Approximation Algorithms for Stochastic Combinatorial Optimization

Part I: Two stage problems

Anupam Gupta
Carnegie Mellon University

stochastic optimization

Question: How to model uncertainty in the inputs?

- data may not yet be available
- obtaining exact data is difficult/expensive/time-consuming

Goal: make (near)-optimal decisions given some predictions (probability distribution on potential inputs).

Studied since the 1950s, and for good reason: many practical applications…

Approximation Algorithms

Recent development of **approximation algorithms** for NP-hard stochastic optimization problems.

I will give an overview of some of the results/ideas in the talks today and tomorrow.

models with recourse

The problem instance is revealed in "stages"

- initially we perform some anticipatory actions
- at each stage, more information released
- we may take some more recourse actions at this point

Initially, given "guesses" about final problem instance

(i.e., given probability distribution π over problem instances)

Want to minimize:

$$\text{Cost(Initial actions)} + E_{\pi} [\text{ cost of recourse actions }]$$

the Steiner tree problem

Input: a metric space

a root vertex r

a subset R of terminals

Output: a tree T connecting R to r

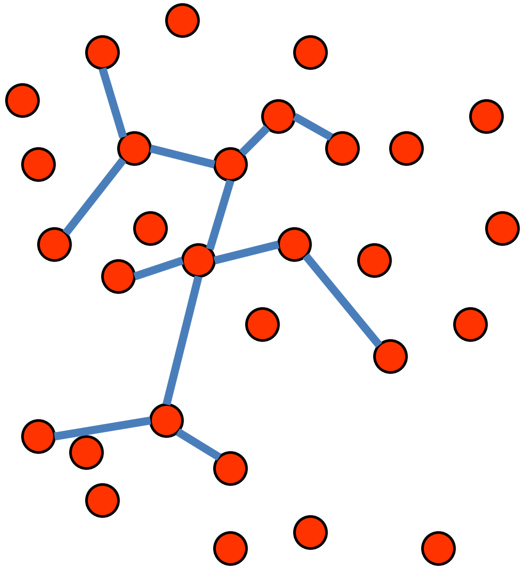
of minimum length/cost.

