Approximation Algorithms for Robust Optimization and Max-Min Optimization

Anupam Gupta

Carnegie Mellon University

with Viswanath Nagarajan (IBM Watson) and R.Ravi (CMU)

robust problems

Input: covering problem,

collection of "demand scenarios" K_1, K_2,…, K_N

Output:

build "first-stage" and "second-stage$_t$" partial solutions

such that first-stage + second-stage$_t$ satisfies K_t

Objective:

cost = first-stage-cost + λ (max$_t$ second-stage-cost$_t$)

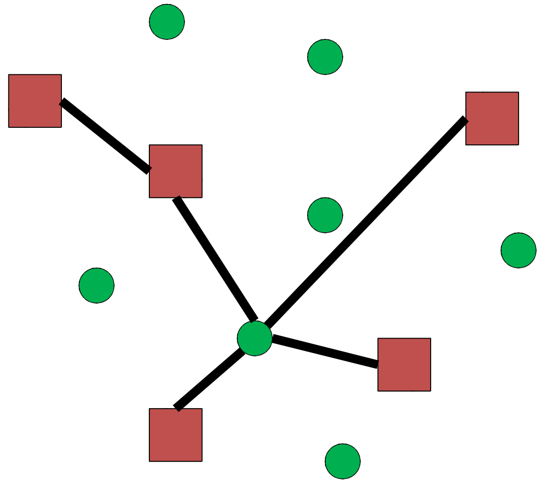
Steiner tree

Given a metric space (V, ℓ)

and a subset K of k terminals

find the least cost network

connecting K



Results:

1.39-approximation [Byrka Grandoni Rothvoss Sanita '10]

APX-hard [Bern Plassmann '89]

robust Steiner tree

Given a metric space (V, ℓ)

