

A 2-APPROXIMATION ALGORITHM FOR THE UNDIRECTED FEEDBACK VERTEX SET PROBLEM*

VINEET BAFNA[†], PIOTR BERMAN[‡], AND TOSHIHIRO FUJITO[§]

Abstract. A feedback vertex set of a graph is a subset of vertices that contains at least one vertex from every cycle in the graph. The problem considered is that of finding a minimum feedback vertex set given a weighted and undirected graph. We present a simple and efficient approximation algorithm with performance ratio of at most 2, improving previous best bounds for either weighted or unweighted cases of the problem. Any further improvement on this bound, matching the best constant factor known for the vertex cover problem, is deemed challenging.

The approximation principle, underlying the algorithm, is based on a generalized form of the classical local ratio theorem, originally developed for approximation of the vertex cover problem, and a more flexible style of its application.

Key words. approximation algorithm, performance guarantee, feedback vertex set problem, local ratio theorem

AMS subject classifications. 68Q25, 90C27, 05C85, 05C38

PII. S0895480196305124

1. Introduction. We are concerned with polynomial time approximation of the feedback vertex set problem on weighted, undirected graphs. A *feedback vertex set* (FVS) of a graph G is a set of vertices such that every cycle in G contains at least one vertex of the set. In a general setting some rational value is associated with each vertex of G as its *weight*. The FVS problem is then defined to be that of finding an FVS of minimum total weight in a given graph. In this paper we always assume the case of *weighted* (i.e., arbitrary weights on vertices) and *undirected* graphs unless otherwise specified.

This problem is of fundamental importance in combinatorial optimization. One typical application appears, as suggested by the name, in the context of combinatorial circuit design. The circuits are represented by graphs in which cycles potentially imply a “racing condition”; that is, some circuit element might receive new inputs before it stabilizes. One way to avoid such a condition is by placing a clocked register at each cycle in the circuit; in that case, we would like to keep the number of clocked registers as low as possible. The minimum FVS for the graph gives a bound on the number of registers needed. For other applications, e.g., in the areas of constraint satisfaction problems and Bayesian inference, see Bar-Yehuda et al. [4].

1.1. Short history and related work. The FVS problem is NP-hard; for directed graphs Karp showed its NP-completeness even if graphs are unweighted [12], and essentially the same transformation shows that it is equally hard for undirected

*Received by the editors June 12, 1996; accepted for publication (in revised form) November 23, 1998; published electronically September 7, 1999. A preliminary version of this paper was presented at the Sixth Annual International Symposium on Algorithms and Computation, Cairns, Australia, 1995.

<http://www.siam.org/journals/sidma/12-3/30512.html>

[†]DIMACS Center, Piscataway, NJ 08854 (bafna@dimacs.rutgers.edu). The work of this author was supported by Special Year National Science Foundation grant BIR-9412594.

[‡]Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA 16802 (berman@cse.psu.edu).

[§] Department of Electrical Engineering, Hiroshima University, 1-4-1 Kagamiyama, Higashi-Hiroshima 739-8527 Japan (fujito@huis.hiroshima-u.ac.jp).

graphs. Given this intractability it is natural to consider a next best approach: a polynomial time algorithm for computing a near optimal FVS. The quality of an approximation algorithm is measured by its *performance ratio*: the worst case ratio of weight of an approximate solution computed by the algorithm to the optimal solution weight. An algorithm with performance ratio r is also called an r -*approximation* algorithm.

The first nontrivial (i.e., better than $|V| = n$) approximation ratio of $2 \log n$ for unweighted graphs appeared in the early work of Erdős and Pósa [6], where they studied the number of (vertex) disjoint cycles in a graph. It was later improved to $\sqrt{\log n}$ by Monien and Schulz [14], who considered and compared various approaches to the problem. Only recently, Bar-Yehuda et al. were able to show that the smallest cardinality FVS (i.e., *unweighted* version) can be approximated within a constant factor of 4 [4]. Moreover, they considered the weighted version as well and obtained a performance ratio of $\min\{4 \log n, 2\Delta^2\}$, where Δ is the maximum vertex degree of a graph.