Facts: NP-hard

MST is a 2-approximation

$\text{cost}(\text{MST}(R \cup r)) \leq 2\ \text{OPT}(R)$

[Robins Zelikovsky '99] gave a 1.55-approximation



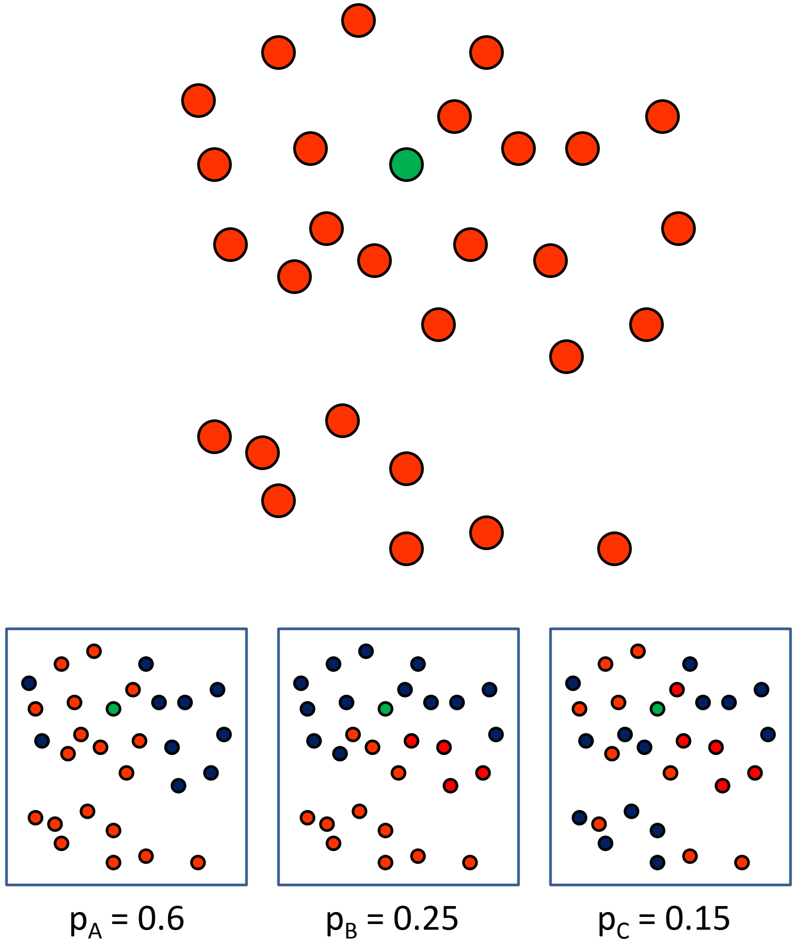
“two-stage” Steiner tree

The Model:

Instead of one set R, we are given probability distribution π over subsets of nodes.

E.g., each node v independently belongs to R with probability p_v

Or, may be explicitly defined over a small set of “scenarios”



“two-stage” Steiner tree

Stage I (“Monday”)

Pick some set of edges E_M at $cost_M(e)$ for each edge e

Stage II (“Tuesday”)

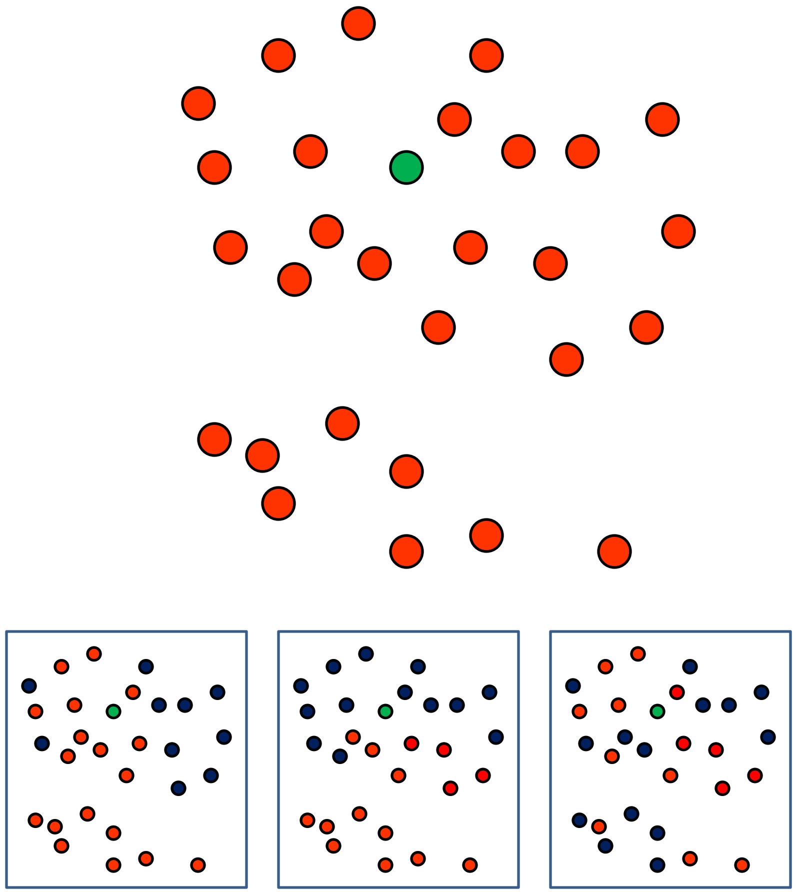
Random set R is drawn from π

Pick some edges $E_{T,R}$ so that $E_M \cup E_{T,R}$ connects R to root

cost change to $cost_{T,R}(e)$

Objective Function:

$$cost_M (E_M) + \mathbf{E}_{\pi} [cost_{T,R} (E_{T,R})]$$



$p_A = 0.6$ $p_B = 0.25$ $p_C = 0.15$

approximation algorithm

Objective Function:

