Approximation algorithms for prize-collecting forest problems with submodular penalty functions

Yogeshwer Sharma (Cornell University)
Chaitanya Swamy (University of Waterloo)
David P. Williamson (Cornell University)

January 09, 2006 (Tuesday)

Overview of the talk

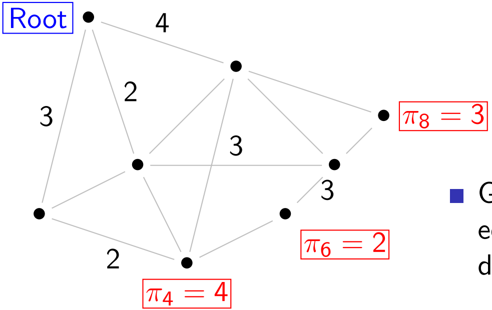
1. Getting familiar with the problem
2. A primal-dual approach
3. An LP-rounding approach

Problems in context: PCST
[Bienstock, Goemans, Simchi-Levi, and Williamson]

Prize-collecting
Steiner tree

Prize-collecting Steiner tree problem (PCST)

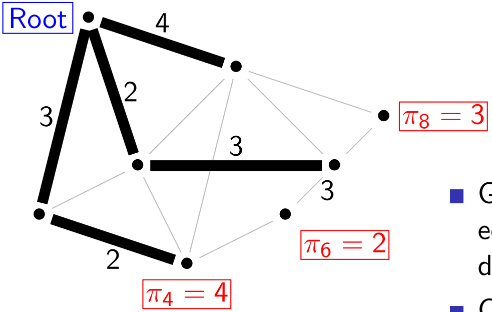
[Bienstock, Goemans, Simchi-Levi, and Williamson]



(Not all penalties are shown)

- Input:
  - A graph,
  - weights on edges,
  - a root r, and
  - penalties π on vertices for not connecting to root.
- Goal: Minimize the cost of edges + sum of penaltes of disconnected vertices.

Prize-collecting Steiner tree problem (PCST)
[Bienstock, Goemans, Simchi-Levi, and Williamson]

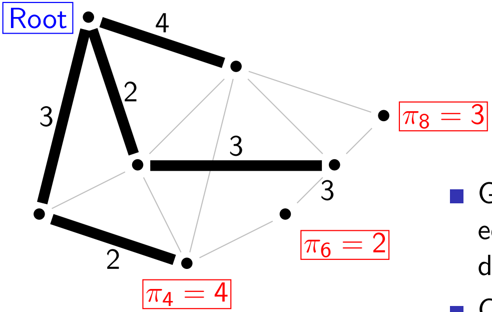


(Not all penalties are shown)

- Input:
  - A graph,
  - weights on edges,
  - a root r, and
  - penalties π on vertices for not connecting to root.
- Goal: Minimize the cost of edges + sum of penaltes of disconnected vertices.
- Cost of the solution is $(4+2+3+3+2)+(2+3) = 19$.

Prize-collecting Steiner tree problem (PCST)

[Bienstock, Goemans, Simchi-Levi, and Williamson]



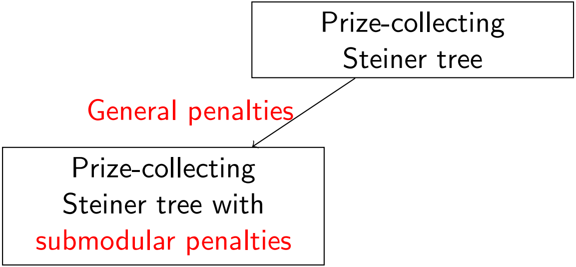
(Not all penalties are shown)

- Input:
  - A graph,
  - weights on edges,
  - a root r, and
  - penalties π on vertices for not connecting to root.
- Goal: Minimize the cost of edges + sum of penaltes of disconnected vertices.
- Cost of the solution is $(4+2+3+3+2)+(2+3) = 19$.

Known results

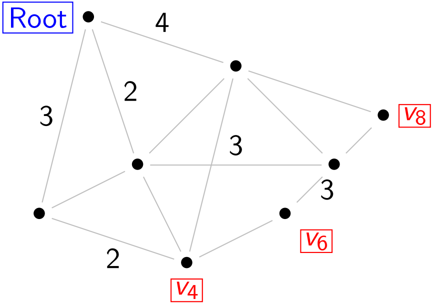
A primal-dual 2-approximation by [Goemans and Williamson].

