A 2-Approximation Algorithm for Feedback Vertex Set in Tournaments

Daniel Lokshtanov[1], Pranabendu Misra[2], Joydeep Mukherjee[3], Fahad Panolan[2], G Philip[4], and Saket Saurabh[2,5]

[1] *UC Santa Barbara, California, USA*
[2] *University of Bergen, Bergen, Norway*
[3] *Indian Statistical Institute, Kolkata, India*
[4] *Chennai Mathematical Institute, Chennai, India*
[5] *The Institute of Mathematical Sciences, Chennai, India*

Recent Trends in Algorithms, NISER Bhubaneswar
February 9, 2019

Definitions

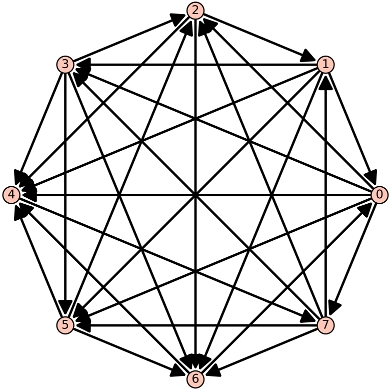
Tournaments

- A *tournament* is a directed graph in which there is exactly one arc between any two vertices.
  - Take a complete graph and give each edge an orientation.

Definitions

Tournaments

- A *tournament* is a directed graph in which there is exactly one arc between any two vertices.
  - Take a complete graph and give each edge an orientation.



- Observation: Deleting vertices preserves the tournament property

Definitions

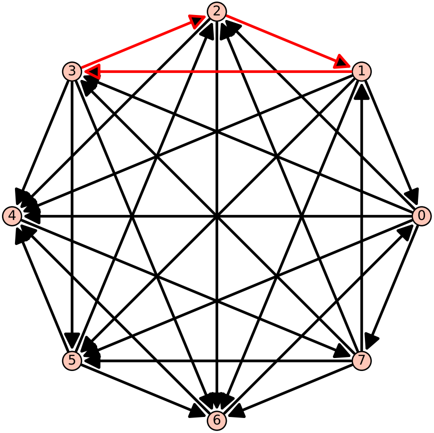
Acyclic Tournaments

- A tournament is *acyclic* if it does not contain any directed cycle.
  - The example tournament is *not* acyclic

Definitions

Acyclic Tournaments

- A tournament is *acyclic* if it does not contain any directed cycle.
  - The example tournament is *not* acyclic



Definitions

Acyclic Tournaments

- A tournament is *acyclic* if it does not contain any directed cycle.
  - The example tournament is *not* acyclic

- A tournament
  - has a directed cycle if and only if it has a directed *triangle*
  - is acyclic if and only if it contains *no* directed triangle

Definitions

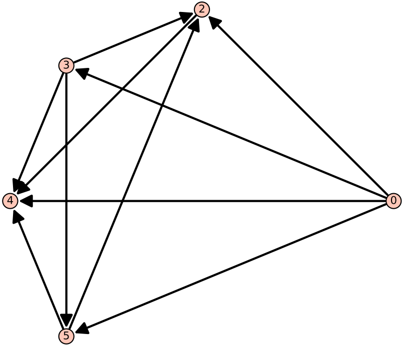
Acyclic Tournaments

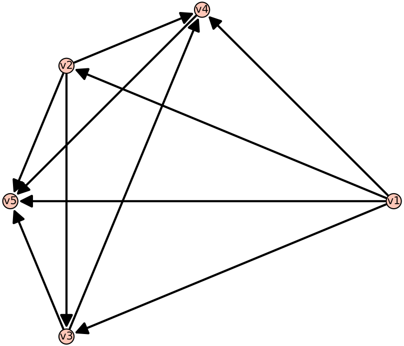
- A tournament is *acyclic* if it does not contain any directed cycle.
  - The example tournament is *not* acyclic
- A tournament
  - has a directed cycle if and only if it has a directed *triangle*
  - is acyclic if and only if it contains *no* directed triangle
- An acyclic tournament has a *unique* topological ordering of vertices
  - We can re-label its vertices as $v_1, v_2, \ldots, v_n$ such that every arc is *from* a "smaller" vertex *to* a "larger" vertex.
  - In exactly one way.

