Approximating Minimum Subset Feedback Sets in Undirected Graphs with Applications

Guy Even, Joseph (Seffi) Naor, Baruch Schieber, and Leonid Zosin

Gustavo Ciotto Pinton

UNICAMP

University of Campinas - UNICAMP
MO418 - Approximation Algorithms

November 28th, 2018

Outline

- $G = (V, E)$ be an undirected graph with a weight function w associated with either the vertices or edges of G.
- A **feedback vertex set** (FVS) is defined to be a set of vertices that intersects all cycles in G.
- A **feedback edge set** (FES) is defined equally, but in respect to the edges.
- SUBSET-FVS or SUBSET-FES: restrict the set of cycles (**interesting cycles**) that the feedback set should intersect:
  - Let $S \subseteq V$ be a set of *special* vertices. A cycle is **interesting** if it contains a **special** vertex.
- We are interested in computing a **minimum weight subset feedback set**.

Introduction

Hardness

- Computing a minimum weight FVS is a classical NP-complete problem [R.M. KARP, *Reducibility among combinatorial problems*].
- FES problem can be solved in polynomial time: a minimum weight FES is the complement of a maximum weight spanning tree in G (Kruskal's algorithm).
- SUBSET-FES and SUBSET-FVS are NP-complete, even for $|S| = 1$:
  - **Multiway cut problem**: A multiway cut is a set of edges (or vertices) whose removal disconnects every pair of terminals in $T \subseteq V$. As seen in class, computing such a minimum cut in **NP-complete**.
  - Polynomial time reduction to the subset feedback set problem: connect a single special vertex s to all $t \in T$.
  - In vertex version, s has infinite weight. In the edge version, set the weight of all adjacent edges to s to infinite.
  - Any cycle passing through s corresponds to a path connecting two terminals in T.
  - A subset feedback set with respect to s is also a multiway cut of T.

Approximating the subset SUBSET-FES problem

Some details before the algorithm

- $G = (V, E)$ be an undirected graph with a weight function $w : E \to \mathbb{R}$.
- $S = \{s_1, s_2, \ldots s_k\}$ is the set of special vertices.
- Define $V_i \triangleq V - \{s_i, \ldots, s_k\}$. Consequently, $V_i \subset V_{i+1}$, $V_{k+1} = V$ and V_1 is the set of of nonspecial vertices.
- Without loss of generality, assume G is connected and that for each special vertex:
  - $deg(s) = 2$.
  - Its two neighbors are not special vertices.
  - its two adjacent edges have infinite weight.
- Whenever a vertex s does not satisfy these conditions:
  - For each edge e adjacent to s, split it in e_1, e_2 and e_3.
  - Add vertex s' between e_1 and e_2.
  - Add vertex s'' between e_2 and e_3.
  - $w(e_1) = w(e_2) = \infty$ and $w(e_3) = w(e)$.
  - Add s' to S and remove s from this set.

Approximating the SUBSET-FES problem

Algorithm

- Let the two neighbors of s_i be x_i and y_i.
- Algorithm:

Algorithm 1 SUBSET-FES

$M \leftarrow \emptyset$
for $i = 1$ to k **do**
  $M_i \leftarrow$ minimum cut between x_i and y_i in $G_i = (V, E - \cup_{j=1}^{i-1} M_j)$
end for
$M \leftarrow M_1 \cup M_2 \cup \ldots \cup M_k$
return M

- The solution is feasible.

Approximating the SUBSET-FES problem

Analysis

- Let OPT denote a minimum weight solution.
- Define $H = (V, E - OPT)$ and H_i as the subgraph of H induced by V_i.
- **Fact:** H does not contain any interesting cycles.
- **Claim:** the number of connected components $|C|$ in H_1 is $k + 1$.
- *Proof:*
  - H_1 consists of connected components: for each special vertex s_i, x_i and y_i cannot belong to the same connected component. Otherwise, there would be another interesting cycle.
  - Build H' by contracting each component of H_1 into a vertex.
  - Since H does not have any cycle, H' must be a tree.
  - Number of edges in H' is $|E'| = 2k$, since each s_i has two neighbors.
  - **Euler's theorem**: $|V'| + |F'| = |E'| + 2$. With $|F'| = 1$ and $|E'| = 2k \implies |V'| = 2k + 1$.
  - Therefore $|C| = |V'| - |S| = k + 1$.

Approximating the SUBSET-FES problem

Analysis (II)

- Connected components of $H_1 : C_1, \dots, C_{k+1}$.
- Let OPT_i denote the edges in OPT with one endpoint in C_i.
- Each edge in OPT touches two connected components $\implies \sum_{i=1}^{k+1} w(OPT_i) = 2w(OPT)$.
- Injective mapping from $M_1, \dots, M_k$ to $C_1, \dots, C_{k+1}$, i.e., each component can be the image of at most one cut:
- For each M_i, we identify $C_{l(i)}$ and $C_{r(i)}$:
  - p is a simple path from x_i to y_i in $G_i = (V, E - \cup_{j=1}^{i-1} M_j)$.
  - Discard from p all nonspecial vertices, except for x_i and y_i.
  - The remaining vertices, in order of appearance, are called the *backbone* of p.
  - Follow the *backbone*, from x_i until the first vertex v that is not connected in H_i to its subsequent vertex in the backbone.
  - Such a vertex always exists since the p is disconnected by OPT.
  - v can be either x_i or a special vertex s_j such that $j < i$.
  - Define $C_{l(i)} = C_j$ such that $v \in C_j$.
  - Define $C_{r(i)}$ similarly by considering p in reverse order.