As for a lower bound on the performance ratio, the problem is known to be *MAX SNP*-hard [13, 16], implying that the ratio cannot go down arbitrarily close to 1 unless $P = NP$ [1]. In fact, a more direct implication is available due to the fact that the vertex cover (VC) problem is reducible to the FVS problem in an approximation preserving manner [13], so that any performance ratio r for the FVS problem would imply the same ratio r for the VC problem. A better approximation of the VC problem has been a subject of extensive research over the years, yet the best constant approximation ratio has remained at 2. (The overall best one is $2 - \log \log n / 2 \log n$ [3, 15].) On the other hand, a lower bound on the performance ratio for the VC problem has been continuously improved in the last few years, and currently it is known to be NP-hard to guarantee a factor of $7/6 - \epsilon$ for any $\epsilon > 0$ [11], implying the same bound for the FVS problem.

The FVS problem (or feedback *edge* set problem) for *directed* graphs, largely due to more versatile nature of its applicability, has drawn even more attention in various areas. It appears, however, that the problem is harder to approximate, with $O(\log n \log \log n)$ being the best ratio known today [7].

1.2. Our contributions.

Factor-2 approximation. We will present an approximation algorithm for the weighted FVS problem (in section 3) and show that its performance ratio is bounded above by 2 (in section 4), improving upon the previous best of $\min\{4 \log n, 2\Delta^2\}$. Independently of our work, Becker and Geiger have recently discovered a different 2-approximation algorithm [5], their analysis of which is more complicated than ours, without any elucidation of underlying approximation principles. In light of the facts mentioned earlier concerning the approximability of the VC problem and its reducibility to the FVS problem, achieving a better performance ratio, if at all possible, is deemed quite challenging. Our algorithm is also quite simple and efficient; it can be implemented to run in time $O(\min\{|E| \log |V|, |V|^2\})$.

Generalized local ratio approximation. Our approximation method is based on the *local approximation* principle. In a most simple form it was used already in the Gavril's maximal matching-based approximation for the unweighted VC problem [9, p. 134], and later it was explicitly formalized as the *local ratio theorem* by Bar-Yehuda and Even [3].

This principle, however, has been known mostly only in a doubly limited form;

the formulation allowed extraction of only *uniformly* weighted subgraphs from an arbitrarily weighted graph and, as its name suggests, allowed its application only to subgraphs of *small* size, e.g., short odd cycles for VC approximation [3] and short cycles for weighted FVS approximation [4]. An easy but crucial observation presented in this paper is that neither restriction is necessary, leading to a generalization of the theorem (section 2) and demonstration of the effectiveness resulting from more flexible applications of it. Moreover, the principle is applicable not only to the VC or FVS problems but also to other weighted optimization problems of *covering* type, and hence this approximation technique could be of independent interest. For a simpler presentation, however, we restrict ourselves to the FVS problem in this paper.

1.3. Definitions and notation. We use the following definitions and notation throughout the paper. For any graph G let $V(G)$ denote the vertex set of G and $G[U]$ the subgraph of G induced by U for $U \subseteq V(G)$. A (vertex) weighted graph with a weight function $w : V(G) \rightarrow \mathbf{Q}_+$ is denoted as (G, w) , where a nonnegative rational $w(u)$ represents a weight associated with each vertex u of G . The sum of weights of vertices in U is denoted by $w(U) (= \sum_{u \in U} w(u))$. A collection $\{(G_i, w_i)\}$ of weighted graphs is called a *decomposition* of (G, w) if G_i is a subgraph of $G \forall i$ and $\sum_{i: u \in V(G_i)} w_i(u) \leq w(u) \forall u \in V$. A weight function $w : V \rightarrow \mathbf{Q}_+$ is called *degree-proportional* if, for some constant $c > 0$, $w(u) = c(d(u) - 1)$ for every $u \in V$, where $d(u)$ denotes degree of u . A graph is called *clean* if it contains no vertex of degree less than 2, and a cycle C is *semidisjoint* if, for every vertex u of C , $d(u) = 2$ with at most one exception. Let $\text{opt}(G, w)$ denote any optimal FVS in (G, w) . An FVS F is *minimal* in G if no smaller FVS is contained in F or, equivalently, if $F - \{u\}$ is not an FVS in G for all $u \in F$.