Definitions

Topological Ordering

- An acyclic tournament has a *unique* topological ordering of vertices





Definitions

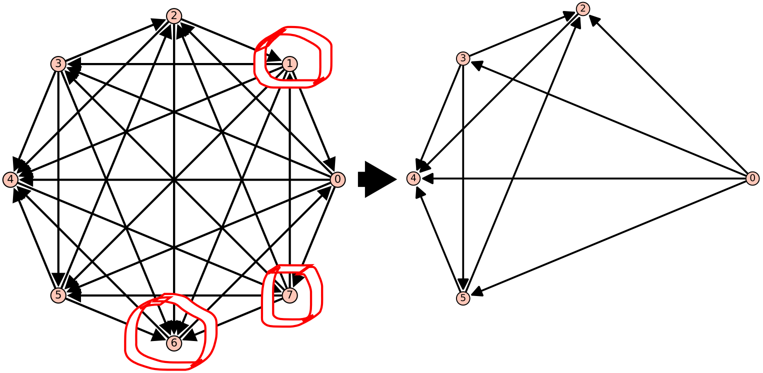
Feedback Vertex Sets

- A *feedback vertex set* (FVS for short) of a tournament T is any subset S of its vertices such that *deleting* S from T gives an acyclic tournament.

Definitions

Feedback Vertex Sets

- A *feedback vertex set* (FVS for short) of a tournament T is any subset S of its vertices such that *deleting* S from T gives an acyclic tournament.



The Problem

Weighted Feedback Vertex Set in Tournaments

- Input: A *tournament* $T = (V, A)$ and a *weight function* $w : V \to \mathbb{N}$
  - Non-negative integral weights on vertices
- Task: Find a feedback vertex set of T of the *smallest total weight*

Weighted Feedback Vertex Set in Tournaments

Some known results

NP-hardness

- Is NP-hard, even in the unweighted case
  - When all vertices have the same weight

Weighted Feedback Vertex Set in Tournaments

Some known results

NP-hardness

- Is NP-hard, even in the unweighted case
  - When all vertices have the same weight

Polynomial-time approximation algorithms

- Unweighted case: Simple 3-factor approximation algorithm
- Weighted case:
  - 3-approximation: Local-ratio technique
  - $\frac{5}{2}$-approximation: Cai et al., 2000
    - Local-ratio technique
  - $\frac{7}{3}$-approximation: Mnich, Vassilevska-Williams, and Végh, ESA 2016
    - Iterative rounding

Weighted Feedback Vertex Set in Tournaments

Some known results

NP-hardness

- Is NP-hard, even in the unweighted case
  - When all vertices have the same weight

Polynomial-time approximation algorithms

- Best known approximation ratio, weighted case:
  - $\frac{7}{3}$: Mnich et al., 2016
- Under the Unique Games Conjecture:
  - No $(2 - \varepsilon)$-approximation
  - Even for the unweighted case
  - Reduction from Vertex Cover

Weighted Feedback Vertex Set in Tournaments

Our results

- A randomized polynomial-time 2-factor approximation algorithm
  - Runs in time $\mathcal{O}(n^c)$
  - Outputs an FVS
  - Is a 2-factor-approximate solution with probability $\frac{1}{2}$
- Derandomized in quasi-polynomial time

Our 2-Approximation Algorithm

Main Ingredients

- The Local Ratio Technique
- Randomization
- Divide and Conquer

The Local Ratio Technique

Weighted FVS in Tournaments, 3-Approximation

- Input: A tournament $T = (V, A)$ and a weight function $w : V \to \mathbb{N}$
- Task: Find a feedback vertex set of T of the *smallest total weight*

The Local Ratio Technique

Weighted FVS in Tournaments, 3-Approximation

- Input: A tournament $T = (V, A)$ and a weight function $w : V \to \mathbb{N}$
- Task: Find a feedback vertex set of T of the *smallest total weight*

- Recall: Necessary and sufficient to "hit" all directed triangles

The Local Ratio Technique

Weighted FVS in Tournaments, 3-Approximation

- Input: A tournament $T = (V, A)$ and a weight function $w : V \to \mathbb{N}$
- Task: Find a feedback vertex set of T of the *smallest total weight*

- Find a triangle with all three weights positive
- Subtract the least weight from all three weights
  - At least one vertex weight becomes zero
- Repeat this till no triangle has all three weights positive
- Return the set of zero-weight vertices

The Local Ratio Technique

Weighted FVS in Tournaments, 3-Approximation

- Input: A tournament $T = (V, A)$ and a weight function $w : V \to \mathbb{N}$
- Task: Find a feedback vertex set of T of the *smallest total weight*

- Find a triangle with all three weights positive
- Subtract the least weight from all three weights
  - At least one vertex weight becomes zero
- Repeat this till no triangle has all three weights positive
- Return the set of zero-weight vertices
  - Gets us a 3-approximate solution in the *unweighted* case ...
  - ... and *also* in the weighted case.

The Local Ratio Technique

Weighted FVS in Tournaments, 3-Approximation, contd.

