Approximations for Multi-Stage Stochastic Optimization

Part I

Anupam Gupta
Carnegie Mellon University

Autumn School on Approximation Algorithms for Stochastic Optimization

stochastic optimization

Question: How to model uncertainty in the inputs?

- data may not yet be available
- obtaining exact data is difficult/expensive/time-consuming
- but we do have some stochastic predictions about the inputs

Goal: design algorithms to 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

Development of **approximation algorithms** for
NP-hard stochastic optimization problems is relatively new.

Several different models, several different techniques.

I'll talk about **multi-stage stochastic optimization with recourse**

- model is interesting in its own right
- algorithmic techniques are interesting and useful

Marc and Vish will talk about other models.

the major themes

1 two-stage optimization "with recourse"

linear-programming based techniques

approx algos for stochastic vertex cover, MST

combinatorial techniques

cost shares

approx algos for stochastic Steiner tree, etc.

2 extensions and deeper investigation

how to solve large stochastic LPs

the "sample average approximation" (SAA) method

stochastic optimization online

example 1: Steiner tree

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.

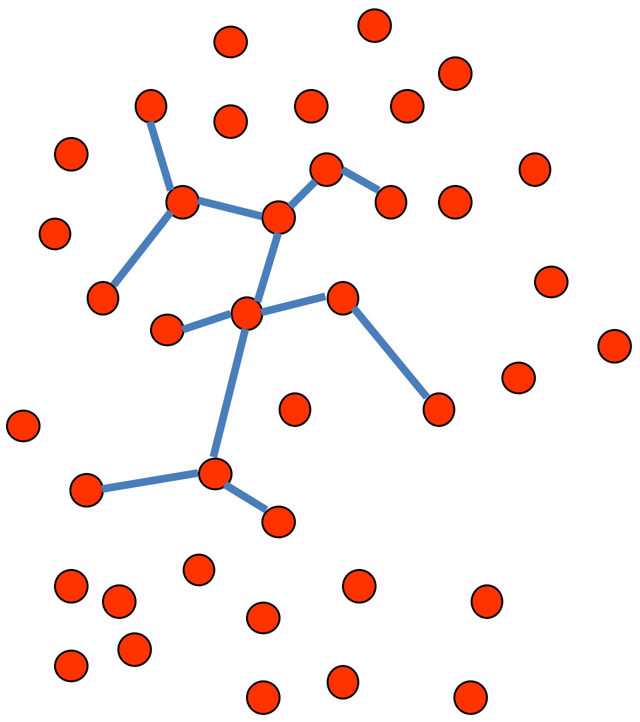
Stochastic Question:

actual terminal set appears tomorrow

(say) random sample of 10% of nodes

but edges costs are changed tomorrow

Q: What should you do?



example 1: Steiner tree

Input: a metric space

a root vertex r

a subset R of terminals

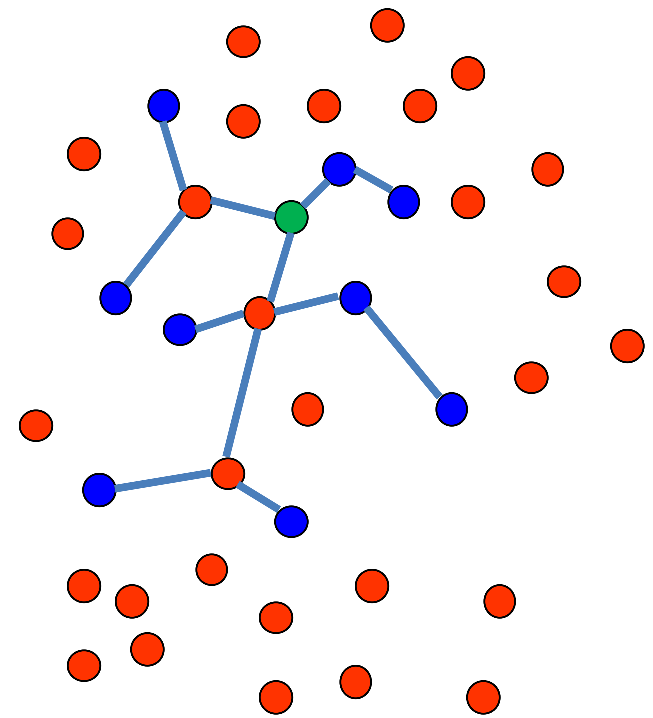
Output: a tree T connecting R to r
of minimum length/cost.