2. Generalization of the local ratio theorem. The local ratio approximation, in its most elementary use, is based on the following principle: If an (*unweighted*) structure G contains a substructure H such that every optimal solution occupies a large portion of H , one can afford to take the whole of H into a solution and reduce the problem by removing H from G . More specifically and when G is a weighted graph, this idea can be implemented by the following operations:

1. Choose a suitable subgraph H (i.e., local structure) of usually small size.
2. "Subtract" H with a *uniform* weight distribution on it from G (i.e., entire structure).
3. Accept into a solution all the vertices of weight *reduced to zero*.

We extend this technique in two directions, one in its formulation and the other in its application. First it will be allowed to include *nonuniformly* weighted subgraphs in our target local structures. This change enables us to choose not only a subgraph but also a weight distribution on it to be sliced off from the whole distribution. Formally, we have the following theorem.

THEOREM 2.1. *Let $\{(G_i, w_i)\}$ be a decomposition of (G, w) and F be any FVS in G such that $w(F) = \sum_i w_i(F \cap V(G_i))$. Then,*

$$\frac{w(F)}{w(\text{opt}(G, w))} \leq \max_i \left\{ \frac{w_i(F \cap V(G_i))}{w_i(\text{opt}(G_i, w_i))} \right\}$$

Proof. Since $\{(G_i, w_i)\}$ is a decomposition of (G, w) , $w(X) \geq \sum_i w_i(X \cap V(G_i))$ for any set $X \subseteq V$. Thus, using the assumption on $w(F)$,

$$\frac{w(F)}{w(\text{opt}(G, w))} \leq \frac{\sum_i w_i(F \cap V(G_i))}{\sum_i w_i(\text{opt}(G, w) \cap V(G_i))}.$$

Besides, if F is an FVS in G , so is its restriction $F \cap V(G_i)$ in every subgraph G_i . Therefore, we have $w_i(\text{opt}(G, w) \cap V(G_i)) \geq w_i(\text{opt}(G_i, w_i))$ for all i , which gives

$$\frac{\sum_i w_i(F \cap V(G_i))}{\sum_i w_i(\text{opt}(G, w) \cap V(G_i))} \leq \frac{\sum_i w_i(F \cap V(G_i))}{\sum_i w_i(\text{opt}(G_i, w_i))}.$$

Since every summand appearing in the fraction on the right-hand side is nonnegative, it can be bounded above by $\max_i \{w_i(F \cap V(G_i))/w_i(\text{opt}(G_i, w_i))\}$. \square

In this way the ratio of an FVS weight to the optimal one in (G, w) can be reduced to the ones in its subgraphs (G_i, w_i) .

For a subgraph G_1 of G let $\gamma \stackrel{\text{def}}{=} \min\{w(v) : v \in V(G_1)\}$, w_1 , be a function on $V(G_1)$ whose value is constantly γ , $w_2 \stackrel{\text{def}}{=} w - w_1$, and let V_0 be a set of vertices u with $w_2(u) = 0$. Then $\{(G_1, w_1), (G[V - V_0], w_2)\}$ is easily a decomposition of (G, w) and, for any FVS F_2 for $G[V - V_0]$, $F \stackrel{\text{def}}{=} V_0 \cup F_2$ is an FVS for G . Let c^* denote the *cardinality* of an optimal FVS for unweighted G_1 . The original local ratio theorem of Bar-Yehuda and Even [3] states that the approximation ratio of F is bounded by

$$\max \left\{ \frac{|V(G_1)|}{c^*}, \frac{w_2(F_2)}{w_2(\text{opt}(G[V - V_0], w_2))} \right\},$$

and this follows easily from Theorem 2.1 since $w_1(F \cap V(G_1)) \leq w_1(V(G_1)) = \gamma|V(G_1)|$; thus $w_1(F \cap V(G_1))/w_1(\text{opt}(G_1, w_1)) \leq \gamma|V(G_1)|/\gamma c^* = |V(G_1)|/c^*$ and $F \cap V(G_2) = F \cap (V - V_0) = F_2$.