a collection of subsets $K_1, K_2, \ldots, K_N$

and a parameter λ

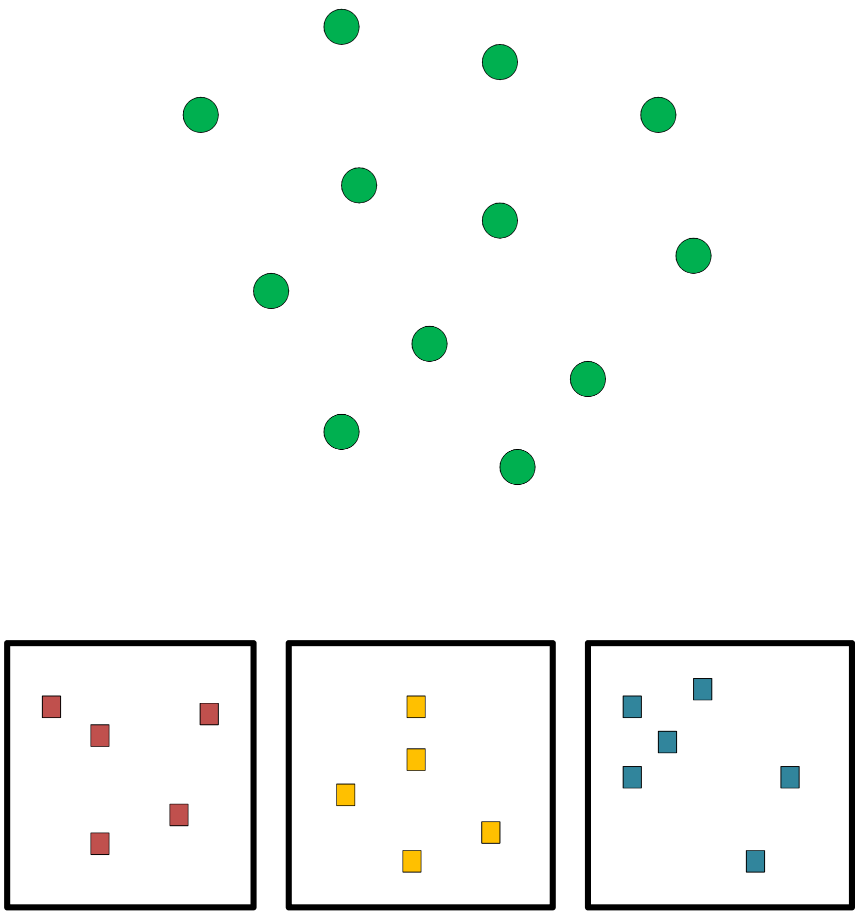
find a set of edges E_0

and edge sets $E_1, E_2, \ldots, E_N$

such that

$E_0 \cup E_t$ connects K_t

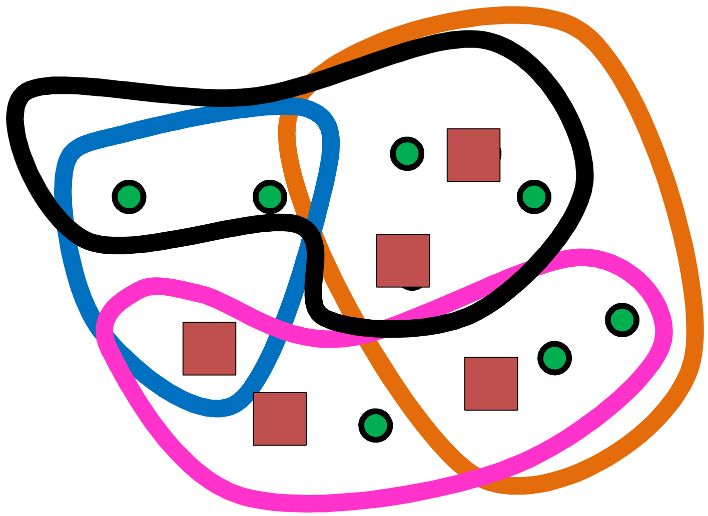
cost = $\ell(E_0) + \max_t \lambda\, \ell(E_t)$



set cover

Given a set system $(U, \mathcal{F})$

and a subset K of k terminals



find the least cost collection

of sets covering K

Results:

(ln n)-factor approximable [Chvatal, Johnson, Lovasz]

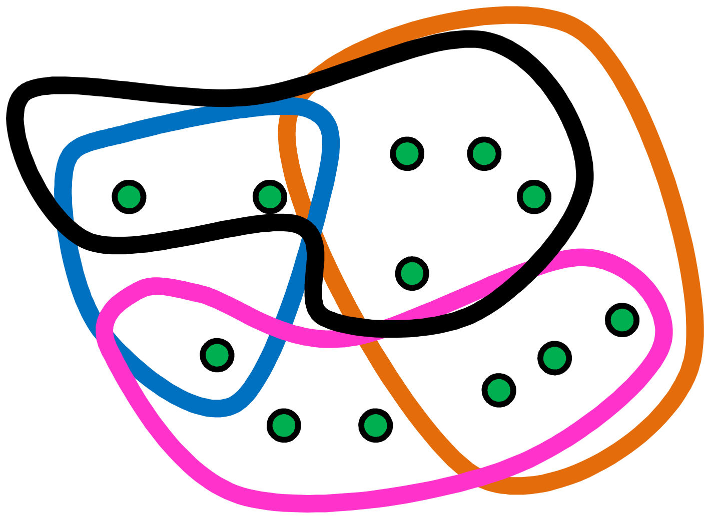
$(1-\epsilon)$(ln n)-hard [Feige '98]