- Claim 1: The set S of zero-weight vertices in the final graph is a feedback vertex set of T.
  - Proof

The Local Ratio Technique

Weighted FVS in Tournaments, 3-Approximation, contd.

- Claim 1: The set S of zero-weight vertices in the final graph is a feedback vertex set of T,
- Claim 2: ... of weight at most 3 times the weight of an optimum FVS.

The Local Ratio Technique

Weighted FVS in Tournaments, 3-Approximation, contd.

- Claim 1: The set S of zero-weight vertices in the final graph is a feedback vertex set of T,
- Claim 2: ... of weight at most 3 times the weight of an optimum FVS.
  - The total original weight of the vertices in S is not more than the total weight we reduced from all vertices during the procedure

The Local Ratio Technique

Weighted FVS in Tournaments, 3-Approximation, contd.

- Claim 1: The set S of zero-weight vertices in the final graph is a feedback vertex set of T,
- Claim 2: ... of weight at most 3 times the weight of an optimum FVS.
  - The total original weight of the vertices in S is not more than the total weight we reduced from all vertices during the procedure
  - If we reduced a total weight of $3r$ in a round then the weight of an optimal FVS reduced by *at least* r
    - Let $\{x, y, z\}$ be the triangle we modified in this round
    - Let $w(x) = r$. Then $w(y) \geq r$, $w(z) \geq r$
    - An optimal FVS S of the pre-round graph *must* contain at least one of $\{x, y, z\}$
    - The *same* set S is an FVS of the post-round graph, now with weight lesser by at least r

The Local Ratio Technique

Weighted FVS in Tournaments, 3-Approximation, contd.

- Claim 1: The set S of zero-weight vertices in the final graph is a feedback vertex set of T,
- Claim 2: ... of weight at most 3 times the weight of an optimum FVS.
  - The total original weight of the vertices in S is not more than the total weight we reduced from all vertices during the procedure
  - If we reduced a total weight of $3r$ in a round then the weight of an optimum FVS reduced by *at least* r
  - If we reduced a total weight of $3q$ over *all* rounds then the total reduction in the weight of an optimum FVS is *at least* q

The Local Ratio Technique

Weighted FVS in Tournaments, 3-Approximation, contd.

- Claim 1: The set S of zero-weight vertices in the final graph is a feedback vertex set of T,
- Claim 2: … of weight at most 3 times the weight of an optimum FVS.
  - The total original weight of the vertices in S is not more than the total weight we reduced from all vertices during the procedure
  - If we reduced a total weight of $3r$ in a round then the weight of an optimum FVS reduced by *at least* r
  - If we reduced a total weight of $3q$ over *all* rounds then the total reduction in the weight of an optimum FVS is *at least* q
  - The weight of an optimum FVS of the final instance is zero

The Local Ratio Technique

Weighted FVS in Tournaments, 3-Approximation, contd.

- Claim 1: The set S of zero-weight vertices in the final graph is a feedback vertex set of T,
- Claim 2: … of weight at most 3 times the weight of an optimum FVS.
  - The total original weight of the vertices in S is not more than the total weight we reduced from all vertices during the procedure
  - If we reduced a total weight of $3r$ in a round then the weight of an optimum FVS reduced by *at least* r
  - If we reduced a total weight of $3q$ over *all* rounds then the total reduction in the weight of an optimum FVS is *at least* q
  - The weight of an optimum FVS of the final instance is zero
  - The weight of an optimum FVS of the *original* instance is *at least* q

The Local Ratio Technique

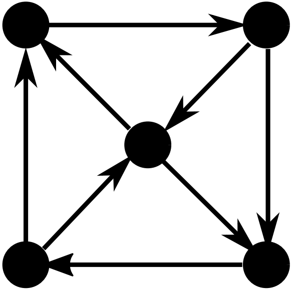
Weighted FVS in Tournaments, 3-Approximation, contd.

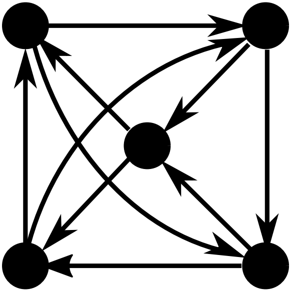
- Claim 1: The set S of zero-weight vertices in the final graph is a feedback vertex set of T,
- Claim 2: ... of weight at most 3 times the weight of an optimum FVS.
  - The total original weight of the vertices in S is not more than the total weight we reduced from all vertices during the procedure
  - If we reduced a total weight of $3r$ in a round then the weight of an optimum FVS reduced by *at least* r
  - If we reduced a total weight of $3q$ over *all* rounds then the total reduction in the weight of an optimum FVS is *at least* q
  - The weight of an optimum FVS of the final instance is zero
  - The weight of an optimum FVS of the *original* instance is *at least* q
  - $w(S) \leq 3q \leq 3 \times$ (wt. of an optimum FVS of the original instance)