The second extension of the local ratio technique will be demonstrated in the next section, where our algorithm slices up a weight distribution from the entire structure.

3. Approximation algorithm. Our algorithm, called FEEDBACK, is presented in Figure 3.1, where text in square brackets are comments used for analysis only.

Given a graph (G, w) with $G = (V, E)$, any vertex of weight zero is removed from G and placed in the solution set F at the outset. FEEDBACK then decomposes (G, w) into subgraphs (G_i, w_i) 's (in the first While loop) by iteratively subtracting w_i from w , removing vertices of weight reduced to zero, adding them into F , and cleaning up G (by procedure CLEANUP, which recursively deletes vertices of degree ≤ 1), until G becomes empty.

The subgraph G_i derived in the i th iteration is either a semidisjoint cycle C contained in G or, otherwise, G itself. Note that the first case has precedence over the second; that is, G_i is a semidisjoint cycle whenever G contains one. When G_i is a cycle C it is uniformly weighted with $w_i(u) = \gamma = \min\{w(u) : u \in V(C)\}$, the minimum weight on C , for all $u \in G_i$. Otherwise, G_i is clean and degree-proportionally weighted with $w_i(u) = \gamma(d(u) - 1) \forall u \in V$. In either case the value of γ is determined such that w_i is maximal without exceeding w , and hence some vertex u of G necessarily has its weight $w(u)$ reduced to zero when w_i is subtracted from w . Such vertices are removed from G , making progress toward emptying G , and at the same time we collect them all in F . The sole purpose of using an auxiliary stack data structure, STACK, is to keep track of the (reverse) order in which these vertices are added into F .

The graph G eventually becomes empty (in at most $|V|$ iterations). At this point (i.e., right after the first While) every vertex was swept out in the process, or otherwise it is kept in F . Observe that F is indeed an FVS for the original G because any vertex was cleaned up only after it was found to be useless.

The second While loop examines vertices of this F , one by one, in the reverse order of their inclusion into F . Whenever a vertex is found to be extraneous, it is

Input: an undirected graph $G = (V, E)$ with vertex weights $w : V \rightarrow \mathbf{Q}_+$

Output: a feedback vertex set F

```

Initialize  $F = \{u \in V : w(u) = 0\}, V = V - F. [i = 0]$ 
Cleanup( $G$ )
While  $V \neq \emptyset$  do
   $[i \leftarrow i + 1]$ 
  If  $G$  contains a semidisjoint cycle  $C$ , then
    Let  $\gamma \leftarrow \min\{w(u) : u \in V(C)\}$ .
    Set  $w(u) \leftarrow w(u) - \gamma, \forall u \in V(C)$ .
     $[G_i = C$  and  $w_i(u) = \gamma, \forall u \in V(C)]$ 
  Else [ $G$  is clean and contains no semidisjoint cycle]
    Let  $\gamma \leftarrow \min\{w(u)/(d(u) - 1) : u \in V\}$ .
    Set  $w(u) \leftarrow w(u) - \gamma(d(u) - 1), \forall u \in V$ .
     $[G_i = G$  and  $w_i(u) = \gamma(d(u) - 1), \forall u \in V]$ 
  For each  $u \in V$  with  $w(u) = 0$  do
    Remove  $u$  from  $V$ , add it to  $F$ , and push it onto STACK.
  Cleanup( $G$ )
While STACK  $\neq \emptyset$  do
  Let  $u \leftarrow \text{pop}(\text{STACK})$ .
  If  $F - \{u\}$  is an FVS in original  $G$ , then [ $u$  is redundant]
    Remove  $u$  from  $F$ .

Cleanup( $G$ ):
  While  $G$  contains a vertex of degree at most 1, remove it along with
  any incident edges.