$\mathrm{cost}_M\,(E_M) + \mathbf{E}_\pi\,[\,\mathrm{cost}_{T,R}\,(E_{T,R})\,]$

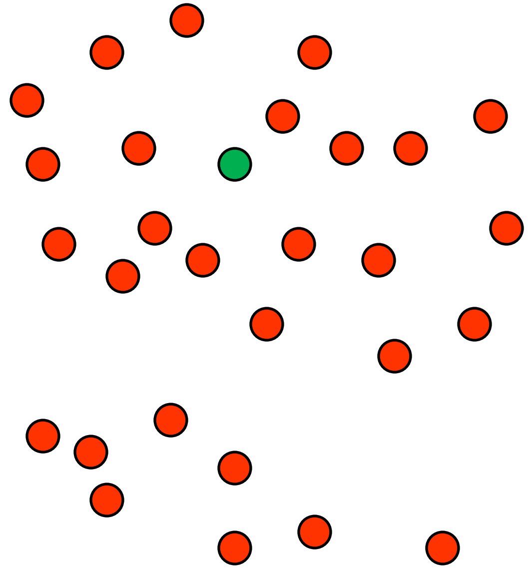
Optimum:

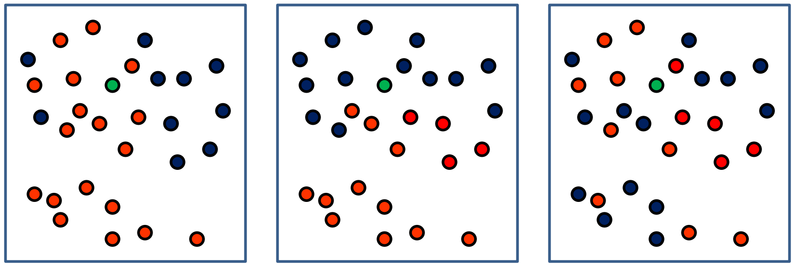
Sets E_M* and $E_{T,R}$* which achieve expected cost Z*

A c-approximation:

Find sets E_M and $E_{T,R}$ that achieve expected cost c.Z*

for some small factor c.

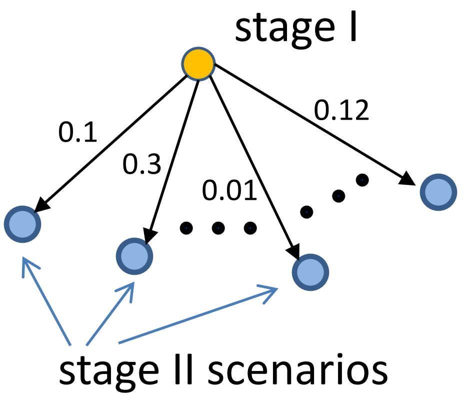




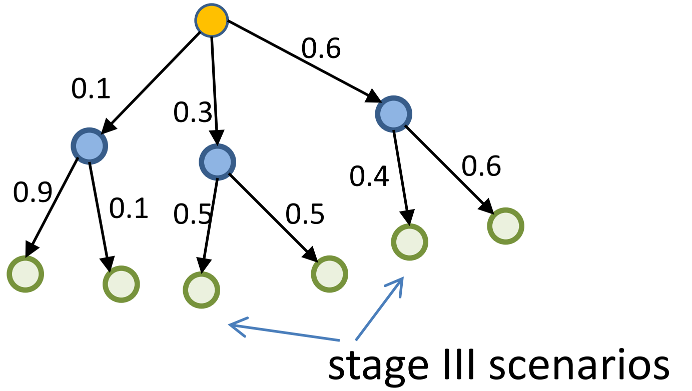
$p_A = 0.6$ $p_B = 0.25$ $p_C = 0.15$

pictures: two-stage and multi-stage

In each stage, the probability distribution π is progressively refined



And the costs change. Usually they increase…



another example: facility location

Input: Metric space

node set R of clients

facility costs f_v for each node v

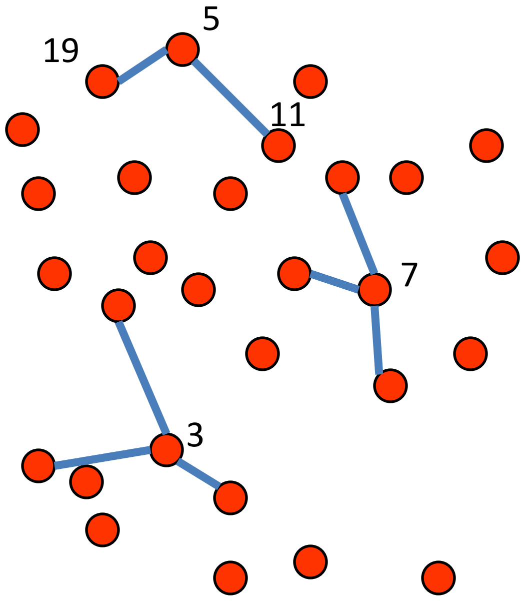
Output: node set F of facilities

Minimize:

$$\sum_{v \text{ in } F} f_v + \sum_{u \text{ in } R} \text{dist}(u, F)$$

Facts: 1.50-approx [Byrka '07]

