15-854: Approximations Algorithms | **Lecturer:** R. Ravi
Topic: The Local Ratio Method | **Date:** 10-10-2005
Scribe: Viswanath Nagarajan

9.1 Introduction

In this lecture, we will study the local ratio technique in approximation algorithms. We will demonstrate this by presenting approximation algorithms for the problems of feedback vertex set and weighted vertex cover, both in undirected graphs.

Roughly speaking, the local ratio method works iteratively, and in each iteration it breaks a suitable structure (the local structure) to solve from the residual structure in that iteration. Typically the local structure is an instance that is very easy to solve. The main claim that is required is that the solution to the local structure can be patched with the solution from the next iteration, to get a good solution in the current iteration.

9.2 Minimum Feedback Vertex Set

A feedback vertex set (FVS) in an undirected graph $G = (V, E)$ is a set $F \subseteq V$ of vertices such that every cycle in the graph G contains some vertex in F. The *minimum feedback vertex set* problem is as follows: Given an undirected graph $G = (V, E)$, with a non-negative cost function on its vertices, $c : V \to \mathbb{R}^+$, find a minimum cost subset of vertices that is an FVS in G. The algorithm that we present here is due to Bafna, Berman & Fujito [1].

Lemma 9.2.1 *Minimum feedback vertex set is NP-complete.*

Proof: We reduce the Vertex Cover problem to minimum feedback vertex set. An instance of the vertex cover problem consists of an undirected graph $G = (V, E)$, and a number k. The decision problem is to determine if there exists a vertex cover of size at most k in G. Define a new graph H on the vertex set $U_v \cup U_e$, where vertices of $U_v = V$ correspond to the vertices of G, and vertices of $U_e = E$ correspond to edges of G. For every edge $e = (v_1, v_2) \in E$, there are three edges in H: an edge between vertices v_1 and v_2 in U_v, an edge between $v_1 \in U_v$ and $e \in U_e$, and an edge between $v_2 \in U_v$ and $e \in U_e$. It is easy to see that H has an FVS of size $\leq k$ iff G has a vertex cover of size $\leq k$. ■

This reduction is approximation preserving, so minimum feedback vertex set is at least as hard to approximate as vertex cover. We will look at is a 2-approximation algorithm for minimum feedback vertex set. Obtaining an approximation guarantee of a constant smaller than 2 for vertex cover (and hence minimum feedback vertex set) is considered a very hard problem.

We will first look at the algorithm seen in class and show that it is a 3-approximation algorithm. Then we will look at the algorithm of [1], and show how it improves upon the algorithm done in class to achieve an approximation guarantee of 2.

9.2.1 A 3-Approximation Algorithm

We need some definitions before we present the algorithm. An FVS F is called a *minimal* feedback vertex set if for every $v \in F$, $F \setminus \{v\}$ is not an FVS. A graph G is called *clean* if it has no vertex of degree less than 2. We will be concerned only with clean graphs; procedure **Clean** shows how any graph can be modified to a clean graph.

LR-FVS(G, c):
    If $(V(G) = \phi)$ return ϕ.
    $\alpha \leftarrow \min\{c_v/(d_v - 1) : v \in V(G)\}$.
    Set $\overline{c}_v \leftarrow \alpha \cdot (d_v - 1)$, for all $v \in V(G)$.
    Set $c'_v \leftarrow c_v - \overline{c}_v$, for all $v \in V(G)$.
    $\overline{F} \leftarrow \{v \in V(G) : c'_v = 0\}$.
    $G' \leftarrow G \setminus \overline{F}$.
    $F' \leftarrow$ **LR-FVS**(G, c).
    Set $F \subseteq F' \cup \overline{F}$ to be a minimal FVS in G.
    Return F.

Clean(G, c):
    While G contains any vertex of degree less than 2, remove it.

A cost function on a graph is called *degree-1 weighted* if the cost of each vertex is proportional to its degree-1. In algorithm **LR-FVS**, the cost function $\overline{c}$ in graph G is degree-1 weighted. Degree-1 weighted cost functions are useful because of the following claim.