The Local Ratio Technique

Weighted FVS in Tournaments, $\frac{5}{2}$-Approximation

- Cai et al. found two graphs:

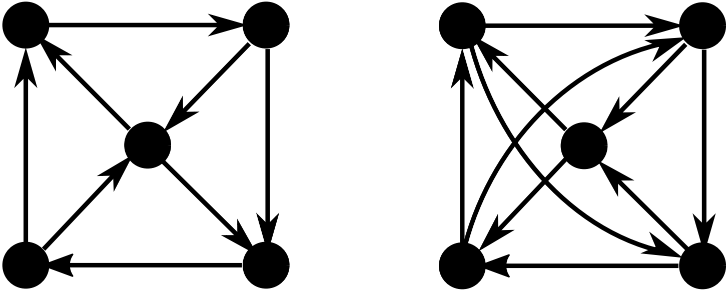




The Local Ratio Technique

Weighted FVS in Tournaments, $\frac{5}{2}$-Approximation

- Cai et al. found two graphs:



- *Any* FVS must pick at least two of the five vertices
- If *neither* graph is present in a tournament:
  - The Weighted FVS problem is polynomial-time solvable

The Local Ratio Technique

Weighted FVS in Tournaments, $\frac{5}{2}$-Approximation

- Due to Cai et al.
- Two five-vertex graphs
  - If present, must pick at least two vertices
  - If not present, polynomial-time solvable!
- Now apply the local ratio technique
  - $\frac{5}{2}$-approximation

The Local Ratio Technique

How to get a 2-Approximation?

- One way could be:
  - Find (say) a set of 10-vertex graphs,
  - from each of which at least 5 vertices must be picked
  - whose absence gives a polynomial-time solvable instance
- Sounds like hard work!
  - (Why should these even exist?)

Our Idea: Use The Local Ratio Technique . . .

. . . "on steroids"

- We find one graph on two vertices
  - from which at least one vertex must be picked
  - and whose absence gives a polynomial-time solvable instance
- (. . . more or less.)

Our Idea: Use The Local Ratio Technique . . .

. . . "on steroids"

- We find one graph on two vertices
  - from which at least one vertex must be picked
  - and whose absence gives a polynomial-time solvable instance
- (. . . more or less.)
- The "steroids":
  - A "global" take on the local ratio technique
  - Randomization
  - Plain old Divide and Conquer

Our Idea: Use The Local Ratio Technique . . .

. . . "on steroids"

- We find one graph on two vertices
  - from which at least one vertex must be picked
  - and whose absence gives a polynomial-time solvable instance
- (. . . more or less.)
- The "steroids":
  - A "global" take on the local ratio technique
  - Randomization
  - Plain old Divide and Conquer
- All three are (well-)known ideas

A Generalized Local Ratio Technique

Applies when there is an optimum solution with **many** vertices

- Input:
  - Tournament $T = (V, A)$; $|V| = n$
  - Weight function $w : V \to \mathbb{N}$
- Suppose there is an optimal solution $S^\star$; $|S^\star| \geq \frac{2n}{3}$
  - Let L be a set of $\frac{n}{6}$ vertices of the smallest weight
  - Then $\frac{w(L)}{w(S^\star)} \leq \frac{1}{6} / \frac{2}{3} = \frac{1}{4}$

A Generalized Local Ratio Technique

Applies when there is an optimum solution with **many** vertices

- Input:
  - Tournament $T = (V, A)$; $|V| = n$
  - Weight function $w : V \to \mathbb{N}$
- Suppose there is an optimal solution $S^\star$; $|S^\star| \geq \frac{2n}{3}$
  - Let L be a set of $\frac{n}{6}$ vertices of the smallest weight
  - Then $\frac{w(L)}{w(S^\star)} \leq \frac{1}{6} / \frac{2}{3} = \frac{1}{4}$
- To get a $\frac{5}{4}$-approximation:
  - Pick *all* of L
  - Find an optimum solution for $G - L$
  - Take their union

A Generalized Local Ratio Technique

Applies when there is an optimum solution with **many** vertices

- Input:
  - Tournament $T = (V, A)$; $|V| = n$
  - Weight function $w : V \to \mathbb{N}$
- Suppose there is an optimal solution $S^\star$; $|S^\star| \geq \frac{2n}{3}$
  - Let L be a set of $\frac{n}{6}$ vertices of the smallest weight
  - Then $\frac{w(L)}{w(S^\star)} \leq \frac{1}{6} / \frac{2}{3} = \frac{1}{4}$
- To get a $\frac{5}{4}$-approximation:
  - Pick *all* of L
  - Find an optimum solution for $G - L$
  - Take their union