Problems in context: PCST with submodular penalties
[Hayrapetyan, Swamy, Tardos]



PCST with submodular penalties
[Hayrapetyan, Swamy, Tardos]

$\pi(A) + \pi(B) \geq \pi(A \cup B) + \pi(A \cap B)$



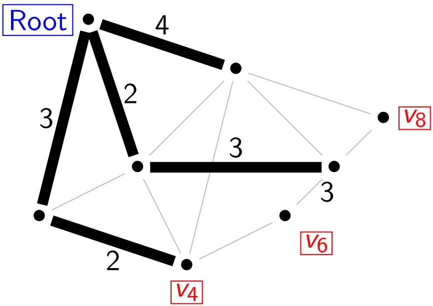
$\pi(\{v_6, v_8\}) = 5; \quad \pi(\{v_4, v_6\}) = 7$
(Not all penalties are shown)

- Input:
  - A graph,
  - weights on edges,
  - a root r, and
  - submodular penalty function $\pi(\cdot)$ on set of disconnected vertices.
- Goal: Minimize the cost of edges + penalty on disconnected vertices.

PCST with submodular penalties

[Hayrapetyan, Swamy, Tardos]

$$\pi(A) + \pi(B) \geq \pi(A \cup B) + \pi(A \cap B)$$



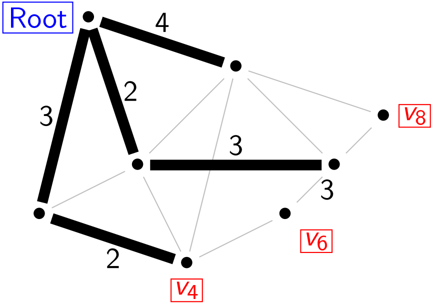
$\pi(\{v_6, v_8\}) = 5; \quad \pi(\{v_4, v_6\}) = 7$

(Not all penalties are shown)

- Input:
  - A graph,
  - weights on edges,
  - a root r, and
  - submodular penalty function $\pi(\cdot)$ on set of disconnected vertices.
- Goal: Minimize the cost of edges + penalty on disconnected vertices.
- Cost of the solution is $(4+2+3+3+2)+(5) = 19$.

PCST with submodular penalties
[Hayrapetyan, Swamy, Tardos]

$\pi(A) + \pi(B) \geq \pi(A \cup B) + \pi(A \cap B)$



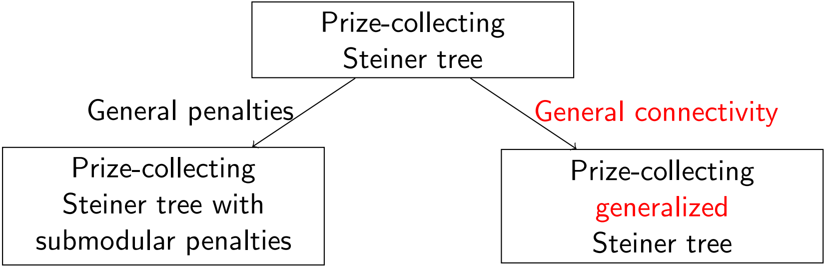
$\pi(\{v_6, v_8\}) = 5; \quad \pi(\{v_4, v_6\}) = 7$
(Not all penalties are shown)

- Input:
  - A graph,
  - weights on edges,
  - a root r, and
  - submodular penalty function $\pi(\cdot)$ on set of disconnected vertices.
- Goal: Minimize the cost of edges + penalty on disconnected vertices.
- Cost of the solution is $(4+2+3+3+2)+(5) = 19$.

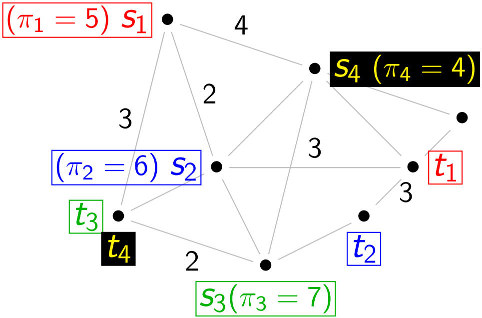
Known results

A primal-dual 2-approximation by [Hayrapetyan, Swamy, Tardos].