Claim 9.2.2 *Let $G = (V, E)$ be a clean graph. If $c_v = \alpha \cdot (d_v - 1), \forall v \in V$, is a degree-1 weighted cost function on G, the cost of any minimal FVS in G is at most thrice the cost of the minimum FVS in G.*

Proof: Let G have $n = |V|$ vertices and $m = |E|$ edges. We assume, without loss of generality, that $\alpha = 1$. We also assume that the graph G is connected - it is easy to extend the proof to the case when G has many components. We prove the claim in two parts. First, we show that the cost of the minimum FVS in G is at least $m - n + 1$. Then we show that any minimal FVS in G has cost at most $3(m - n + 1)$.

Let F^* denote the minimum FVS in G, and $|F^*| = f^*$. Since $G \setminus F^*$ is acyclic, the number of edges in $G \setminus F^*$ is at most $n - f^* - 1$. So the number of edges incident on some vertex of F^* is $|E(F^*, V)| \geq m - (n - f^* - 1)$. Now,

$$c(F^*) = \sum_{v \in F^*} (d_v - 1) = \sum_{v \in F^*} d_v - f^* \geq |E(F^*, V)| - f^* \geq m - n + 1 \qquad (9.2.1)$$

For the next part of the proof, let F be any minimal FVS in G and $|F| = f$. Let $\mathcal{T} = \{T_1, \ldots, T_k\}$ denote the connected components (trees) in $G \setminus F$. From each component $T \in \mathcal{T}$, there are at least two edges into F. So the number of cross edges c, between F and $V \setminus F$ is at least $2k$. The cost of FVS F is $c(F) = \sum_{v \in F} d_v - f = 2m - \sum_{v \in V \setminus F} d_v - f$. But $\sum_{v \in V \setminus F} d_v$ is exactly two times the number of edges in $G \setminus F$ (i.e., $2(n - f - k)$) plus the number of edges from $V \setminus F$ to F (i.e., c). In other words, $\sum_{v \in V \setminus F} d_v = 2(n - f - k) + c$. So,

$$c(F) = 2m - (2n - 2f - 2k + c) - f = 2(m - n) + f + (2k - c) \leq 2(m - n) + f \tag{9.2.2}$$

We observe the following property of any minimal FVS in G.

Proposition 9.2.3 *Let G be a connected graph with n vertices and m edges. Then, the size (number of vertices) of any minimal FVS in G is at most $m - n + 1$.*

Proof: The graph $G \setminus F$ is a forest with the components $\mathcal{T}$. Since G is connected, we can add some edges of G to forest $G \setminus F$ to a obtain a tree S. Since F is a minimal FVS in G, for each vertex $v \in F$, there is some component $T \in \mathcal{T}$ such that v has at least 2 edges, $e_1(v)$ & $e_2(v)$, going into T. Note that for any $v \in F$, not both $e_1(v)$ and $e_2(v)$ can be in the tree S - as this would form a cycle! So there are at least $|F|$ edges of G that are not present in the tree S. Thus the number of edges in G, $m \geq |F| + n - 1$, or $|F| \leq m - n + 1$. ■

Using Proposition 9.2.3 in equation 9.2.2 completes the proof of the claim. ■

For an instance (G, c) of minimum feedback vertex set, we denote the optimal solution by $OPT(G, c)$. The cost of the optimal solution is then $c(OPT(G, c))$. The approximation guarantee of algorithm **LR-FVS** is proved by the following Theorem.

Theorem 9.2.4 *For any instance (G, c) of minimum feedback vertex set, the cost of the feedback vertex set F returned by algorithm* **LR-FVS** *is $c(F) \leq 3 \cdot c(OPT(G, c))$.*

Proof: The proof of this theorem is by induction on the number of vertices in the graph. The base case is when the graph is empty, and this case is trivial. In the inductive step, we first argue that the solution F is a feedback vertex set in G. By induction on (G', c'), F' is an FVS in G'. Since $G' = G \setminus \overline{F}$, the only cycles of G not covered by F' are those on $\overline{F}$. So $F' \cup \overline{F}$ is an FVS in G. So a minimal FVS $F \subseteq F' \cup \overline{F}$ does exist.

For bounding the cost of solution F, note that we can write $c(F) = c'(F) + \overline{c}(F)$. Now, $c'(F) \leq c'(F') + c'(\overline{F}) = c'(F')$. But $c'(F') \leq 3 \cdot c'(OPT(G', c'))$, by induction on G'. So, $c'(F) \leq 3 \cdot c'(OPT(G', c'))$.