```

FIG. 3.1. 2-approximation algorithm FEEDBACK for the FVS problem.

discarded from F . As will be seen later, this process ensures not only that F is a minimal FVS in original G but also that $F \cap V(G_i)$ in $G_i \forall i$.

Running time. The running time of FEEDBACK is dominated by the first While loop. All the operations of cleaning up vertices (along with edges), detecting semidisjoint cycles, computing the minimum weights on them, and deleting them from a graph can be done in time $O(|V| + |E|)$ by maintaining a collection of existent (disjoint) paths consisting solely of degree-2 vertices. Since each iteration takes $O(|V|)$ time for other operations and there are at most $|V|$ of them, the running time is $O(|V|^2)$.

Alternatively, we may maintain the value of $w(u)/(d(u) - 1)$ for each $u \in V$ in a priority queue P , instead of *individual* vertex weights. The computation of $\gamma = \min\{w(u)/(d(u) - 1) : u \in V\}$ is then supported by the **Extract-Min** operation ($O(\log |V|)$). There are two types of updates for these values: one by subtraction of w_i from w and the other caused by decrement of degrees. In the former case, the new value, after subtraction of w_i , becomes $(w(u) - \gamma(d(u) - 1))/(d(u) - 1) = w(u)/(d(u) - 1) - \gamma$, the old value less γ for each vertex. Thus, the actual value for any vertex in V can be recovered from a sequence of γ values, without changing key values stored in P . For the second case, however, some key values must be changed, but only for those vertices adjacent to u , for each removal of vertex u . We do so using both **Insert** and **Delete** operations, each of which takes $O(\log |V|)$ by the standard implementations, and $O(1)$ time calculation of a new value. Since key

values of elements in P need to be modified at most $|E|$ times, it takes $O(|E| \log |V|)$ overall, better than $O(|V|^2)$ when a graph is sparse.

4. Performance ratio. To avoid any possible ambiguity in the following argument, let us fix an input graph $(G = (V, E), \bar{w})$ and the output FVS \bar{F} . Recall that our goal is to achieve a globally good approximation ratio by ensuring a good ratio locally at every derived subgraph in a decomposition of the given graph, and our algorithm FEEDBACK is designed exactly to do so. Toward this end it will be shown below, in this order, that the following hold:

1. FEEDBACK computes \bar{F} and a decomposition $\{(G_i, w_i)\}$ of (G, \bar{w}) such that

(i) $\bar{w}(\bar{F}) = \sum_i w_i(\bar{F} \cap V(G_i))$ (Lemma 4.1, precondition for application of Theorem 2.1), and

(ii) $\bar{F} \cap V(G_i)$ is a minimal FVS in G_i for all i (Lemma 4.2).

2. The weight of any minimal FVS in any clean, degree-proportionally weighted graph without semidisjoint cycles is small relative to the optimal weight (Lemmas 4.3 and 4.4).

LEMMA 4.1. $\bar{w}(\bar{F}) = \sum_i w_i(\bar{F} \cap V(G_i))$.

Proof. Recall that any vertex u of G can enter \bar{F} only after its weight $w(u)$ is reduced completely to zero by a sequence of subtractions, and hence partial weights $w_i(u)$'s must sum up to the total $\bar{w}(u)$ for any u in \bar{F} . (On the other hand, $V - \bar{F}$ consists of those vertices cleaned up from G prematurely with nonzero weights). \square

LEMMA 4.2. $\bar{F} \cap V(G_i)$ is a minimal FVS in $G_i \forall i$.

Proof. Let G^i denote the graph remaining right after the $(i - 1)$ st iteration of the first While is completed. We first claim that $\bar{F} \cap V(G^i)$ is a minimal FVS in G^i . Let F^∞ denote the FVS constructed by the entire run of the first While. Since vertices in F^∞ are examined, in the second While, in the reverse order of their addition to F^∞ , all the vertices in $F^\infty \cap V(G^i)$ are tested for their redundancy before those in $F^\infty - V(G^i)$. Let F^i denote the FVS obtained from F^∞ , during the second While, by removing any vertices in $F^\infty \cap V(G^i)$ which are redundant in G . Then, $F^i \cap V(G^i)$ must be a minimal FVS in G^i since otherwise, i.e., if $(F^i \cap V(G^i)) - \{u\}$ is an FVS in G^i for some $u \in F^i \cap V(G^i)$, $F^i - \{u\}$ would be an FVS in G , a contradiction. Since $F^i \cap V(G^i) = \bar{F} \cap V(G^i)$, the claim follows.