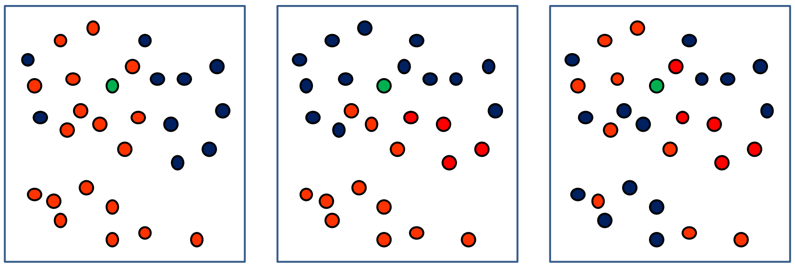
Q: how to represent the distribution?

i.i.d samples?

list of "scenarios"?

just black box access?





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

example 1: Steiner tree

Doesn't "online algorithms/competitive analysis" handle such multi-stage decision-making questions?

The standard approach of competitive analysis
compares our algorithm's performance
to an optimal algorithm that can **see the future**.

In stochastic analysis, we want to compare our algorithm's performance
to the performance of best possible **two-stage algorithm**.

Want to level the playing field…

observations

For stochastic problems:

1. want to avoid competitive analysis as much as possible.
2. want efficient algorithms
3. want provable guarantees on performance

We want algorithms that work for general distributions

Don't want to rely on distributions being "nice", when possible.

(But sometimes, life is too difficult without such assumptions.)

observations

Again, the framework is that of approximation algorithms

our **E**[performance] vs. **E**[performance of optimal algorithm]

Stochastic Optimization is a long-studied area of research.

Dantzig's paper on "Linear Programming under Uncertainty" in 1955

Several books on the subject.

The approximation/online algorithms effort relatively new.

(stochastic scheduling, for special distributions)

(and online algorithms on stochastic models for paging)

two-stage optimization “with recourse”

LINEAR PROGRAMMING UNDER UNCERTAINTY

GEORGE B. DANTZIG

The Rand Corporation, Santa Monica, Cal.

Summary

The essential character of the general models under consideration is that activities are divided into two or more stages. The quantities of activities in the first stage are the only ones that are required to be determined; those in the second (or later) stages can not be determined in advance since they depend on the earlier stages and the random or uncertain demands which occur on or before the latter stage. It is important to note that the set of activities are assumed to be *complete* in the sense that, whatever be the choice of activities in the earlier stages (consistent with the restrictions applicable to their stage), there is a possible choice of activities in the latter stages. In other words *it is not possible to get in a position where the programming problem admits of no solution.*

two-stage optimization "with recourse"

today things are cheaper, tomorrow prices go up by λ

but today we only know the demand distribution π,
tomorrow we'll know the real demands (drawn from π)

"with recourse" just means that whatever we may do today,
we can always fix things up to get a feasible solution
tomorrow

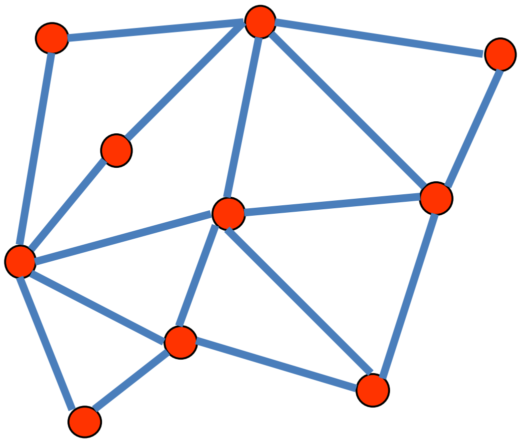
such stochastic problems are (potentially) harder
than their deterministic counterparts

example 2: vertex cover

Input: graph, nodes have costs c(v).