F is a minimal FVS in graph G, and $\overline{c}$ is a degree-1 weighted cost function on G. Using Claim 9.2.2, we get $\overline{c}(F) \leq 3 \cdot \overline{c}(OPT(G, \overline{c}))$.

Now we can write $c(F) = c'(F) + \overline{c}(F) \leq 3 \cdot c'(OPT(G', c')) + 3 \cdot \overline{c}(OPT(G, \overline{c}))$. However, note that $OPT(G, c)$ is an FVS in both the instances (G', c') and $(G, \overline{c})$. So $c'(OPT(G', c')) \leq c'(OPT(G, c))$ and $\overline{c}(OPT(G, \overline{c})) \leq \overline{c}(OPT(G, c))$. Thus $c(F) \leq 3 \cdot c'(OPT(G, c)) + 3 \cdot \overline{c}(OPT(G, c)) = 3 \cdot c(OPT(G, c))$, proving the Theorem. ■

9.2.2 An Improved 2-Approximation Algorithm

In this section, we present the algorithm of Bafna et al. [1], which extends the algorithm **LR-FVS** and obtains a 2-approximation for minimum feedback vertex set. A cycle C in graph G is called *semi-disjoint* if there is at most one vertex of degree (w.r.t. the graph G) more than 2 in C. The algorithm of [1] differs from the one in the previous section in dealing with graphs that have a semi-disjoint cycle. The modified approximation algorithm **LR-FVS-mod** is described below.

LR-FVS-mod(G, c):

Clean(G, c).

If $(V(G) = \phi)$ return ϕ.

If G contains a semi-disjoint cycle C, then

Let u_C be the vertex (if any) of degree more than 2 in C.

$c_{min} \leftarrow \min\{c_v : v \in C\}$.

$u_{min} \leftarrow \text{argmin}\{c_v : v \in C\}$.

Set $c''_v \leftarrow c_v - c_{min}$, for all $v \in C$; and $c''_v \leftarrow c_v$, for all $v \in V \setminus C$.

$G'' \leftarrow G \setminus (C \setminus u_C)$.

$F'' \leftarrow$ **LR-FVS-mod**(G'', c'').

Set $F \leftarrow F'' \cup \{u_{min}\}$.

Return F.

Else,

$\alpha \leftarrow \min\{c_v/(d_v - 1) : v \in V(G)\}$.

Set $\overline{c}_v \leftarrow \alpha \cdot (d_v - 1)$, for all $v \in V(G)$.

Set $c'_v \leftarrow c_v - \overline{c}_v$, for all $v \in V(G)$.

$\overline{F} \leftarrow \{v \in V(G) : c'_v = 0\}$.

$G' \leftarrow G \setminus \overline{F}$.

$F' \leftarrow$ **LR-FVS-mod**(G, c).

Set $F \subseteq F' \cup \overline{F}$ to be a minimal FVS in G.

Return F.

Note that when there is no semi-disjoint cycle, this algorithm is identical to algorithm **LR-FVS**. The improvement is due to the following claim about any minimal FVS (compare with Claim 9.2.2).

Claim 9.2.5 *Let $G = (V, E)$ be a clean graph, with no semi-disjoint cycles. If $c_v = \alpha \cdot (d_v - 1), \forall v \in V$, is a degree-1 weighted cost function on G, the cost of any minimal FVS in G is at most twice the cost of the minimum FVS in G.*

Proof: Let G have $n = |V|$ vertices and $m = |E|$ edges. We assume, without loss of generality, that $\alpha = 1$. The proof of Claim 9.2.2 shows that the cost of the minimum FVS in G is at least $m - n + 1$. Here we show that if there are no semi-disjoint cycles, any minimal FVS in G has cost at most $2(m - n + 1)$.

Let F be any minimal FVS in G and $|F| = f$. Let $\mathcal{T} = \{T_1, \ldots, T_k\}$ denote the connected components (trees) in $G \setminus F$. For any vertex $v \in F$, there is some component $T \in \mathcal{T}$ such that v has at least 2 edges going into T. Define $\pi : F \to \mathcal{T}$ to be a function mapping each vertex of F to some component in $\mathcal{T}$ that has at least two edges to that vertex. From the argument above, this function is well defined. Further, define the weight function $w : \mathcal{T} \to \mathbb{N}^+$ as $w(T_i) = |\pi^{-1}(T_i)|$, $\forall 1 \le i \le k$. Note that $\sum_{i=1}^{k} w(T_i) = f$.