It remains to observe that if $\bar{F} \cap V(G^i)$ is a minimal FVS in G^i , so is $\bar{F} \cap V(G_i)$ in G_i . Recall that G_i is chosen such that either $G_i = G^i$ or $G_i = C$, some semidisjoint cycle contained in G^i . It is obvious when $G_i = G^i$, and now suppose $G_i = C$. Then, $\bar{F} \cap G_i$ contains exactly one vertex of G_i , and hence minimal in G_i , because any minimal FVS in G^i , such as $\bar{F} \cap V(G^i)$, can contain only one vertex from semidisjoint C . \square

We next consider how large the weight of \bar{F} is when estimated in subgraphs (G_i, w_i) 's, relative to the optimal weights for them. Recall that (G_i, w_i) is in the form of either

1. a simple cycle of identically weighted vertices, or
2. a clean and degree-proportionally weighted graph containing no semidisjoint cycles.

In the first case, the minimality of FVS $\bar{F} \cap V(G_i)$ in G_i actually implies its optimality in G_i . The second case is more interesting. We show that, in this case, the weight of any minimal FVS is bounded above by twice the optimum weight.

Clearly, it suffices to prove this only for the case when $w(u) = d(u) - 1$ for every $u \in V$; this is assumed below in Lemmas 4.3 and 4.4. We will also use a potential function $p : V \rightarrow \mathcal{Q}$ defined as $p(u) = d(u)/2 - 1$ for every $u \in V$. Let $p(U)$ denote

$\sum_{u \in U} p(u)$ for any $U \subseteq V$.

LEMMA 4.3. For an arbitrary FVS F in $G = (V, E)$, $w(F) \geq p(V) + 1$.

Proof. Let $\delta(F)$ denote the set of edges incident to some vertex in F . Then, since $\sum_{u \in F} d(u) \geq |\delta(F)|$, $w(F) = \sum_{u \in F} (d(u) - 1) = \sum_{u \in F} d(u) - |F| \geq |\delta(F)| - |F|$. Besides, F is an FVS. Thus $G[V - F]$, with its edge set being $E - \delta(F)$, is acyclic, containing at most $|V - F| - 1$ edges. That is, $|E - \delta(F)| \leq |V - F| - 1$ and hence $w(F) \geq |E| - |V| + 1 = p(V) + 1$ since $p(V) = \sum_{u \in V} (d(u)/2 - 1) = |E| - |V|$. \square

Observe now that if F is a minimal FVS, each vertex of F is *blocked* by a tree in the forest $G[V - F]$; i.e., each vertex $u \in F$ is joined via two edges to some tree T in $G[V - F]$. Let e_T be the number of edges with one end in T and the other in F . Suppose that T has t vertices. Then $\sum_{u \in V(T)} d(u) = e_T + 2(t - 1)$. Consequently,

$$(4.1) \quad \begin{aligned} p(V(T)) &= \sum_{u \in V(T)} \left(\frac{d(u)}{2} - 1 \right) \\ &= \frac{\sum_{u \in V(T)} d(u)}{2} - t = \frac{e_T + 2(t - 1)}{2} - t = \frac{e_T}{2} - 1. \end{aligned}$$

LEMMA 4.4. If F is a minimal FVS in a clean graph $G = (V, E)$ without a semidisjoint cycle, then $w(F) \leq 2p(V)$.