Problems in context: PCGST
[Hajiaghayi and Jain]



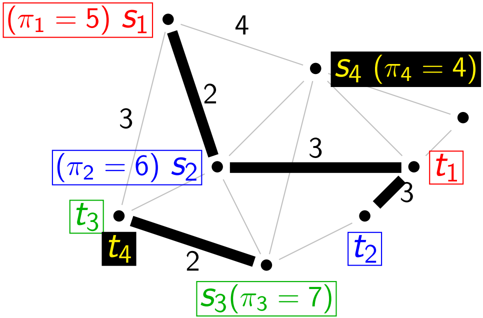
Prize-collecting generalized Steiner tree problem (PCGST)
[Hajiaghayi and Jain]



- Input:
  - A graph,
  - weights on edges,
  - k pairs to connect, and
  - penalty for each pair for not connecting it.
- Goal: Minimize the cost of edges + sum of penalties of disconnected pairs.

Prize-collecting generalized Steiner tree problem (PCGST)

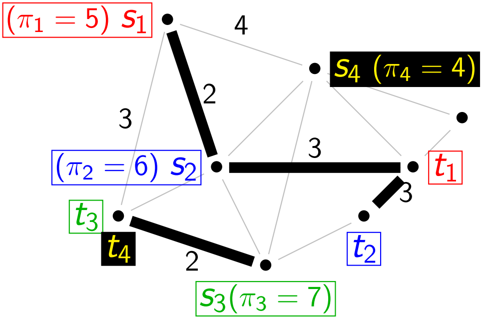
[Hajiaghayi and Jain]



- Input:
  - A graph,
  - weights on edges,
  - k pairs to connect, and
  - penalty for each pair for not connecting it.
- Goal: Minimize the cost of edges + sum of penalties of disconnected pairs.
- Total cost is equal to $(2 + 3 + 3 + 2) + (4) = 14$.

Prize-collecting generalized Steiner tree problem (PCGST)

[Hajiaghayi and Jain]

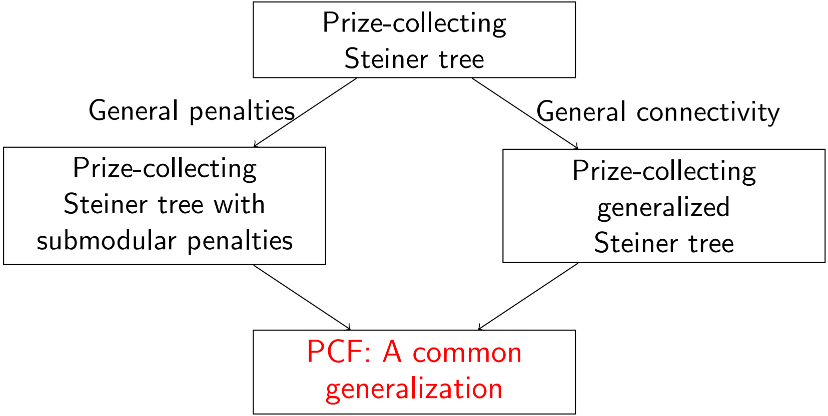


- Input:
  - A graph,
  - weights on edges,
  - k pairs to connect, and
  - penalty for each pair for not connecting it.
- Goal: Minimize the cost of edges + sum of penalties of disconnected pairs.
- Total cost is equal to $(2 + 3 + 3 + 2) + (4) = 14$.

Known results

A primal-dual 3-approximation, and an LP-rounding 2.54-approximation by [Hajiaghayi and Jain].

Problems in context: Prize-collecting forest problem

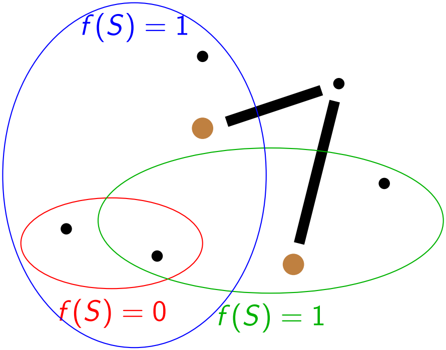


Prize collecting forest problem: Generalization in two dimensions I

Prize-collecting forest problem has two components.

1. Connectivity requirement
   - PCST $\stackrel{\text{Generalize}}{\longrightarrow}$ PCGST $\stackrel{\text{Generalize}}{\longrightarrow}$ PCF.