Let c be the number of cross edges, between vertices in F and vertices in $V \setminus F$. For every $1 \le i \le k$, define $e(T_i)$ to be the number of edges from T_i to F. Clearly, $\sum_{i=1}^{k} e(T_i) = c$. We now have the following proposition.

Proposition 9.2.6 *For every $1 \le i \le k$, $e(T_i) \ge w(T_i) + 2$.*

Proof: Note that $w(T_i) \le \lfloor e(T_i)/2 \rfloor$, since edges from T_i to different vertices of F are disjoint.

The proof is by a case analysis. First suppose $e(T_i) \geq 4$. Then, $w(T_i)+2 \leq w(T_i)+e(T_i)/2 \leq e(T_i)$. If $e(T_i) = 3$, $w(T_i) + 2 \leq \lfloor e(T_i)/2 \rfloor + 2 = \lfloor 3/2 \rfloor + 2 = 3 = e(T_i)$.

Now suppose $e(T_i) = 2$. If both edges from T_i go to the same vertex in F, we would have a semi-disjoint cycle. So it must be that the two edges from T_i go to different vertices of F. This in turn implies that $w(T_i) = 0$, since T_i does not have two edges incident on any vertex of F. So again, $e(T_i) = 2 = w(T_i) + 2$. Finally, note that $e(T_i)$ can never be 1. Since T_i is a tree, this would mean that there is some vertex of degree 1 (w.r.t. G) in T_i. But this is not possible since G is a clean graph! ■

If we add the inequalities given by Proposition 9.2.6, we get $\sum_{i=1}^{k} e(T_i) \geq \sum_{i=1}^{k} w(T_i) + 2k$. In other words, $c \geq f + 2k$. Recall that in the proof of Claim 9.2.2, we showed that the cost of FVS F, $c(F) = 2(m-n) + f + 2k - c$. Now $c \geq f + 2k$, so $c(F) \leq 2(m - n + 1)$. Along with equation 9.2.1, this proves the claim. ■

The approximation guarantee of algorithm **LR-FVS-mod** is proved by the following Theorem, which is similar to Theorem 9.2.4.

Theorem 9.2.7 *For any instance (G, c) of minimum feedback vertex set, the cost of the feedback vertex set F returned by algorithm* **LR-FVS-mod** *is $c(F) \leq 2 \cdot c(OPT(G, c))$.*

Proof: The proof of this theorem is by induction on the number of vertices in the graph. The base case is when the graph is empty, and this case is trivial. In the inductive step, we consider the two cases depending on the presence of a semi-disjoint cycle.

The graph G has a semi-disjoint cycle C: We first argue that the solution F returned by the algorithm is a feedback vertex set. By induction on the smaller graph G'', we may assume that F'' is a feedback vertex set on G''. Since G'' differs from G only in the semi-disjoint cycle C, all other cycles of G are in G'' as well. So C is the only cycle of G that may not be covered by F''. But vertex $u_{min} \in C$ covers cycle C. So $F'' \cup \{u_{min}\}$ is a feedback vertex set in G.

We now argue about the cost of solution F. By induction, we have that $c''(F'') \leq 2 \cdot c''(OPT(G'', c''))$. Note that if S is any feedback vertex set in G, its restriction S'' to vertices in G'', is a feedback vertex set in G''. Also, S must have at least one vertex from the cycle C. So, $c(S) \geq c_{min} + c''(S) \geq c_{min} + c''(S'')$. Using $S = OPT(G, c)$, the optimal feedback vertex set in (G, c), we get $c''(OPT(G'', c'')) \leq c(OPT(G, c)) - c_{min}$. Now it follows that the cost of solution F is,

$$
\begin{aligned}
c(F) &= c(F'') + c(u_{min}) \\
&= c(F'') + c_{min} \\
&\leq c''(F'') + c_{min} + c_{min} \\
&\leq 2 \cdot c''(OPT(G'', c'')) + 2c_{min} \\
&\leq 2 \cdot c(OPT(G, c))
\end{aligned}
$$

The third inequality above follows from the fact that $F'' \subseteq V(G'')$ may contain at most one vertex (namely u_C) from the cycle C.