A Generalized Local Ratio Technique

Applies when there is an optimum solution with **many** vertices

- Input:
  - Tournament $T = (V, A)$; $|V| = n$
  - Weight function $w : V \to \mathbb{N}$
- Suppose there is an optimal solution $S^\star$; $|S^\star| \geq \frac{2n}{3}$
  - Let L be a set of $\frac{n}{6}$ vertices of the smallest weight
  - Then $\frac{w(L)}{w(S^\star)} \leq \frac{1}{6} / \frac{2}{3} = \frac{1}{4}$
- To get a 2-approximation:
  - Pick *all* of L
  - Let $max_L = \max_{v \in L} w(v)$
  - Set $w' : (V \setminus L) \to \mathbb{N}$
    - $w'(x) = w(x) - max_L$
  - Find a 2-approximate solution for $((T - L), w')$
  - Take their union

A Generalized Local Ratio Technique

Optimal solution $S^\star$; $|S^\star| \geq \frac{2n}{3}$

- L: $\frac{n}{6}$ vertices of the smallest weight
- $max_L = \max_{v \in L} w(v)$
- $w' : (V \setminus L) \rightarrow \mathbb{N}$
  - $w'(x) = w(x) - max_L$
- Reduced instance $R = ((T - L), w')$
- R_{approx}: 2-approximate solution for $((T - L), w')$
- **Claim:** $L \cup R_{approx}$ is a 2-approximate solution for (T, w)

A Generalized Local Ratio Technique

$S^\star$: Optimal solution, $|S^\star| \geq \frac{2n}{3}$; L: $\frac{n}{6}$ vertices of least weight
$w'(x) = w(x) - max_L, \;\; R = ((T - L), w')$
R_{approx}: 2-approximation for R

- **Claim:** $L \cup R_{approx}$ is a 2-approximate solution for (T, w)
- Intuition:
  - $S^\star \setminus L$
    1. is very large compared to L, and
    2. is a solution to the reduced instance R
  - Reducing the weight of vertices in $S^\star \setminus L$ by max_L causes a very large drop in the optimum value for R
  - Enough to accommodate putting all of L back in to a 2-approximate solution

A Generalized Local Ratio Technique

$S^\star$: Optimal solution, $|S^\star| \geq \frac{2n}{3}$; L: $\frac{n}{6}$ vertices of least weight
$w'(x) = w(x) - max_L$, $R = ((T - L), w')$
R_{approx}: 2-approximation for R

- **Claim:** $L \cup R_{approx}$ is a 2-approximate solution for (T, w)
- Proof: Let $R^\star$ be an optimum solution for $((T - L), w')$
  - $w'(R_{approx}) \leq 2w'(R^\star)$

A Generalized Local Ratio Technique

$S^\star$: Optimal solution, $|S^\star| \geq \frac{2n}{3}$; L: $\frac{n}{6}$ vertices of least weight
$w'(x) = w(x) - max_L$, $R = ((T - L), w')$
R_{approx}: 2-approximation for R

- **Claim:** $L \cup R_{approx}$ is a 2-approximate solution for (T, w)
- Proof: Let $R^\star$ be an optimum solution for $((T - L), w')$
  - $w'(R_{approx}) \leq 2w'(R^\star)$
  - $w'(R^\star) \leq w'(S^\star \setminus L)$

A Generalized Local Ratio Technique

$S^\star$: Optimal solution, $|S^\star| \geq \frac{2n}{3}$; L: $\frac{n}{6}$ vertices of least weight
$w'(x) = w(x) - max_L$, $R = ((T - L), w')$
R_{approx}: 2-approximation for R

- **Claim:** $L \cup R_{approx}$ is a 2-approximate solution for (T, w)
- Proof: Let $R^\star$ be an optimum solution for $((T - L), w')$
  - $w'(R_{approx}) \leq 2w'(R^\star)$
  - $w'(R^\star) \leq w'(S^\star \setminus L)$
  - $w'(S^\star \setminus L) = w(S^\star \setminus L) - |S^\star \setminus L| \cdot max_L \leq w(S^\star) - |S^\star \setminus L| \cdot max_L$

A Generalized Local Ratio Technique

$S^\star$: Optimal solution, $|S^\star| \geq \frac{2n}{3}$; L: $\frac{n}{6}$ vertices of least weight
$w'(x) = w(x) - max_L, \;\; R = ((T-L), w')$
R_{approx}: 2-approximation for R