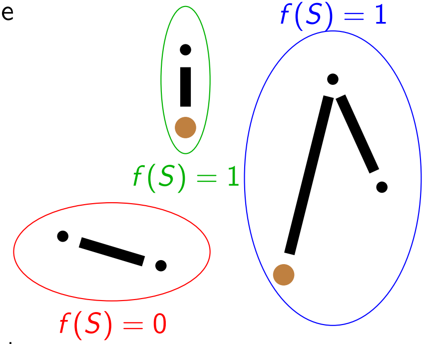
- Generalization to arbitrary functions: $f : 2^V \rightarrow \{0, 1\}$.
  - $f(S) = 1$ means at least one edge is required "leaving" S.



Prize collecting forest problem: Generalization in two dimensions II

2. Penalty
   - PCST $\xrightarrow{\text{Generalize}}$ PCST with submodular penalties $\xrightarrow{\text{Generalize}}$ PCF.
   - Generalize to wider class of penalty: $\pi : 2^{2^V} \to \mathbb{N}_+$.

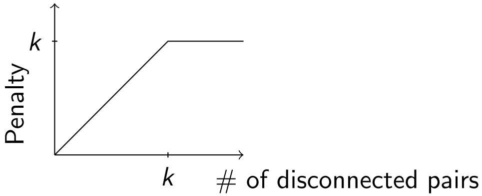
- Given a solution $F \subseteq E$, let $\mathcal{F}$ be the collection of all violated subsets.
  - Violated subset: S such that $f(S) = 1$ but there is no edge leaving them ($\delta(S) \cap F = \emptyset$).
- Then, penalty of the solution is $\pi(\mathcal{F})$.
  - $\pi(\cdot)$ should be submodular [on families of subsets].
  - Need other *natural* requirements.



$\pi(\{Green, Blue\}) = 8$ (example)

How general is this generalization?

- It captures
  - Prize-collecting Steiner tree with submodular penalties.
  - Prize-collecting generalized Steiner tree.
- Penalty of a pair could depend on already disconnected pairs.
  - Submodular penalty function on set of pairs.
  - Penalty function with a cap.



Our results

- Our results
  - A primal-dual 3-approximation.
  - A $\frac{1}{1-e^{-1/2}} \approx 2.54$ approximation via LP-rounding.

Our results

- Our results
  - A primal-dual 3-approximation.
  - A $\frac{1}{1-e^{-1/2}} \approx 2.54$ approximation via LP-rounding.
- Ideas from [Goemans-Williamson] and [Hajiaghayi-Jain].
  - Contribution: A tractable generalization of two earlier problems.
  - Main issue: Handle the doubly exponential-size LP.
  - Primal-dual algorithm implementation
    - Submodular function minimization and
    - rational approximation.
  - LP-rounding approach
    - There is a short solution.
    - Solve the LP by a novel convex programming formulation.

A primal-dual algorithm

An LP formulation (Primal program)
[Hajiaghayi and Jain]

- Natural LP.
- Variables for edges: x_e.
- Variables for families: $z_{\mathcal{F}}$.

$$\text{Min} \quad \sum_e c_e x_e + \sum_{\mathcal{F}} \pi(\mathcal{F}) z_{\mathcal{F}} \qquad \text{(PCF-LP)}$$

$$\text{s.t.} \quad \underbrace{\sum_{e \in \delta(S)} x_e}_{\text{Edge contribution}} + \underbrace{\sum_{\mathcal{F}: S \in \mathcal{F}} z_{\mathcal{F}}}_{\text{Family contribution}} \geq f(S) \qquad \text{for all } S \subseteq V,$$

$$x_e, z_{\mathcal{F}} \geq 0 \qquad \text{for all } e, \mathcal{F}.$$

An LP formulation (Primal program)
[Hajiaghayi and Jain]

- Natural LP.
- Variables for edges: $\boxed{x_e}$.
- Variables for families: $z_{\mathcal{F}}$.

$$\begin{aligned}
\text{Min} \quad & \sum_e c_e \boxed{x_e} + \sum_{\mathcal{F}} \pi(\mathcal{F}) z_{\mathcal{F}} && \text{(PCF-LP)} \\
\text{s.t.} \quad & \underbrace{\sum_{e \in \delta(S)} x_e}_{\text{Edge contribution}} + \underbrace{\sum_{\mathcal{F}: S \in \mathcal{F}} z_{\mathcal{F}}}_{\text{Family contribution}} \geq f(S) && \text{for all } S \subseteq V, \\
& x_e, z_{\mathcal{F}} \geq 0 && \text{for all } e, \mathcal{F}.
\end{aligned}$$