The graph G has no semi-disjoint cycle: This case is identical to the proof of Theorem 9.2.4, and is left to the reader to verify. ■

We note that there is a variant of this algorithm which was found independently by Becker & Geiger

[2]. In this variant, the algorithm finds a cost function that is proportional to the degree, instead of degree-1, and also finds a 2-approximate solution.

9.3 Weighted Vertex Cover

In this section we see how the local ratio method can be applied to get a 2-approximation algorithm for the weighted vertex cover problem. An instance of weighted vertex cover is given by an undirected graph $G = (V, E)$ with a non-negative cost function $c : V \to \mathbb{R}^+$ on its vertices. The cardinality vertex cover that we saw earlier is a special case of this problem, where all vertex weights are the same. We describe the algorithm **LR-WVC** below.

LR-WVC(G, c):
- If G has no edges, return ϕ.
- Pick an edge $e = (u, w)$.
- $\alpha \leftarrow \min\{c_u, c_w\}$.
- Set $\overline{c}_u, \overline{c}_w \leftarrow \alpha$; and $\overline{c}_v \leftarrow 0$, for all $v \in V \setminus \{u, w\}$.
- Set $c'_v \leftarrow c_v - \overline{c}_v$, for all $v \in V$.
- $\overline{F} \leftarrow \{v \in V : c'_v = 0\}$.
- $G' \leftarrow G \setminus \overline{F}$.
- $F' \leftarrow$ **LR-WVC**(G', c').
- Set $F \leftarrow F' \cup \overline{F}$.
- Return F.

One can observe that algorithms **LR-WVC** and **LR-FVS** are similar. Not surprisingly, the proof of the approximation guarantee of **LR-WVC** also follows that of **LR-FVS**. A cost function is called *one-edge weighted* if it assigns the same non zero cost to two vertices that are end points of some edge, and zero cost to all other vertices. The cost function $\overline{c}$ above is one-edge weighted. Similar to Claim 9.2.2, we have the following.

Claim 9.3.1 *If G is a graph with a one-edge weighted cost function $c : V \to \mathbb{R}^+$, the cost of any vertex cover in G is at most two times the minimum vertex cover.*

Proof: Left to the reader to verify. ■

Given an instance (G, c) of weighted vertex cover, we denote the optimal vertex cover by $OPT(G, c)$. So the optimum cost of this instance is $c(OPT(G, c)$. The approximation guarantee of **LR-WVC** is given by the following Theorem (compare with Theorem 9.2.4).

Theorem 9.3.2 *For any instance (G, c) of weighted vertex cover, the vertex cover F returned by algorithm* **LR-WVC** *has cost $c(F) \leq 2 \cdot c(OPT(G, c))$.*

Proof: The proof of this Theorem is by induction on the number of edges in the graph. The base case occurs when the graph has no edges, and this case is trivial. For the induction step, we first argue that the solution F is a vertex cover. By induction on the smaller graph G', F' is a vertex cover on G'. The only edges of G that are not covered by F' are those incident on $\overline{F}$. So $F = F' \cup \overline{F}$ is a vertex cover in G.

To argue the cost of solution F, note that we can write $c(F) = c'(F) + \overline{c}(F)$. $\overline{c}$ is a one-edge weighted cost function, and from Claim 9.3.1, we have $\overline{c}(F) \leq 2 \cdot \overline{c}(OPT(G, \overline{c}))$. Since $OPT(G, c)$ is a feasible solution to instance $(G, \overline{c})$, we have $\overline{c}(F) \leq 2 \cdot \overline{c}(OPT(G, c))$. Now, $c'(F) = c'(F') + c'(\overline{F}) = c'(F') \leq 2 \cdot c'(OPT(G', c'))$, by induction on (G', c'). $OPT(G, c)$ is a feasible solution to instance (G', c') as well; so $c'(OPT(G', c')) \leq c'(OPT(G, c))$. So we get $c(F) = c'(F) + \overline{c}(F) \leq 2 \cdot c'(OPT(G, c)) + 2 \cdot \overline{c}(OPT(G, c)) = 2 \cdot c(OPT(G, c))$. ∎

References

[1] Vineet Bafna, Piotr Berman, and Toshihiro Fujito. A 2-approximation algorithm for the undirected feedback vertex set problem. *SIAM J. Discret. Math.*, 12(3):289–297, 1999.

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