Proof. Each vertex $u \in F$ is a priori allocated a potential of $(d(u) - 2)/2$. We show that each $u \in F$ also receives an additional potential of $1/2$ from vertices in $V - F$. Let T be a tree blocking u (so $e_T \geq 2$). Notice that $e_T = 2$ would imply either G is not clean or G contains a semidisjoint cycle; hence $e_T \geq 3$. Also, the total number of vertices each tree T' in $G[V - F]$ can block is at most $\lfloor e_{T'}/2 \rfloor$. Thus, using (4.1), an extra potential of $(e_T/2 - 1)/\lfloor e_T/2 \rfloor$ can be shipped to every vertex u of F from $V - F$, which is at least $1/2$ when $e_T \geq 3$. Therefore, u contributes $d(u) - 1$ to $w(F)$ and at least $(d(u) - 1)/2$ to $p(V)$. \square

Lemmas 4.3 and 4.4 jointly assert that when G_i is a clean and degree-proportionally weighted graph without semidisjoint cycles, since $\bar{F} \cap V(G_i)$ is a minimal FVS in G_i (Lemma 4.2), the ratio $w_i(\bar{F} \cap V(G_i))/w_i(\text{opt}(G_i, w_i))$ is bounded by $2p(V(G_i))/(p(V(G_i)) + 1)$.

THEOREM 4.5. The algorithm FEEDBACK finds an FVS \bar{F} in (G, \bar{w}) , where $G = (V, E)$, with approximation factor of $2 - 2/(|E| - 3)$ in time $O(\min(|E| \log |V|, |V|^2))$.

Proof. Apply Theorem 2.1 using $\{(G_i, w_i)\}$ computed (implicitly) by FEEDBACK as a decomposition of (G, \bar{w}) . As observed above, when G_i is a uniformly weighted simple cycle (case 1), the local ratio $w_i(\bar{F} \cap V(G_i))/w_i(\text{opt}(G_i, w_i)) = 1$. On the other hand, when G_i is a clean graph without semidisjoint cycles, it is bounded by

$$\frac{2p(V(G_i))}{p(V(G_i)) + 1} = 2 - \frac{2}{p(V(G_i)) + 1} = 2 - \frac{2}{|E(G_i)| - |V(G_i)| + 1} \leq 2 - \frac{2}{|E| - 3}$$

since G_i must contain at least four vertices. \square

Additionally, it can be seen that the analysis above is essentially tight: there is an infinite sequence of graphs for which the approximation factor of FEEDBACK approaches arbitrarily close to 2. Consider, e.g., a graph G consisting of k triangles $\{a_i, b_i, c_i\}$, $i = 1, \dots, k$, which are chained together by edges $\{b_i, a_{i+1}\}$, $i = 1, \dots, k - 1$, and $\{b_k, a_1\}$ (see Figure 4.1). Suppose now that G is degree-proportionally weighted, e.g., $w(a_i) = w(b_i) = 2$, and $w(c_i) = 1$, $\forall i$. The set $A = \{a_i : 1 \leq i \leq k\}$ is a minimal FVS with $w(A) = 2k$. On the other hand, an optimal FVS could be formed by c_i 's, $1 \leq i \leq k - 1$, plus a_1 , with its weight totaling $(k - 1) + 2 = k + 1$.

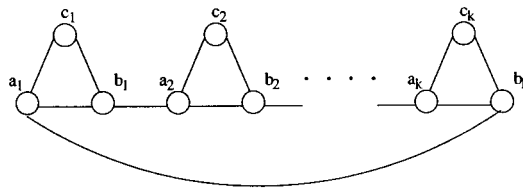


FIG. 4.1. Example with ratio = 2.

5. Final remarks. In this paper we have presented a simple and efficient approximation algorithm for the FVS problem on undirected graphs, with a performance ratio of at most 2. While this ratio matches the best constant approximation factor known for the VC problem, there still exists a small gap from the overall best of $2 - \log \log n / 2 \log n$ for VC. Also, some related directions for further research are suggested when it is taken into account that VC, when restricted to some special classes of graphs, becomes polynomially solvable or easier to approximate. For instance, one of the well-studied cases occurs when graphs are restricted to be of small vertex degree, for which currently the best-known bound is $2 - 3/(\Delta + 2)$ [10], where Δ is the maximal vertex degree. Although the technique introduced in the paper alone is not powerful enough to do any better for this special case, it can be shown that, when combined with other approximation preserving reductions, it yields the performance ratio of $2 - 2/(3\Delta - 2)$. (Interested readers are referred to [2].) It remains an open question whether one can approximate the FVS problem *exactly* as well as the VC problem can be, whether the degree is bounded or not.