An LP formulation (Primal program)
[Hajiaghayi and Jain]

- Natural LP.
- Variables for edges: x_e.
- Variables for families: $\boxed{z_{\mathcal{F}}}$.

$$
\begin{array}{llllll}
\text{Min} & \sum_{e} c_e x_e & + \sum_{\mathcal{F}} \pi(\mathcal{F}) \boxed{z_{\mathcal{F}}} & & & \text{(PCF-LP)} \\
\text{s.t.} & \underbrace{\sum_{e \in \delta(S)} x_e}_{\text{Edge contribution}} & + \underbrace{\sum_{\mathcal{F}: S \in \mathcal{F}} z_{\mathcal{F}}}_{\text{Family contribution}} & \geq & f(S) & \text{for all } S \subseteq V, \\
 & & x_e, z_{\mathcal{F}} & \geq & 0 & \text{for all } e, \mathcal{F}.
\end{array}
$$

An LP formulation (Primal program)
[Hajiaghayi and Jain]

- Natural LP.
- Variables for edges: x_e.
- Variables for families: $z_{\mathcal{F}}$.

$$\text{Min} \quad \sum_e c_e x_e + \sum_{\mathcal{F}} \pi(\mathcal{F}) z_{\mathcal{F}} \qquad \text{(PCF-LP)}$$

$$\text{s.t.} \quad \underbrace{\sum_{e \in \delta(S)} x_e}_{\text{Edge contribution}} + \underbrace{\sum_{\mathcal{F}: S \in \mathcal{F}} z_{\mathcal{F}}}_{\text{Family contribution}} \geq f(S) \qquad \text{for all } \boxed{S \subseteq V},$$

$$x_e, z_{\mathcal{F}} \geq 0 \qquad \text{for all } e, \mathcal{F}.$$

An LP formulation (Dual program)

- Variables for subsets: y_S.

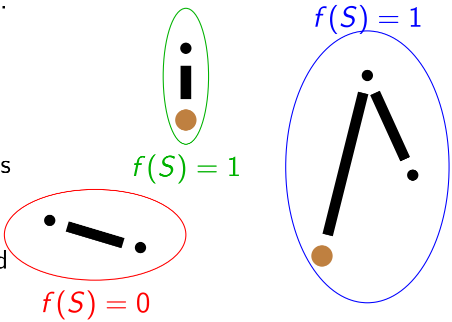
$$\begin{aligned}
\text{Max} \quad & \sum_{S \subseteq V} f(S) \cdot y_S && && \text{(PCF-D)} \\
\text{s.t.} \quad & \sum_{S : e \in \delta(S)} y_S \leq c_e && \text{for all } e \in E && \text{(type-(edge))} \\
& \sum_{S : S \in \mathcal{F}} y_S \leq \pi(\mathcal{F}) && \text{for all } \mathcal{F} \subseteq 2^V && \text{(type-(family))} \\
& y_S \geq 0 && \text{for all } S \subseteq V.
\end{aligned}$$

The primal dual algorithm
[Goemans and Williamson] and [Hajiaghayi and Jain]

$$\sum_{S:e\in\delta(S)} y_S \le c_e \qquad \text{for all } e \in E \qquad \text{(type-(edge))}$$

$$\sum_{S:S\in\mathcal{F}} y_S \le \pi(\mathcal{F}) \qquad \text{for all } \mathcal{F} \subseteq 2^V \qquad \text{(type-(family))}$$

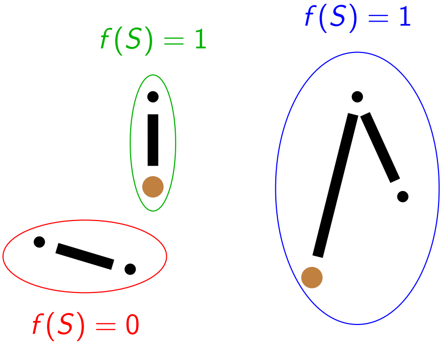
- n components for n vertices.
    1. Active component (No edge leaving it, but need one).
    2. Inactive component (Does not need an edge leaving it).
    3. Marked component (Need an edge leaving it, but instead pay the penalty).