1.463-hard [Guha Khuller '98]

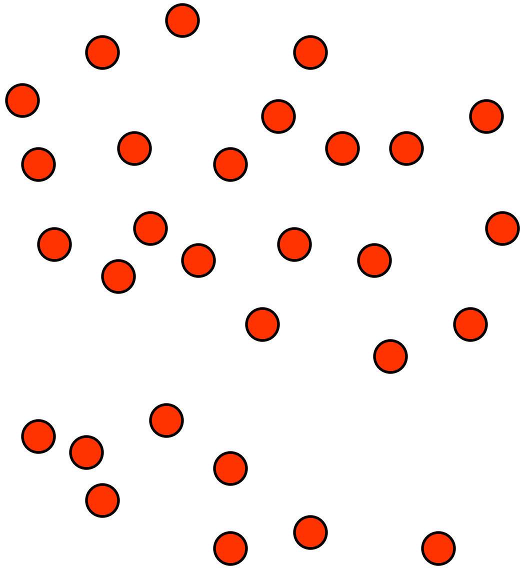


and the stochastic version

Initially facility at v costs f_v

Distribution π on tuples τ = (R, $f_v(\tau)$)

- random node set R of clients
- facility costs $f_v(\tau)$ in this scenario



Monday: buy facilities F_M

Tuesday: scenario τ drawn from π

buy some more facilities $F_T(\tau)$

minimize:

$$\sum_{v \text{ in } F_M} f_v + \mathbf{E}_{\tau \leftarrow \pi} \left[\sum_{v \text{ in } F_T} f_v(\tau) + \sum_{u \text{ in } R} \text{dist}(u, F_M \cup F_T) \right]$$

complexity?

- Stochastic discrete optimization problems can be solved using Mixed Integer Program formulations
  - no poly-time algorithms unless P=NP.

- Also, stochastic problems are harder than deterministic ones
  - E.g., many 2-stage stochastic versions of Shortest paths are NP-hard.
  - Two-stage stochastic linear programming is #P-hard.

background(1)

Scheduling with stochastic data

- Substantial work [Pinedo ’95]
- Also on approximation algorithms
  [Möhring Schulz Uetz, Skutella & Uetz, Scharbrodt et al, Souza & Steger,…]

Approximation Algorithms

- Resource provisioning using LP rounding
  [Dye Stougie Tomasgard; Nav. Res. Qtrly ’03]
- Approximation for Steiner tree, facility location
  [Immorlica Karger Minkoff Mirrokni SODA ’04]
- Facility location, vertex cover, etc using LP rounding
  [Ravi Sinha IPCO ’04]

background(2)

Main citations relevant to this talk:

The “Boosted Sampling” approach:

- two-stage problems — [Gupta Pal Ravi Sinha, STOC ’04]
- multistage problems — [Gupta Pal Ravi Sinha, APPROX ’05]

Solving and Rounding stochastic linear programs:

- two stage problems — [Shmoys Swamy, FOCS ’04]
- multistage problems — [Shmoys Swamy, FOCS ’05]

both reduce stochastic case to deterministic case in different ways

recap: two stage

- Given probability distribution π over the second-stage data
- Two stages of decision-making.
  - Monday: make anticipatory decisions based on π
  - Tuesday: make recourse decisions after seeing actual data.
- Minimize the expected cost incurred.