- **Claim:** $L \cup R_{approx}$ is a 2-approximate solution for (T, w)
- Proof: Let $R^\star$ be an optimum solution for $((T-L), w')$
  - $w'(R_{approx}) \leq 2w'(R^\star)$
  - $w'(R^\star) \leq w'(S^\star \setminus L)$
  - $w'(S^\star \setminus L) = w(S^\star \setminus L) - |S^\star \setminus L| \cdot max_L \leq w(S^\star) - |S^\star \setminus L| \cdot max_L$
  - $|S^\star \setminus L| \geq \frac{n}{2}$

A Generalized Local Ratio Technique

$S^\star$: Optimal solution, $|S^\star| \geq \frac{2n}{3}$; L: $\frac{n}{6}$ vertices of least weight
$w'(x) = w(x) - max_L, \quad R = ((T - L), w')$
R_{approx}: 2-approximation for R

- **Claim:** $L \cup R_{approx}$ is a 2-approximate solution for (T, w)
- Proof: Let $R^\star$ be an optimum solution for $((T - L), w')$
  - $w'(R_{approx}) \leq 2w'(R^\star)$
  - $w'(R^\star) \leq w'(S^\star \setminus L)$
  - $w'(S^\star \setminus L) = w(S^\star \setminus L) - |S^\star \setminus L| \cdot max_L \leq w(S^\star) - |S^\star \setminus L| \cdot max_L$
  - $|S^\star \setminus L| \geq \frac{n}{2}$
  - $w'(S^\star \setminus L) \leq w(S^\star) - \frac{max_L \cdot n}{2}$

A Generalized Local Ratio Technique

$S^\star$: Optimal solution, $|S^\star| \geq \frac{2n}{3}$; L: $\frac{n}{6}$ vertices of least weight
$w'(x) = w(x) - max_L, \;\; R = ((T-L), w')$
R_{approx}: 2-approximation for R

- **Claim:** $L \cup R_{approx}$ is a 2-approximate solution for (T, w)
- Proof: Let $R^\star$ be an optimum solution for $((T-L), w')$
  - $w'(R_{approx}) \leq 2w'(R^\star)$
  - $w'(R^\star) \leq w'(S^\star \setminus L)$
  - $w'(S^\star \setminus L) = w(S^\star \setminus L) - |S^\star \setminus L| \cdot max_L \leq w(S^\star) - |S^\star \setminus L| \cdot max_L$
  - $|S^\star \setminus L| \geq \frac{n}{2}$
  - $w'(S^\star \setminus L) \leq w(S^\star) - \frac{max_L \cdot n}{2}$
  - $w'(R_{approx}) \leq 2w(S^\star) - max_L \cdot n$

A Generalized Local Ratio Technique

$S^\star$: Optimal solution, $|S^\star| \geq \frac{2n}{3}$; L: $\frac{n}{6}$ vertices of least weight
$w'(x) = w(x) - max_L$, $R = ((T - L), w')$
R_{approx}: 2-approximation for R

- **Claim:** $L \cup R_{approx}$ is a 2-approximate solution for (T, w)
- Proof: Let $R^\star$ be an optimum solution for $((T - L), w')$
  - $\mathbf{w'(R_{approx}) \leq 2w(S^\star) - max_L \cdot n}$
  - $$\begin{aligned} w(R_{approx}) &= w'(R_{approx}) + |R_{approx}| \cdot max_L \\ &\leq (2w(S^\star) - max_L \cdot n) + |R_{approx}| \cdot max_L \\ &= 2w(S^\star) - max_L(n - |R_{approx}|) \end{aligned}$$

A Generalized Local Ratio Technique

$S^\star$: Optimal solution, $|S^\star| \geq \frac{2n}{3}$; L: $\frac{n}{6}$ vertices of least weight
$w'(x) = w(x) - max_L, \ \ R = ((T - L), w')$
R_{approx}: 2-approximation for R

- **Claim:** $L \cup R_{approx}$ is a 2-approximate solution for (T, w)
- Proof: Let $R^\star$ be an optimum solution for $((T - L), w')$
  - $\mathbf{w'(R_{approx}) \leq 2w(S^\star) - max_L \cdot n}$
  - $\mathbf{w(R_{approx}) \leq 2w(S^\star) - max_L(n - |R_{approx}|)}$
  - $$\begin{aligned} w(L \cup R_{approx}) &= w(R_{approx}) + w(L) \\ &\leq (2w(S^\star) - max_L(n - |R_{approx}|) + |L| \cdot max_L) \\ &= (2w(S^\star) - max_L(n - |R_{approx}| - |L|)) \\ &= (2w(S^\star) - max_L(n - |R_{approx} \cup L|)) \\ &\leq 2w(S^\star). \end{aligned}$$