Other interesting questions for further investigation include applicability of the techniques developed herein to other problems. For instance, see [8], where, inspired by our work, a similar approach was employed and shown to be effective in approximation of other *node-deletion* problems as well.

Acknowledgment. We gratefully acknowledge a number of valuable comments and suggestions given by the anonymous referees.

REFERENCES

- [1] S. ARORA, C. LUND, R. MOTWANI, M. SUDAN, AND M. SZEGEDY, *Proof verification and hardness of approximation problems*, in 33rd Annual Symposium on Foundations of Computer Science, Pittsburgh, PA, 1992, pp. 14–23.
- [2] V. BAFNA, P. BERMAN, AND T. FUJITO, *Constant Ratio Approximations of Feedback Vertex Sets in Weighted Undirected Graphs*, Tech. Report TR-96-29, DIMACS, Piscataway, NJ, 1996.
- [3] R. BAR-YEHUDA AND S. EVEN, *A local-ratio theorem for approximating the weighted vertex cover problem*, in Ann. Discrete Math. 25, North-Holland, Amsterdam, 1985, pp. 27–46.
- [4] R. BAR-YEHUDA, D. GEIGER, J. NAOR, AND R. M. ROTH, *Approximation algorithms for the feedback vertex set problem with applications to constraint satisfaction and Bayesian inference*, SIAM J. Comput., 27 (1998), pp. 942–959.
- [5] A. BECKER AND D. GEIGER, *Approximation algorithms for the loop cutset problem*, in Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence, San Francisco, CA, Morgan Kaufman, 1994, pp. 60–68.
- [6] P. ERDŐS AND L. PÓSA, *On the maximal number of disjoint circuits of a graph*, Publ. Math. Debrecen, 9 (1962), pp. 3–12.
- [7] G. EVEN, J. NAOR, B. SCHIEBER, AND M. SUDAN, *Approximating minimum feedback sets and multi-cuts in directed graphs*, in Fourth Integer Programming and Combinatorial Opti-

- mization, Lecture Notes in Comput. Sci. 920, Springer-Verlag, New York, 1995, pp. 14–28.
- [8] T. FUJITO, *A unified approximation algorithm for node-deletion problems*, Discrete Appl. Math., 86 (1998), pp. 213–231.
 - [9] M. GAREY AND D. JOHNSON, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, New York, 1979.
 - [10] M. HALLDÓRSSON AND H. LAU, *Low-degree graph partitioning via local search with applications to constraint satisfaction, max cut, and coloring*, J. Graph Algorithms Appl., 1 (1997), pp. 1–13.
 - [11] J. HÅSTAD, *Some optimal in-approximability results*, in Proceedings of the 29th Annual ACM Symposium on Theory of Computing, El Paso, TX, 1997, pp. 1–10.
 - [12] R. KARP, *Reducibility among combinatorial problems*, in Complexity of Computer Computations, R. Miller and J. Thatcher, eds., Plenum Press, New York, 1972, pp. 85–103.
 - [13] J. LEWIS AND M. YANNAKAKIS, *The node-deletion problem for hereditary properties is NP-complete*, J. Comput. System Sci., 20 (1980), pp. 219–230.
 - [14] B. MONIEN AND R. SCHULZ, *Four approximation algorithms for the feedback vertex set problem*, in Proceedings of the Seventh Conference on Graph Theoretic Concepts of Computer Science, Munich, Hanser Verlag, 1981, pp. 315–326.
 - [15] B. MONIEN AND E. SPECKENMEYER, *Ramsey numbers and an approximation algorithm for the vertex cover problem*, Acta Inform., 22 (1985), pp. 115–123.
 - [16] C. PAPADIMITRIOU AND M. YANNAKAKIS, *Optimization, approximation and complexity classes*, J. Comput. System Sci., 43 (1991), pp. 425–440.