roadmap

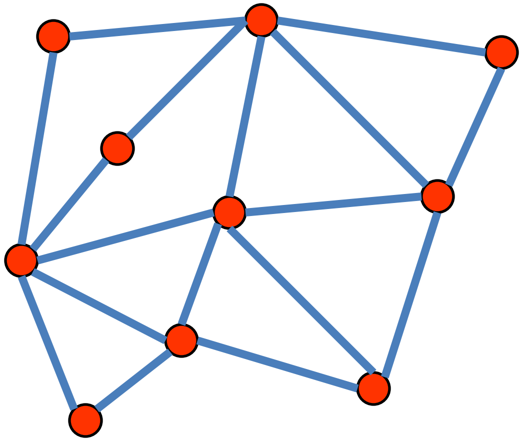
- example: stochastic vertex cover
  (using LPs)
- example: stochastic Steiner tree
  (using “boosted sampling”)
- comparison between the two general approaches:
  boosted sampling vs. LP-based approaches.

representations of π

- "Explicit scenarios" model
  - Complete listing of the sample space
- "Black box" access to probability distribution
  - generates an independent random sample from π
- Also, independent decisions
  - Each vertex v appears with probability p_v indep. of others.

vertex cover

vertex cover = set of vertices that hit all edges.



- Finding minimum cost vertex cover is NP-hard.

  2-approx: several algorithms

  easy one: solve the linear program relaxation and round

integer-program formulation

Boolean variable $x(v) = 1$ iff vertex v chosen in the vertex cover

minimize $\sum_v c(v)\, x(v)$

subject to

$x(v) + x(w) \geq 1$ for each edge (v,w) in edge set E

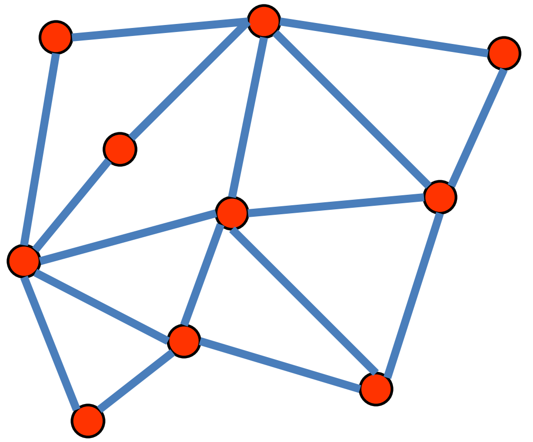
and

x's are in {0,1}

stochastic vertex cover

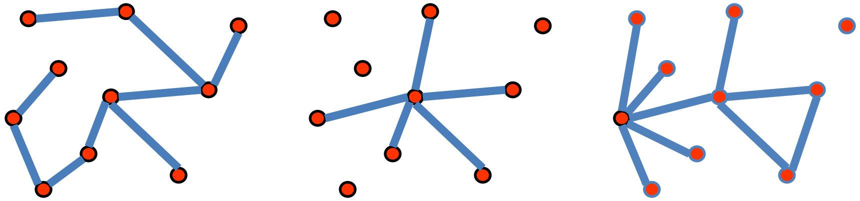
Explicit scenario model:

M scenarios explicitly listed.
Edge set E_k appears with prob. p_k



Vertex costs c(v) on Monday, $c_k(v)$ on Tuesday if scenario k appears.

Pick V_0 on Monday, V_k on Tuesday such that $(V_0 \cup V_k)$ covers E_k.



Minimize $c(V_0) + \mathbf{E}_k [\, c_k(V_k) \,]$

integer-program formulation

Boolean variable $x(v) = 1$ iff v chosen on Monday,
$y_k(v) = 1$ iff v chosen on Tuesday if scenario k realized

minimize $\sum_v c(v)\, x(v) + \sum_k p_k \left[\sum_v c_k(v)\, y_k(v) \right]$

subject to

$[\, x(v) + y_k(v) \,] + [\, x(w) + y_k(w) \,] \geq 1$ for each k, edge (v,w) in E_k

and

x's, y's are Boolean

linear-program relaxation

minimize $\sum_v c(v)\, x(v) + \sum_k p_k \,[\, \sum_v c_k(v)\, y_k(v) \,]$

subject to

$[\, x(v) + y_k(v) \,] + [\, x(w) + y_k(w) \,] \geq 1$ for each k, edge (v,w) in E_k

Now choose $V_0 = \{ v \mid x(v) \geq ¼ \}$, and $V_k = \{ v \mid y_k(v) \geq ¼ \}$