A Generalized Local Ratio Technique

Optimal solution $S^\star$; $|S^\star| \geq \frac{2n}{3}$

- L: $\frac{n}{6}$ vertices of the smallest weight
- $max_L = \max_{v \in L} w(v)$
- $w' : (V \setminus L) \to \mathbb{N}$
  - $w'(x) = w(x) - max_L$
- Reduced instance $R = ((T - L), w')$
- R_{approx}: 2-approximate solution for $((T - L), w')$
- **Lemma:** $L \cup R_{approx}$ is a 2-approximate solution for (T, w)

A Generalized Local Ratio Technique

If there is a "large" optimum solution

- Find L, compute w'
- Recursively find a 2-approximate solution R_{approx} for $((T - L), w')$
- Return $L \cup R_{approx}$

What if there is **no** optimum solution with $\geq \frac{2n}{3}$ vertices?

Steroid II: Randomization

- Pick a vertex p uniformly at random
  - "Pivot" vertex
  - There is an optimum solution which does not contain p, with probability $\geq \frac{1}{3}$
  - So there is such a 2-approximate solution as well

What if there is **no** optimum solution with $\geq \frac{2n}{3}$ vertices?

Steroid II: Randomization

- Pick a vertex p uniformly at random
  - With probability $\geq \frac{1}{3}$, $p \notin S$ for a 2-approximate solution S
  - We look for such an S

What if there is **no** optimum solution with $\geq \frac{2n}{3}$ vertices?

Steroid II: Randomization

- Pick a vertex p uniformly at random
  - With probability $\geq \frac{1}{3}$, $p \notin S$ for a 2-approximate solution S
  - We look for such an S
- If p is not part of any directed triangle
  - Recurse on the in- and out- neighbourhoods of p
  - Get 2-approximate solutions S_{in}, S_{out}
  - Return $S = S_{in} \cup S_{out}$

What if there is **no** optimum solution with $\geq \frac{2n}{3}$ vertices?

Steroid II: Randomization

- Pick a vertex p uniformly at random
  - With probability $\geq \frac{1}{3}$, $p \notin S$ for a 2-approximate solution S
  - We look for such an S
- If $p \to x \to y \to p$ is a triangle
  - $\{x,y\} \cap S \neq \emptyset$
  - Apply the Local Ratio Technique to $\{x,y\}$
  - Repeat till p is not in any directed triangle

Tying it all together

- Input: $(T = (V, A)\ ;\ |V| = n, w : V \to \mathbb{N})$
- If T has a smallest-weight FVS with at least $\frac{2n}{3}$ vertices
  - Pick the $\frac{n}{3}$ least-weight vertices L into a 2-approximate solution
  - Delete L from T
  - Adjust the weights of the remaining vertices
  - Recursively find a 2-approximate solution of the resulting instance

Tying it all together

- Input: $(T = (V, A)\ ;\ |V| = n, w : V \to \mathbb{N})$
- If T has a smallest-weight FVS with at least $\frac{2n}{3}$ vertices
  - Do stuff
- If T has no smallest-weight FVS with at least $\frac{2n}{3}$ vertices
  - Pick a "pivot" vertex p uniformly at random
  - While p is part of a directed triangle $\{p, x, y\}$, apply the local ratio technique on $\{x, y\}$
    - This deletes at least one of $\{x, y\}$
  - If p is not in any directed triangle:
    - Recursively find 2-approximate solutions of the in- and out-neighbourhoods of p

Tying it all together

- Input: $(T = (V, A)\ ;\ |V| = n, w : V \to \mathbb{N})$
- If T has a smallest-weight FVS with at least $\frac{2n}{3}$ vertices
  - Do stuff
- If T has no smallest-weight FVS with at least $\frac{2n}{3}$ vertices
  - Do stuff

Tying it all together

- Input: $(T = (V, A)\ ;\ |V| = n, w : V \to \mathbb{N})$
- IF T has a smallest-weight FVS with at least $\frac{2n}{3}$ vertices
  - Do stuff
- If T has no smallest-weight FVS with at least $\frac{2n}{3}$ vertices
  - Do stuff

Tying it all together

Steroid III: Branching + Divide and Conquer

- Input: $(T = (V, A)\ ;\ |V| = n, w : V \to \mathbb{N})$
- If $n \leq 10$ then solve by brute force

Tying it all together

Steroid III: Branching + Divide and Conquer

- Input: $(T = (V, A)\ ;\ |V| = n, w : V \to \mathbb{N})$
- If $n \leq 10$ then solve by brute force
- When $n > 10$:
  - Compute a solution S_0 assuming there is an optimum solution with $\geq \frac{2n}{3}$ vertices