The primal dual algorithm
[Goemans and Williamson] and [Hajiaghayi and Jain]

$$\sum_{S:e\in\delta(S)} y_S \leq c_e \quad \text{for all } e \in E \qquad \text{(type-(edge))}$$

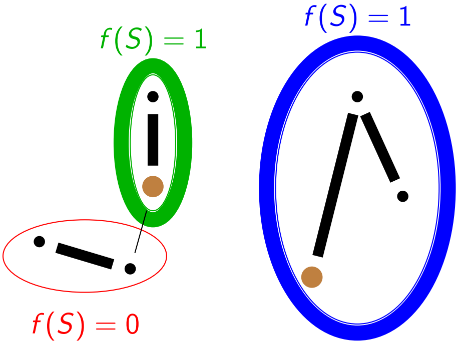
- Increase dual for active components (uniformly), until one of the following events happens.
- An edge goes tight.
  - Merge the two components.



The primal dual algorithm
[Goemans and Williamson] and [Hajiaghayi and Jain]

$$\sum_{S: e \in \delta(S)} y_S \leq c_e \quad \text{for all } e \in E \qquad \text{(type-(edge))}$$

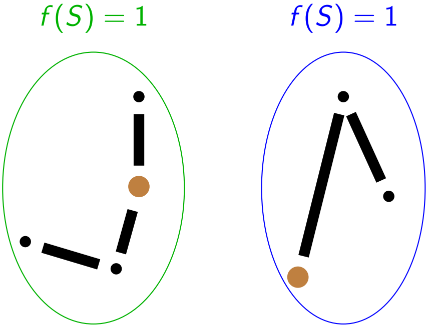
- Increase dual for active components (uniformly), until one of the following events happens.
- An edge goes tight.
  - Merge the two components.



The primal dual algorithm
[Goemans and Williamson] and [Hajiaghayi and Jain]

$$\sum_{S: e \in \delta(S)} y_S \leq c_e \quad \text{for all } e \in E \qquad \text{(type-(edge))}$$

- Increase dual for active components (uniformly), until one of the following events happens.
- An edge goes tight.
  - Merge the two components.



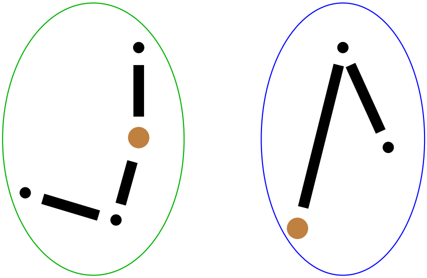
The primal dual algorithm

[Goemans and Williamson] and [Hajiaghayi and Jain]

$$\sum_{S:S\in\mathcal{F}} y_S \leq \pi(\mathcal{F}) \text{ for all } \mathcal{F} \subseteq 2^V \qquad \text{(type-(family))}$$

$\pi(\{Green, Blue\}) = 8$

$y_S = 3$ $\qquad$ $y_S = 4$



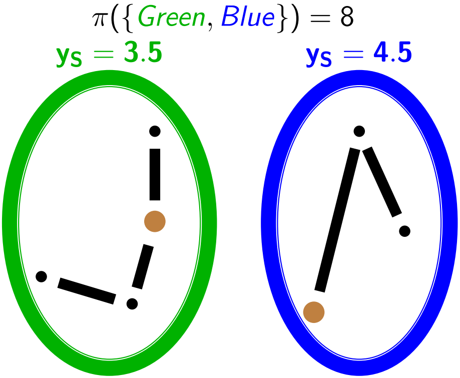
- A family goes tight.
  - Stop increasing the dual for all participating sets, and mark all of them.
  - The value of dual is equal to the penalty of this family (tightness condition).

The primal dual algorithm

[Goemans and Williamson] and [Hajiaghayi and Jain]

$$\sum_{S: S \in \mathcal{F}} y_S \leq \pi(\mathcal{F}) \text{ for all } \mathcal{F} \subseteq 2^V \qquad \text{(type-(family))}$$

- A family goes tight.
  - Stop increasing the dual for all participating sets, and mark all of them.
  - The value of dual is equal to the penalty of this family (tightness condition).



The primal dual algorithm

[Goemans and Williamson] and [Hajiaghayi and Jain]

$$\sum_{S:e\in\delta(S)} y_S \leq c_e \qquad \text{for all } e \in E \qquad \text{(type-(edge))}$$

$$\sum_{S:S\in\mathcal{F}} y_S \leq \pi(\mathcal{F}) \qquad \text{for all } \mathcal{F} \subseteq 2^V \qquad \text{(type-(family))}$$

- Stop when no active component remains.
- Penalty for all marked subsets has been paid for.
- Reverse delete step
  1. Consider edge in the reverse order of what they were added to the solution.
  2. Consider deleting an edge. Has the penalty of two resulting components already been paid for? If yes, delete the edge, else retain it.