We are increasing variables by factor of 4
$\Rightarrow$ we get a 4-approximation

Note: if we have explicit multi-stage solution with k stages, gives 2k approximation

solving the LP and rounding

- This idea useful for many stochastic problems
  - Set cover, Facility location, some cut problems
- Tricky when the sample space is exponentially large
  - exponential number of variables *and* constraints
  - natural (non-trivial) approaches have run-times depending on the variance of the problem…
- Shmoys and Swamy approach:
  - consider this doubly-exponential vertex cover LP in black-box model
  - can approximate it arbitrarily well, smaller run-times.
  - solution has exponential size,

    but we need only polynomially-large parts of it at a time.

roadmap

- example: stochastic vertex cover
  (using LPs)

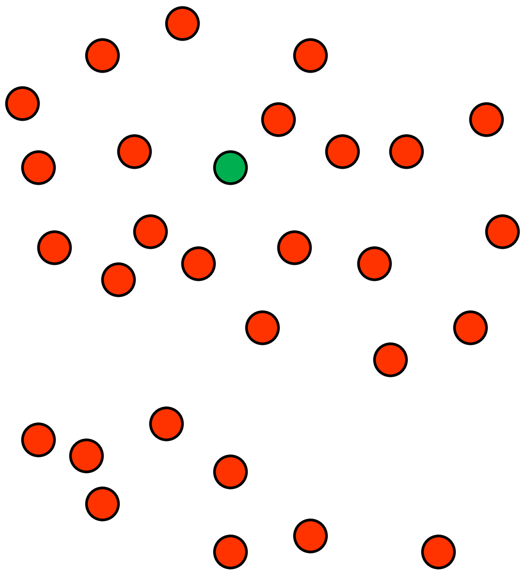
- example: stochastic Steiner tree
  (using “boosted sampling”)

- comparison between the two general approaches:
  boosted sampling vs. LP-based approaches.

two-stage Steiner tree

Stage I ("Monday")

Pick some set of edges E_M
at $cost_M(e)$ for each edge e



Stage II ("Tuesday")

Random set R is drawn from π

Pick some edges $E_{T,R}$ so that
$E_M \cup E_{T,R}$ connects R to root

Distribution π given as black-box

Objective Function:

$$cost_M (E_M) + \mathbf{E}_\pi [\, cost_{T,R} (E_{T,R}) \,]$$

$$\text{inflation } \lambda_{e,R} = \frac{cost_{T,R}(e)}{cost_M (e)}$$

simplifying assumption

"Proportional costs"

- On Tuesday, inflation for all edges is a fixed factor λ.
  i.e., there is some λ such that $cost_{T,R}(e) = \lambda\, cost_M(e)$.

- Results generalize to case when inflation λ_R depends on scenario, but still same for all edges.

- If different edges have different inflation,
  Steiner tree problem **much** harder to approximate.

Bottom line: every edge costs exactly λ times more on Tuesday

Objective Function: $c_M(E_M) + \lambda\, \mathbf{E}_\pi [\, c_M(E_{T,R}) \,]$

boosted sampling algorithm

- Sample from the distribution π of clients λ times
  - Let sampled set be *S*

inflation factor

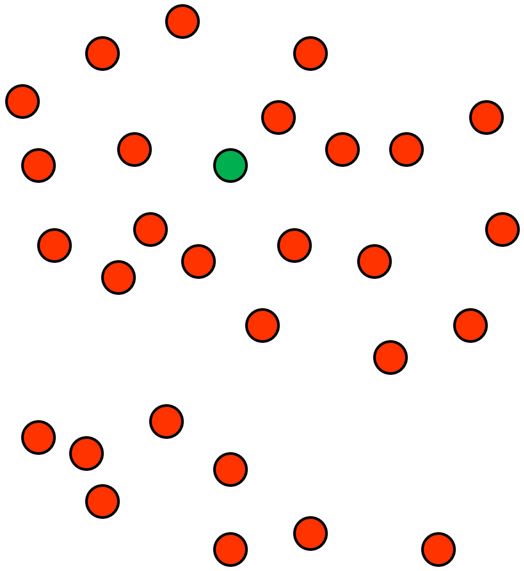
- Build minimum spanning tree T_0 on $S \cup root$
  - Recall: MST is a 2-approximation to Minimum Steiner tree
- 2nd stage: actual client set *R realized*
  - Extend T_0 with some edges in T_R so as to span *R*