Approximating the SUBSET-FES problem

Analysis (III)

- **Claim 1:** All simple paths connecting x_i with y_i in G_i have the same backbone.
- *Proof*:
  - Consider two paths p and q and suppose that they have different sequences of special vertices.
  - Union of p and q must contain an interesting cycle.
  - Let $j < i$ be the maximum index of a special vertex in this cycle.
  - Path connecting x_j and y_j, obtained by removing s_j from the cycle, is in G_j.
  - M_j must intersect this path and it cannot belong to $G_i \implies$ contradiction.

Approximating the SUBSET-FES problem

Analysis (IV)

- **Claim 2:** For $i = 1, \ldots, k$, $w(M_i) \leq \min\{w(OPT_{l(i)}), w(OPT_{r(i)})\}$
- *Proof* $w(M_i) \leq w(OPT_{l(i)})$:
  - By Claim 1, all the paths from x_i to y_i in G_i traverse a vertex in $C_{l(i)}$.
  - By definition of $C_{l(i)}$ all these paths emanate from $C_{l(i)}$ using an edge from $OPT_{l(i)} \implies OPT_{l(i)}$ cuts them all.
  - Since M_i is a cut of minimum weight, $w(M_i) \leq w(OPT_{l(i)})$.
- *Proof* $w(M_i) \leq w(OPT_{r(i)})$:
  - It can be shown as the previous case.
- **Injective mapping**:
  - Auxiliary graph H'': vertex set consists of the the special vertices and one vertex for each connected component of H_1.
  - H'' is a tree for the same reasons H' is. It has $2k$ edges and $2k+1$ vertices.
  - Root this tree at an arbitrary nonspecial vertex.
  - Each special vertex s_i has exactly one child which corresponds to either $C_{l(i)}$ or $C_{r(i)}$.
  - Map M_i to this component.

- **Theorem:** Algorithm SUBSET-FES computes a feasible solution whose weight is at most twice the weight of an optimal solution.
- *Proof:*
  - $\sum_{i=1}^{k} M_i \leq \sum_{i=1}^{k} \min\{w(OPT_{l(i)}), w(OPT_{r(i)})\}$ (Claim 2)
  - $\sum_{i=1}^{k} \min\{w(OPT_{l(i)}), w(OPT_{r(i)})\} \leq \sum_{i=1}^{k} w(OPT_{M(i)})$ (Mapping)
  - $\sum_{i=1}^{k} w(OPT_{M(i)}) \leq \sum_{i=1}^{k+1} w(OPT_i) = 2w(OPT)$ (Definition)
  - Therefore $\sum_{i=1}^{k} M_i \leq 2w(OPT)$.
- The algorithm is tight.

Approximating the SUBSET-FVS problem

Analysis

- With loss of generality, for each special vertex s: $w(s) = \infty$ and its two neighbors have infinite weight too.
- Same approximation algorithm as SUBSET-FES, however:
  - M_i is a minimum weight vertex cut between x_i and y_i.
  - Same algorithm as before. In iteration i, $G_i = (V - \cup_{j=1}^{i-1} M_j, E)$. Return $\cup_{i=1}^{k} M_i$.
  - $H = (V - OPT, E)$.
  - H_1 has at least $k+1$ connected components, denoted by $C_1, \ldots, C_{k+1}, \ldots, C_n$.
  - OPT_i are the vertices in OPT neighbors of C_i.
  - Δ is the maximum vertex degree.
  - Since every vertex of OPT may be the neighbor of at most Δ components, $\sum_{i=1}^{k+1} w(OPT_i) \leq \Delta w(OPT)$.
  - By repeating a similar analysis done for SUBSET-FES (finding an injective mapping), we can show that $\sum_{i=1}^{k} w(M_i) \leq \Delta w(OPT)$.

Linear Programming Formulation and Integrality Gap

Definitions

- Given a **minimization problem**, the integrality gap is defined as

$$\mathrm{IG} = \max_{I} \frac{PLI(I)}{PL(I)}$$

  where **PLI** is the integer linear problem's solution and **PL**, its fractional relaxation.
- **Integer programming formulation** for FES and FVS:
  - Let G = (V,E) be an undirected graph with $|V| = n$.
  - Denote $\mathcal{C}$ the set of cycles in G.
  - $\mathcal{C}_v$ is set of cycles passing through vertex v.
  - Let $x_v \in \{0, 1\}$ (or x_e) be an indicator variable for membership in a feedback set in G.

