nevertheless, a corollary similar to Corollary 3.2 still holds.

Acknowledgements

We thank David Shmoys for several useful conversations.

References

[1] V. Bafna, P. Berman, T. Fujito, Constant ratio approximation of the weighted feedback vertex set problem for undirected graphs, in: J. Staples, P. Eades, N. Katoh, A. Moffat (Eds.), ISAAC '95 Algorithms and Computation, number 1004 in Lecture Notes in Computer Science, Springer, Berlin, 1995, pp. 142–151.

[2] R. Bar-Yehuda, D. Geiger, J. Naor, R.M. Roth, Approximation algorithms for the vertex feedback set problem with applications to constraint satisfaction and Bayesian inference, in: Proc. 5th Ann. ACM-SIAM Symp. on Discrete Algorithms, 1994, pp. 344–354.

[3] A. Becker, D. Geiger, Approximation algorithms for the loop cutset problem, in: Proc. 10th Conf. on Uncertainty in Artificial Intelligence, 1994, pp. 60–68.
Update: A. Becker, D. Geiger: Optimization of Pearl's method of conditioning and greedy-like approximation algorithms for the vertex feedback set problem, Artificial Intelligence 83 (1996), 167–188.

[4] G. Even, J. Naor, B. Schieber, L. Zosin, Approximating minimum subset feedback sets in undirected graphs with applications, in: Proc. 4th Israel Symp. on Theory of Computing and Systems, 1996, pp. 78–88.

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

[6] M.X. Goemans, D.P. Williamson, Primal–dual approximation algorithms for feedback problems in planar graphs, in: W.H. Cunningham, S.T. McCormick, M. Queyranne (Eds.), Integer Programming and Combinatorial Optimization, number 1084, in Lecture Notes in Computer Science, Springer, Berlin, 1996, pp. 147–161, to appear in Combinatorica.

[7] M.X. Goemans, D.P. Williamson, The primal–dual method for approximation algorithms and its application to network design problems, in: D.S. Hochbaum (Ed.), Approximation Algorithms for NP-hard Problems, Ch. 4, PWS, Boston, 1996.

[8] D.S. Hochbaum, Approximating covering and packing problems: Set cover, vertex cover, independent set, and related problems, in: D.S. Hochbaum (Ed.), Approximation Algorithms for NP-hard Problems, Ch. 3, PWS, Boston, 1996.

[9] P. Klein, R. Ravi, When cycles collapse: a general approximation technique for constrained two-connectivity problems, in: Proc. 3rd MPS Conf. on Integer Programming and Combinatorial Optimization, 1993, pp. 39–55, also appears as Brown University Technical Report CS-92-30, to appear in Algorithmica.

[10] D.P. Williamson, M.X. Goemans, M. Mihail, V.V. Vazirani, A primal–dual approximation algorithm for generalized Steiner network problems, Combinatorica 15 (1995) 435–454.