Theorem: 4-approximation to Stochastic Steiner Tree

Algorithm: Illustration

Input, with *$\lambda=3$*

- Sample λ times from client distribution
- Build MST T_0 on S
- When actual scenario R is realized, extend T_0 to span R in a min cost way



the analysis

- 1^{st} stage: Sample from the distribution of clients λ times
  - Build minimum spanning tree T_0 on $S \cup root$
- 2^{nd} stage: actual client set *R realized*
  - Extend T_0 with some edges in T_R so as to span R

Proof Strategy:

$$\underbrace{\mathrm{OPT} = c(T_0^*) + E_\pi[\,\lambda \cdot c(T_R^*)]}$$

- **E**[Cost(1^{st} stage)] ≤ 2 × OPT
- **E**[Cost(2^{nd} stage)] ≤ 2 × OPT

Analysis of 1st stage cost

Claim 1: **E**[cost(T_0) ≤ 2 × OPT

Proof: Our λ samples: $S = S_1 \cup S_2 \cup \ldots \cup S_\lambda$

If we take T_0^* and all the $T_{S_j}^*$ from OPT's solution,
we get a feasible solution for a Steiner tree on S ∪ root.

An MST on S costs at most 2 times this Steiner tree.

Analysis of 1st stage cost(formal)

- Let $OPT = c(T_0^*) + \sum_X p_X \cdot \lambda \cdot c(T_X^*)$
- Claim: $E[c(T_0)] \leq 2.OPT$
- Our λ samples: $S=\{S_1, S_2, \ldots, S_\lambda\}$

$$MST(S) \leq 2\{c(T_0^*) + c(T_{S_1}^*) + \ldots + c(T_{S_\lambda}^*)\}$$

$$E[MST(S)] \leq 2\{c(T_0^*) + E[c(T_{S_1}^*)] + \ldots + E[c(T_{S_\lambda}^*)]\}$$

$$= 2\{c(T_0^*) + \lambda E_X[c(T_X^*)]\}$$

the analysis

- 1^{st} stage: Sample from the distribution of clients λ times
  - Build minimum spanning tree T_0 on $S \cup root$
- 2^{nd} stage: actual client set *R realized*
  - Extend T_0 with some edges in T_R so as to span R

Proof Strategy:

$$\underbrace{\mathrm{OPT} = c(T_0^*) + E_\pi[\,\boldsymbol{\lambda} \cdot c(T_R^*)]}$$

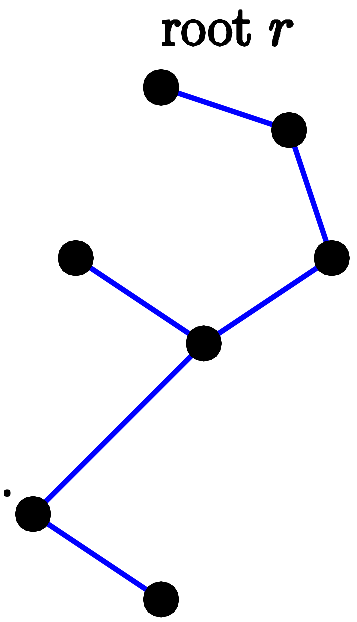
- **E**[Cost(1^{st} stage)] ≤ 2 × OPT
- **E**[Cost(2^{nd} stage)] ≤ 2 × OPT

a “cost sharing” scheme for MST

Associate each node v with its parent edge pe_v

1. [“Budget Balance”]
   cost of MST(S) = $\sum_{v \in S} c(pe_v)$.

2. [“Late-comers OK”]
   If $S = B \cup G$, then
   spanning-tree(B) $\cup$ $\{pe_v \mid v \in G\}$ spans S.