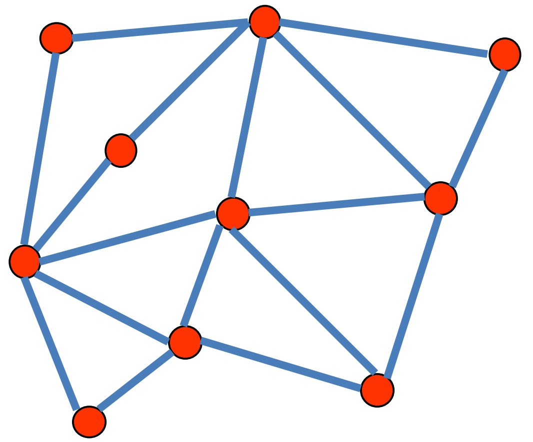
Vertex cover: subset of nodes that
hit ("cover") every edge.

Output: pick the vertex cover with
least cost.



2-stage stochastic vertex cover

Each potential outcome is called a "scenario", and π gives probability distribution over scenarios.



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

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

Minimize $c(V_0) + \mathbf{E}_{k \leftarrow \pi} [\, c_k(V_k) \,]$

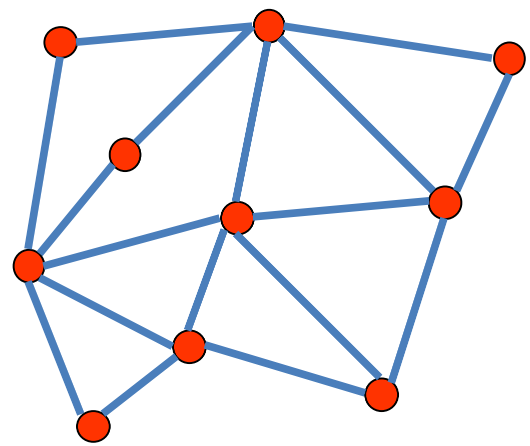
[Ravi Sinha 04, Immorlica Karger Mahdian Mirrokni 04, Shmoys Swamy 04]

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.
  - special case of black-box, sometimes much easier
- other reasonable special distributions?

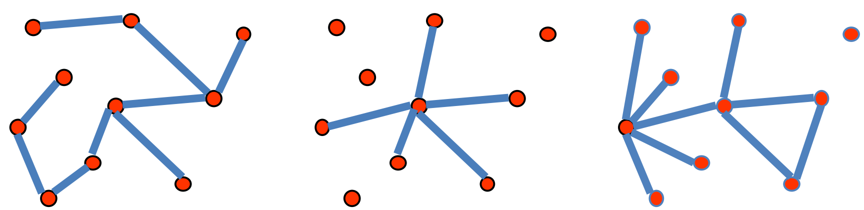
2-stage stochastic vertex cover

Each potential outcome is called a "scenario", and π gives probability distribution over scenarios.



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

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



$p_1 = 0.1$ $p_2 = 0.6$ $p_3 = 0.3$

Minimize $c(V_0) + \mathbf{E}_{k \leftarrow \pi}[\, c_k(V_k) \,]$

Let's consider "explicit scenarios" for this discussion

how to approximate ordinary vertex cover?

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}~~ $x \geq 0$

Taking all the vertices such that $x^*(v) \geq ½$ will give a feasible solution

at most twice the optimum LP solution,
hence at most twice integer solution

same idea for stochastic vertex cover

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

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

and

~~x's, y's are Boolean~~

same idea for stochastic vertex cover

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

subject to

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

Now choose $V_0 = \{ v \mid x^*(v) \geq 1/4 \}$, and $V_k = \{ v \mid y^*_k(v) \geq 1/4 \}$

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

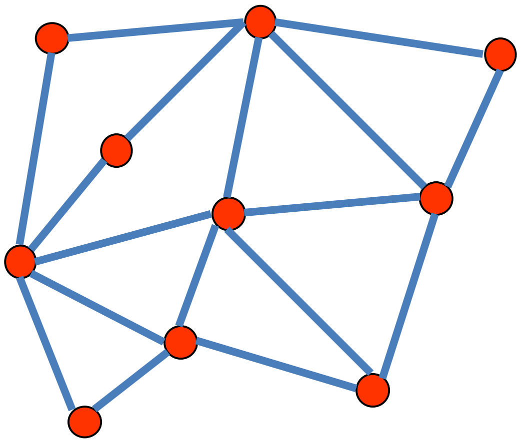
Problem set: get a 2-approximation, avoid all extra loss!

example 3: stochastic MST

Input: graph G = (V,E)

"Monday" edge lengths $c_M(e)$

Scenario k specifies $c_{T,k}(e)$, the edge lengths on Tuesday for that outcome.