Linear Programming Formulation and Integrality Gap

Formulation for FES and FVS

- Integer programming formulation for **FVS**:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{v \in V} w(v) * x_v \\
\text{subject to} \quad & \sum_{v \in C} x_v \geq 1 \text{ for every } C \in \mathcal{C} \\
& x_v \in \{0, 1\} \text{ for every } v \in V
\end{aligned}
$$

- Integer programming formulation for **FES**:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{e \in E} w(e) * x_e \\
\text{subject to} \quad & \sum_{e \in C} x_e \geq 1 \text{ for every } C \in \mathcal{C} \\
& x_e \in \{0, 1\} \text{ for every } e \in E
\end{aligned}
$$

- We can extend the previous problems to consider only the cycles containing the special vertices S.
- Integer programming formulation for **SUBSET-FVS**:

$$\begin{aligned} \text{minimize} \quad & \sum_{v \in V} w(v) * x_v \\ \text{subject to} \quad & \sum_{v \in C} x_v \geq 1 \text{ for every } C \in \bigcup_{s \in S} \mathcal{C}_s \\ & x_v \in \{0, 1\} \text{ for every } v \in V \end{aligned}$$

- Integer programming formulation for **SUBSET-FES**:

$$\begin{aligned} \text{minimize} \quad & \sum_{e \in E} w(e) * x_e \\ \text{subject to} \quad & \sum_{e \in C} x_e \geq 1 \text{ for every } C \in \bigcup_{s \in S} \mathcal{C}_s \\ & x_e \in \{0, 1\} \text{ for every } e \in E \end{aligned}$$

Linear Programming Formulation and Integrality Gap

Integrality Gap for FES and FVS

- Let's show that the integrality gap in the FVS and FES can be as big as $\Omega(\log |V|)$.
  - Let $H_n = (V, E)$ be a connected, 3-regular graph, with $|V(H_n)| = n$ and girth of H_n is $\lceil \log n \rceil$.
  - Such graphs exist and can be constructed explicitly. (See, e.g., Bollobás [6, pp. 108–110]).
  - Set all weights (edges or vertices) in G to 1.
  - **Claim 1:** *The value of the optimal fractional FVS solution in H_n is at most $\frac{n}{\lceil \log n \rceil}$. The value of the optimal fractional FES solution in H_n is at most $1.5\frac{n}{\lceil \log n \rceil}$.*
  - *Proof:*
    - Length of the shortest cycle contained in G is $\lceil \log n \rceil$.
    - A feasible fractional solution for FVS could be $x_v = \frac{1}{\lceil \log n \rceil} \forall v \in V$.

      Its cost is $\displaystyle\sum_{v \in V} w(v) * x_v = \sum_{v \in V} \frac{1}{\lceil \log n \rceil} = \frac{n}{\lceil \log n \rceil}$
    - For FES, it could be $x_e = \frac{1}{\lceil \log n \rceil} \forall e \in E$.
    - $|E| = 1.5n$, since $\displaystyle\sum_{v \in V} deg(v) = 2|E|$ and $deg(v) = 3 \ \forall v \in V$.
    - Therefore, FES solution's cost would be $\displaystyle\sum_{e \in E} w(e) * x_e = 1.5\frac{n}{\lceil \log n \rceil}$.

Linear Programming Formulation and Integrality Gap

Integrality Gap for FES and FVS (II)

- An optimal integral solution for the FES is the complement of any spanning tree of H_n.
- Its cost is $1.5n - (n-1) = \frac{n}{2} + 1$.
- Therefore, $IG_{FES} = \frac{\frac{n}{2}+1}{1.5\frac{n}{\lceil \log n \rceil}} = \lceil \log n \rceil \frac{n+2}{3n}$, which is $\Omega(\log |V|)$.
- **Claim 2:** *The value of an optimal integral solution for the FVS problem in H_n is $\Omega(n)$.*
- *Proof:*
  - An optimal solution to the FES problem costs at most $\Delta = 3$ times the cost of an optimal solution to the FVS problem.
  - Therefore, the cost of the FVS is at least $\frac{\frac{n}{2}+1}{3} = \frac{n}{6} + \frac{1}{3}$, which is $\Omega(n)$.
- We have then $IG_{FVS} = \frac{\frac{n}{6}+\frac{1}{3}}{\frac{n}{\lceil \log n \rceil}} = \lceil \log n \rceil \frac{n+2}{6n}$, which is also $\Omega(\log |V|)$.

- Let F_n be the union of two graphs:
  - $H_k = (S, E)$, with $|S| = k$.
  - Clique C_{n-k} on the remaining $n - k$ vertices.
- Connect H_k and C_{n-k} by a *single* edge.
- No cycle that goes through a vertex in S intersects the clique.
- Thus, for integrality gap, consider only H_k which has k vertices.
- By using claims 1 and 2, we obtain that $IG_{SUBSET-FES}$ and $IG_{SUBSET-FVS}$ are in $\Omega(\log k)$.