robust set cover

Given a set system $(U, \mathcal{F})$

a family of subsets K_1, K_2,…, K_N

and a parameter λ

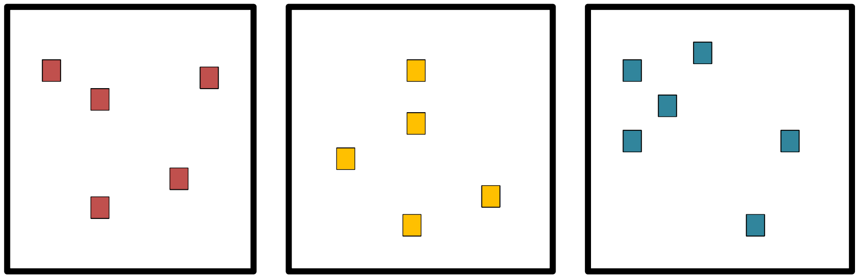


find a collection of sets $\mathcal{F}_0$

and collections $\mathcal{F}_1$, $\mathcal{F}_2$, …, $\mathcal{F}_N$

such that

$\mathcal{F}_0 \cup \mathcal{F}_t$ covers K_t



cost = $c(\mathcal{F}_0) + \max_t \lambda\, c(\mathcal{F}_t)$

robust problems

Input: covering problem,

collection of "demand scenarios" K_1, K_2,..., K_N

Output:

build "first-stage" and "second-stage$_t$" partial solutions

such that first-stage + second-stage$_t$ satisfies K_t

Objective:

$$\text{cost} = \text{first-stage-cost} + \lambda\,(\max_t \text{second-stage-cost}_t)$$

known results (1)

When the N sets K_t are given explicitly:

O(1)-factor approximation for Steiner tree

O(log n)-factor for set cover

[Dhamdhere Goyal Ravi Singh ’05]

known results (2)

What if we want to handle an exponential number of sets?

Case studied: the K_t's are all sets of size at most k

O(log m log n)-approximation for robust set cover

[Feige Jain Mahdian Mirrokni 07]

(used Ellipsoid to reduce to max-min set cover:
"which subset of size k is most difficult to cover?")

O(1)-approximation for robust Steiner tree

[Khandekar Kortsarz Mirrokni Salavatipour 08]

(combinatorial solution)

our results (1)

When scenarios K_t are <u>all</u> sets of size at most k,

$O(\log m + \log n)$-approximation for robust set cover

$O(1)$-approximation for robust Steiner tree/forest

$O(1)$-approximation for robust min-cut

$O(\log^2 n)$-approximation for robust multicut

all using the “same” simple algorithm…

what's the algorithm?

Recall: want to minimize

$$\text{cost} = \text{first-stage-cost} + \lambda\ (\max_t \text{second-stage-cost}_t)$$

call this T*, guess it

Generic Algorithm:

1. If something costs "much more" than T*/k to satisfy today, add it to a set X.
2. First stage: buy an anticipatory solution on X.
3. Second stage: do what you need to do.

our results (1)

When scenarios K_t are all sets of size at most k,

$O(\log m + \log n)$-approximation for robust set cover

$O(1)$-approximation for robust Steiner tree/forest

$O(1)$-approximation for robust min-cut

$O(\log^2 n)$-approximation for robust multicut

all using this simple algorithm…

our results imply similar approximations for the max-min problem
"which subset of size k is most difficult to cover?"

our results (2)

Would like results when scenarios are integer points of some down-monotone polytope.

E.g., when scenarios K_t are <u>all</u> independent sets of some matroid:

Theorem:

if you can solve some covering problem offline

and you can solve the problem online

$\Rightarrow$ you can solve the matroid robust version of the problem.

Usually offline × online approximation.

for other results, see paper…

rest of the talk

Generic Algorithm:

1. If something costs “much more” than T*/k to satisfy today,
   add it to a set X.

2. First stage: buy an anticipatory solution on X.

3. Second stage: do what you need to do.

Show how generic algorithm applies to:

a) robust Steiner tree

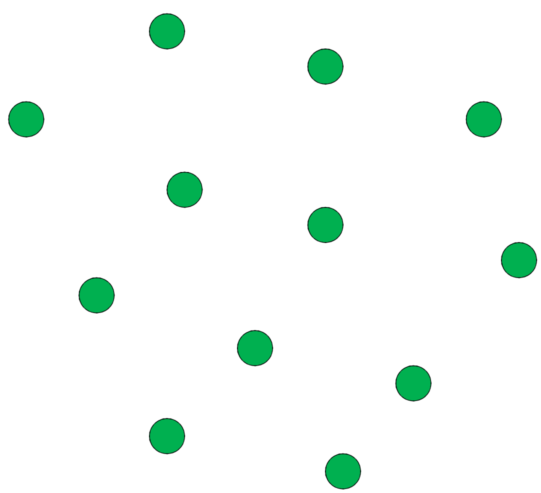
b) robust set cover (sketch)

robust Steiner tree

Given a metric space (V, ℓ)

all possible k-subsets $K_1, K_2, \ldots, K_N$ of V

and a parameter λ



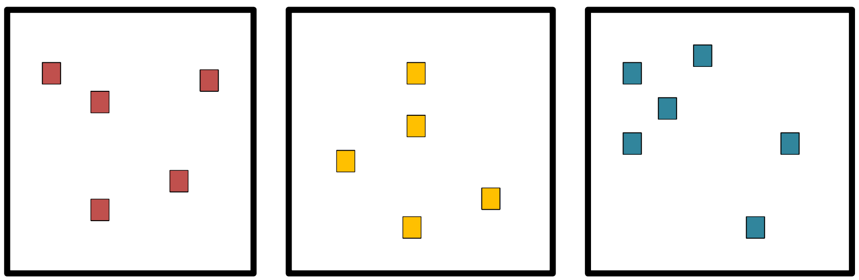
find a set of edges E_0

and edge sets $E_1, E_2, \ldots, E_N$

such that

$E_0 \cup E_t$ connects K_t

cost = $\ell(E_0) + \max_t \lambda\, \ell(E_t)$



the precise algorithm

Algorithm:

1. add an arbitrary vertex to **X**.
2. while exists a vertex v such that distance(v, **X**) > 4 T*/k
   add v to **X**
3. first-stage = Steiner tree on **X**

Fact: cost of connecting any set of k guys in second stage ≤ 4λ T*

Theorem: cost of Steiner-tree(**X**) ≤ O(OPT)

analysis

OPT = first-stage-cost + λ (max$_t$ second-stage-cost$_t$)

call this Φ^* — call this T*, guess it

Theorem: cost of Steiner-tree(**X**) ≤ O(Φ^*)

$$|X| \times 4T^*/k \times ½ \quad \leq \quad \text{optimal tree on } X \quad \leq \quad \Phi^* + |X|/k \times T^*$$

analysis

$$\text{OPT} = \underbrace{\text{first-stage-cost}}_{\text{call this } \Phi^*} + \lambda \underbrace{(\max_t \text{second-stage-cost}_t)}_{\text{call this } T^*\text{, guess it}}$$

Theorem: cost of Steiner-tree(**X**) ≤ O(Φ^*)

$$|\mathbf{X}| \times 4T^*/k \times \tfrac{1}{2} \quad \leq \quad \begin{array}{c}\text{optimal tree}\\ \text{on } \mathbf{X}\end{array} \quad \leq \quad \Phi^* + |\mathbf{X}|/k \times T^*$$

$$\Rightarrow |\mathbf{X}| \times T^*/k \leq \Phi^*$$

$$\Rightarrow \text{our cost of Steiner-tree}(\mathbf{X}) \leq \rho_{ST} \times 2\Phi^*$$

wrapping up Steiner tree

Algorithm:

1. add an arbitrary vertex to **X**.
2. while exists a vertex v such that distance(v, **X**) > 4 T*/k
   add v to **X**
3. first-stage = Steiner tree on **X**

Fact: cost of connecting any set of k guys in second stage ≤ 4λ T*

Theorem: cost of Steiner-tree(**X**) ≤ O(Φ*)