Output: pick some edges on Monday, pick some on Tuesday once actual scenario appears.

Minimize expected total edge cost.

We'll see an approximation algorithm, and a matching hardness result in the explicit scenario model.

[Dhamdhere Ravi Singh 05]

main idea: write an LP

$\min \sum_e [\, c_M(e)\, x_e + \sum_k p_k\, c_{T,k}(e)\, x_{k,e} \,]$

$\sum_{e \text{ in } \delta(S)} [\, x_e + x_{k,e} \,] \geq 1$ for all scenarios k, for all cuts (S,S')

$x_e, x_{k,e} \geq 0$ for all scenarios k, for all edges

Again, solve this using an LP solver to get an optimal fractional solution x*

How to round?

In fact, how to round such an LP even for basic (non-stochastic) MST?

main idea: write an LP

$$\min \sum_e [\, c_M(e)\, x_e + \sum_k p_k\, c_{T,k}(e)\, x_{k,e} \,]$$

$$\sum_{e \text{ in } \delta(S)} [\, x_e + x_{k,e} \,] \geq 1 \qquad \text{for all scenarios } k, \text{ for all cuts } (S,S')$$

$$x_e,\, x_{k,e} \geq 0 \qquad \text{for all scenarios } k, \text{ for all edges}$$

Simple algorithm: sample each edge indep. w.p. x_e.

Repeat this independently for O(log n) rounds.

Theorem: The union is a spanning subgraph whp. (Can then drop edges)

Proof Sketch: the number of connected components halves at each round

main idea: write an LP

$$\min \sum_e [\, c_M(e)\, x_e + \sum_k p_k\, c_{T,k}(e)\, x_{k,e} \,]$$

$\sum_{e \text{ in } \delta(S)} [\, x_e + x_{k,e} \,] \geq 1$ for all scenarios k, for all cuts (S,S')

$x_e, x_{k,e} \geq 0$ for all scenarios k, for all edges

Use the same idea: independently add each edge to A wp x_e, to B_k wp $x_{k,e}$

Repeat this O(log nm) times, where n vertices, and m scenarios.

Claim: the expected cost is O(log nm) times the LP cost

Claim: Whp, for each k, A union B_k is a spanning tree of the graph.

Can get O(log n + log λ) approximation too

$\lambda = \max(c_{T,k}(e) \,/\, c_M(e))$

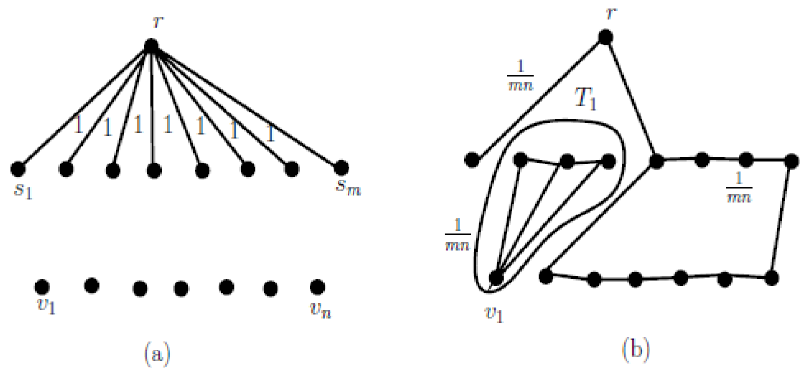
hardness: reduction from set cover

Set Cover: given a universe U, subsets S_1, S_2, …, S_m, find a smallest collection of these sets whose union is U. Let n = |U|.