Tying it all together

Steroid III: Branching + Divide and Conquer

- Input: $(T = (V, A)\ ;\ |V| = n, w : V \to \mathbb{N})$
- If $n \leq 10$ then solve by brute force
- When $n > 10$:
  - Compute a solution S_0 assuming there is an optimum solution with $\geq \frac{2n}{3}$ vertices
  - Compute 25 solutions $S_1, \ldots, S_{25}$:
    - Pick a vertex p u.a.r from the set $\{v \in V\ ;\ |N^+(v)| \leq \frac{8n}{9}, |N^-(v)| \leq \frac{8n}{9}\}$
    - Apply the "local" Local Ratio procedure with p as the pivot to get solution S_i
  - Return the minimum-weight set from among $S_0, S_1, \ldots, S_{25}$

Tying it all together

Running time analysis

- Recurrence: $T(n) \leq 51 \cdot T(8n/9) + \mathcal{O}(n^2)$
  - The "large-solution" step recurses on a graph with $\frac{5n}{6} < \frac{8n}{9}$ vertices
  - Each "small-solution" step recurses on two graphs, each with at most $\frac{8n}{9}$ vertices
  - There are 25 "small-solution" steps

Tying it all together

Running time analysis

- Recurrence: $T(n) \leq 51 \cdot T(8n/9) + \mathcal{O}(n^2)$
- Resolves to $T(n) = \mathcal{O}(n^{34})$ by the Master Theorem
  - Let $T(n) = aT(n/b) + f(n)$; $a \geq 1, b > 1$
    - If $f(n) = \mathcal{O}(n^{\log_b a - \varepsilon})$ then $T(n) = \Theta(n^{\log_b a})$
    - $\log_{9/8} 51 \approx 33.382$

Tying it all together

Probability of success

- Claim: The procedure outputs a 2-approximate solution of (T, w) with probability at least half.
- Proof: Induction on the number n of vertices in T
  - If $n \leq 10$: brute force, exact solution

Tying it all together
Probability of success

- Claim: The procedure outputs a 2-approximate solution of (T, w) with probability at least half.
- Proof: Induction on the number n of vertices in T
  - If (T, w) has an optimum solution with at least $\frac{2n}{3}$ vertices:
    - S_0 is a 2-approximate solution with probability at least half.

Tying it all together

Probability of success

- Claim: The procedure outputs a 2-approximate solution of (T, w) with probability at least half.
- Proof: Induction on the number n of vertices in T
  - Say (T, w) has no optimum solution with at least $\frac{2n}{3}$ vertices
  - In computing *each* $S_1, \ldots, S_{25}$, the probability that p is not in an optimum solution is at least $\frac{1}{9}$.
    - There are at least $\frac{n}{9}$ non-solution vertices v with $|N^+(v)| \leq \frac{8n}{9}$ and $|N^-(v)| \leq \frac{8n}{9}$

Tying it all together

Probability of success

- Claim: The procedure outputs a 2-approximate solution of (T, w) with probability at least half.
- Proof: Induction on the number n of vertices in T
  - Say (T, w) has no optimum solution with at least $\frac{2n}{3}$ vertices
  - In computing *each* $S_1, \ldots, S_{25}$, the probability that p is not in an optimum solution is at least $\frac{1}{9}$.
  - Inductively, the two recursive solutions are 2-approximate with probability at least half.
  - Each S_i is good with probability at least $\frac{1}{36}$
  - At least one of the 25 S_is is good with probability at least
    - $1 - (1 - \frac{1}{36})^{25} \geq \frac{1}{2}$

In Conclusion . . .

- **Theorem:** There is a randomized polynomial-time algorithm which, given an instance (T, w) of Weighted Tournament Feedback Vertex Set on n vertices, runs in $\mathcal{O}(n^{34})$ time and outputs a 2-approximate solution with probability at least half.

In Conclusion ...

- **Theorem:** There is a randomized polynomial-time algorithm which, given an instance (T, w) of Weighted Tournament Feedback Vertex Set on n vertices, runs in $\mathcal{O}(n^{34})$ time and outputs a 2-approximate solution with probability at least half.
- Can be derandomized to run in $n^{\mathcal{O}(\log n)}$ time
  - Try each "good" vertex as pivot, instead of picking 25 of them at random
    - $T(n) \leq (2n+1) \cdot T(8n/9) + \mathcal{O}(n^2)$
    - Resolves to $T(n) = n^{\mathcal{O}(\log n)}$

Open Problems

- Deterministic polynomial time algorithm?
- Reasonable degree for the polynomial?
- 2-approximation algorithms for other 3-hitting set problems?
  - E.g: CLUSTER VERTEX DELETION
    - $\frac{9}{4}$-approximation
    - Local Ratio Technique
    - Fiorini et al., August 2018.

Thank You!