$\Rightarrow$ our total cost ≤ O(Φ* + λ T*)

set cover algorithm

Algorithm:

1. for each element e such that
   cheapest set covering e costs > $O(\log m)\ T^*/k$
   add v to **X**
2. first-stage = set cover for **X**

Fact: cost of covering any k-subset in second stage ≤ $\lambda\ O(\log m)\ T^*$

Theorem: cost of set cover(**X**) ≤ $O(\log n)\ (\Phi^* + T^*)$

$\Rightarrow$ our total cost ≤ $O(\log m + \log n)\ (\Phi^* + \lambda\ T^*)$

sketch of the proof (1)

Theorem: cost of set cover(**X**) ≤ O(log n) (Φ^* + T*)

⇑

Let Y ⊂ X be covered by OPT's first stage
(costs O(log n) Φ^* to cover these)

Z = **X** \ Y

Theorem: cost of set cover(**Z**) ≤ O(log n) T*

⇑

Theorem: $LP_{\text{set cover}}$(**Z**) ≤ O(T*)

sketch of the proof (2)

Theorem: $LP_{\text{set cover}}(\mathbf{Z}) \leq O(T^*)$

Proof strategy:

1. For sake of contradiction, assume $LP_{\text{set cover}}(\mathbf{Z}) > O(T^*)$
2. Cost of any set covering elements from **Z** > $O(\log m)\, T^*/k$
3. Every k-subset of **Z** can be covered at cost ≤ T^*

Gives us a contradiction.

sketch of the proof (3)

1. $LP_{\text{set cover}}(Z) > O(T^*)$
2. Cost of any set covering elements from Z > O(log m) T*/k
3. Every k-subset of Z can be covered at cost ≤ T*

$\min \sum_S c_S x_S$

$\sum_{S \ni e} x_S \geq 1 \quad$ for e in **Z**

$\max \sum_{e \in Z} y_e$

$\sum_{e \in S \cap Z} y_e \leq c_S$

LP-value > 20 T*

Let's scale down all the costs by factor 4T*/k

sketch of the proof (4)

1. $LP_{\text{set cover}}(Z) > O(T^*)$ O(k)
2. Cost of any set covering elements from Z > O(log m) T*/k
3. Every k-subset of Z can be covered at cost ≤ T* k/4

$$\min \sum_S \hat{c}_S x_S$$

$$\sum_{S \ni e} x_S \geq 1 \quad \text{for } e \text{ in } Z$$

$$\max \sum_{e \in Z} y_e$$

$$\sum_{e \in S \cap Z} y_e \leq \hat{c}_S$$ ← large!

LP-value > 20 T* 5k

What if the solution y_e was integral?

Then set containing k of the e's with highest y_e values gives a dual with value > k.

Contradicts the fact that it has small set cover!

sketch of the proof (4)

1. $LP_{\text{set cover}}(Z) > O(T^*)$ O(k)
2. $\hat{C}$ost of any set covering elements from Z > O(log m) T*/k
3. Every k-subset of Z can be covered at $\hat{c}$ost ≤ T* k/4

$$\min \sum_S \hat{c}_S x_S$$

$$\sum_{S \text{ ni } e} x_S \geq 1 \quad \text{for } e \text{ in } Z$$

$$\max \sum_{e \text{ in } Z} y_e$$

$$\sum_{e \text{ in } S \cap Z} y_e \leq \hat{c}_S$$ ← large!

LP-value > 20 T* 5K

What if the solution y_e was integral?

If not, randomized rounding does not lose much, since costs are large!

to conclude

algorithms for robust problems with exponentially many scenarios

- 4.5-approximation for Steiner tree
- O(log m + log n)-approximation for Set Cover

FJMM showed such dependence on number of sets m necessary…

- also for Steiner forest, min-cut, multicut
- simple algorithm
- "dual-rounding" analysis

directly gives algorithms for max-min problems

- "which subset of k elements is costliest to cover?"

paper's on the arxiv

thanks!