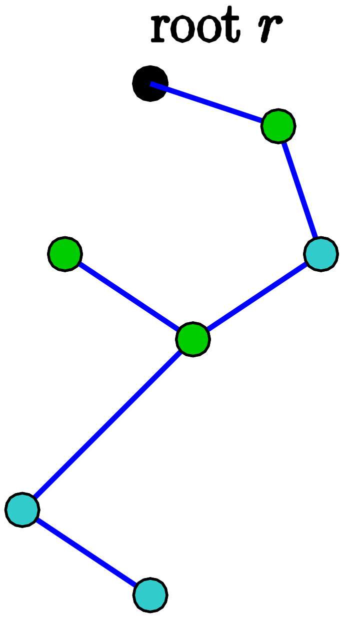


a useful "cost sharing" scheme

Associate each node v with its parent edge pe_v

1. ["Budget Balance"]
   cost of MST(S) = $\sum_{v \in S} c(pe_v)$.

2. ["Late-comers OK"]
   If $S = B \cup G$, then
   spanning-tree(B) $\cup$ $\{pe_v \mid v \in G\}$ spans S.



a useful "cost sharing" scheme

Associate each node v with its parent edge pe_v

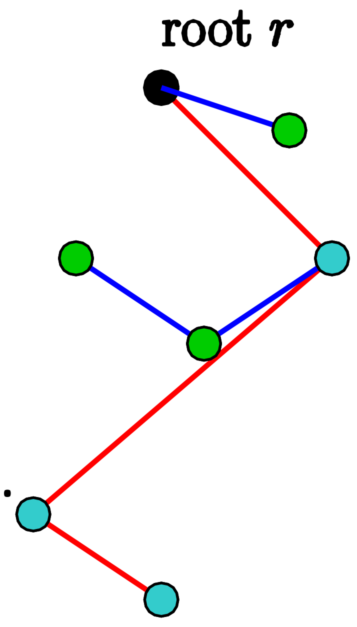
1. ["Budget Balance"]

   cost of MST(S) = $\Sigma_{v \in S}\, c(pe_v)$.

2. ["Late-comers OK"]

   If $S = B \cup G$, then

   spanning-tree(B) $\cup \{pe_v \mid v \in G\}$ spans S.



Let $pe(X) = \{pe_v \mid v \in X\}$.

Analysis of 2nd stage cost

- Consider this:

  take $\lambda+1$ samples from the distribution, instead of λ

- **E**[Cost of MST on these $\lambda+1$ samples] $\leq \frac{2(\lambda+1)\,\text{OPT}}{\lambda}$

- Pick one sample at random, call it real terminal set R.

  Others λ samples are $S_1, S_2, \ldots, S_\lambda$ with $S = \cup S_j$

Expected cost of pe(R) $\leq \frac{\text{MST}(R \cup S)}{\lambda+1} \leq \frac{2\ \text{OPT}}{\lambda}$

Analysis of 2^{nd} stage cost

Expected cost of pe(R) $\leq \frac{2\ \mathrm{OPT}}{\lambda}$

But pe(R) $\cup$ MST(S) is a feasible Steiner tree for R.

$\Rightarrow$ buying pe(R) is a feasible action for the second stage!

Hence, **E**[cost of second stage] ≤ **E**[**λ** c(pe(R))] ≤ 2 OPT.

Recap

- Algorithm for Stochastic Steiner Tree:
  - 1st stage: Sample λ times, build MST
  - 2nd stage: Extend MST to realized clients
- **Theorem:** Boosted-Sample is a 4-approximation to Stochastic Steiner Tree.
- Other problems like Facility location, Vertex cover, also have such sampling based algorithms
  - Require analogous notions of cost-shares for these problems
  - we call these “strict” cost-shares.

roadmap

- example: stochastic vertex cover
  (using LPs)
- example: stochastic Steiner tree
  (using “boosted sampling”)
- comparison between the two general approaches:
  boosted sampling vs. LP-based approaches.

a quick comparison

Boosted Sampling

- combinatorial
- require λ samples
- cost-shares: “primal-dual”?
- only proportional costs

Shmoys-Swamy

- convex programming-based
- require more samples
- primal-only techniques
- general cost structure

last slide...

- This was perhaps the simplest model, still interesting results
  - what about algorithms for other models?
- Can we improve the approximation bounds given by these algorithms?
  - is stochastic Steiner tree actually harder than its deterministic variant?
- Which other problems can be solved in this model?
  - Not known how to solve set cover using boosted sampling.