The idea behind the proof

- The penalty paid by the algorithm is at most OPT.
  - The family of all marked sets is tight (Union of two tight families is also tight).
  - Penalty of the solution is at most

$$\pi(\text{Marked Sets}) = \sum_{S \in \text{Marked}} y_S \leq \text{OPT}.$$

- The cost of edges is bounded by twice the dual objective.
  - The idea here is to use the fact that the average degree of a tree is at most 2.
  - This follows the proof in [Goemans and Williamson '95], with some care because of the more general situation.

Implementation: Finding a tight family

- A family is a collection of subsets of V.
- There are doubly exponentially many families.
- Which family goes tight next (in the dual increase phase)?
  - Reduce the number of candidate families to $2^{O(n)}$ (from 2^{2^n}) and use submodular function minimization.

Claim

If a set S has $\hat{y}_S = 0$ and is not active, then it will not be in the next tight family.

Finding a tight family

- There are only $3n$ sets which
  - have positive dual or
  - are active.
- We only need to consider families which contain just these sets.

Maximize ε

subject to

$$\sum_{S \in \mathcal{F}} \hat{y}_S + \varepsilon \cdot (\text{Number of active sets in } \mathcal{F}) \leq \pi(\mathcal{F}); \qquad \text{for all } \mathcal{F}.$$

$\mathcal{F}$ here is a subfamily of subsets which either have positive dual or are active.

- This is still an exponential-size LP.

How to solve the program?

Maximize ε

subject to

$$\sum_{S \in \mathcal{F}} \hat{y}_S + \varepsilon \cdot (\text{Number of active sets in } \mathcal{F}) \leq \pi(\mathcal{F}); \qquad \text{for all } \mathcal{F}.$$

- Use binary search to find the correct value of ε.
- The interval can be halved using submodular function minimization.
  - Ground elements: active subsets and subsets having positive dual.
  - Once the interval becomes sufficiently small, use rational approximation to find the optimal value.

An LP-rounding approach

- Challenges.
  - What do we mean by solving the linear program? There are 2^n constraints and 2^{2^n} variables.
  - [Hajiaghayi and Jain] had an underlying compact formulation.

An LP-rounding approach

- Challenges.
  - What do we mean by solving the linear program? There are 2^n constraints and 2^{2^n} variables.
  - [Hajiaghayi and Jain] had an underlying compact formulation.
- A way out
  - There exists a solution with very short description.

An LP-rounding approach

- Challenges.
  - What do we mean by solving the linear program? There are 2^n constraints and 2^{2^n} variables.
  - [Hajiaghayi and Jain] had an underlying compact formulation.
- A way out
  - There exists a solution with very short description.
- Idea
  - Given the edge values, the optimum values corresponding to families can be computed by a convex program.
  - Compact, convex programming formulation of the problem (in x_e variables).

An LP-rounding approach

- Challenges.
  - What do we mean by solving the linear program? There are 2^n constraints and 2^{2^n} variables.
  - [Hajiaghayi and Jain] had an underlying compact formulation.
- A way out
  - There exists a solution with very short description.
- Idea
  - Given the edge values, the optimum values corresponding to families can be computed by a convex program.
  - Compact, convex programming formulation of the problem (in x_e variables).
- Rounding process similar to the one in [Goemans].

An LP-rounding approach

- Challenges.
  - What do we mean by solving the linear program? There are 2^n constraints and 2^{2^n} variables.
  - [Hajiaghayi and Jain] had an underlying compact formulation.
- A way out
  - There exists a solution with very short description.
- Idea
  - Given the edge values, the optimum values corresponding to families can be computed by a convex program.
  - Compact, convex programming formulation of the problem (in x_e variables).
- Rounding process similar to the one in [Goemans].
- Details in the paper. . .

Future directions

- Are there other problems that fit our framework?
- Is there a 2-approximation for PCF?
  - [Hayrapetyan, Swamy, Tardos] had a 2-approximation for a special case.
- Is there a general enough subclass of PCF which admits a 2-approximation?

Thank You for your time and attention

Comments/Questions?