Theorem: the greedy algorithm is an (ln n) approximation

Theorem: there is no (1-ε)(ln n) approximation unless P = NP, even for m = poly(n)

Theorem: C-approx algo for stochastic MST ⇒ C-approx algo for set cover



results so far

Stochastic Vertex Cover

Theorem: 4-approximation for explicit scenarios. (2-approx in problems.)

Theorem: better than 2-approx refutes the unique games conjecture.

Stochastic MST

Theorem: O(log nm)-approximation for explicit scenarios.

Theorem: better than O(log n)-approx even for m = poly(n) => P=NP.

both based on solving the LP, and rounding; explicit scenarios model

combinatorial approaches

problem 1: two-stage Steiner tree

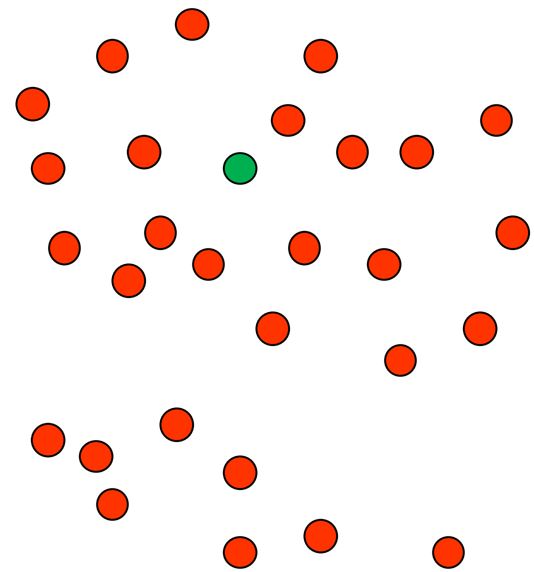
Stage I ("Monday")

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:

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

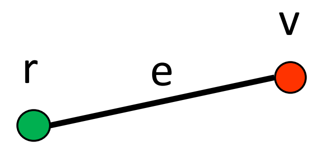
inflation $\lambda_{e,R} = \dfrac{cost_{T,R}(e)}{cost_M(e)}$

a thought experiment

Trivial instance:

costs 1, and $\lambda > 1$.

v materializes with probability p.



Should we buy the edge in the first stage or the second?

Optimal strategy: if $\lambda p > 1$ then buy, else wait.

Can estimate p well (up to additive $\pm\,\varepsilon$) using $O(1/\varepsilon^2)$ samples

Q: What value of ε is good enough?

Slightly different algorithm: sample from the distribution λ times

if v materializes in any of these, buy, else wait.

will buy with probability $q := 1 - (1-p)^\lambda$ "$\approx p\lambda$"

simplifying assumption for this part

"Proportional costs"

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

- If different edges have different inflation,
  at least $\Omega(\log n)$ hard – captures StocMST.

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

Can extend to setting where costs are proportional but scenario dependent.

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

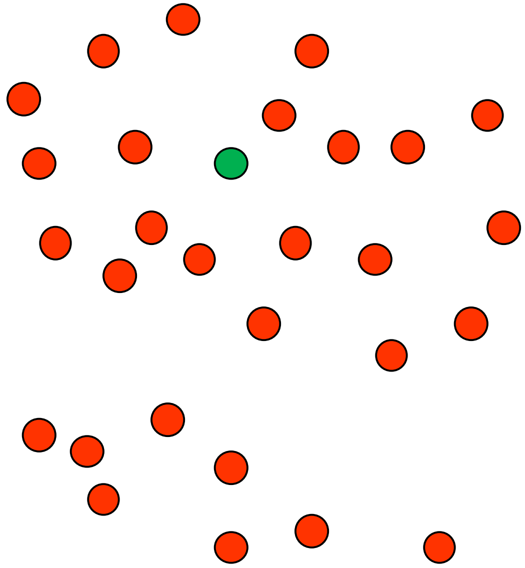
- 2^{nd} stage: actual client set R *realized*
  - Extend T_0 with some edges in T_R so as to span R

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:

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

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

analysis of 1st stage cost

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

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 $S \cup$ 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={S₁, S₂,…,S_λ}*

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

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

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

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:

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

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

a useful "cost sharing" scheme

Associate each node v with its parent edge parent_v

1. ["Budget Balance"]

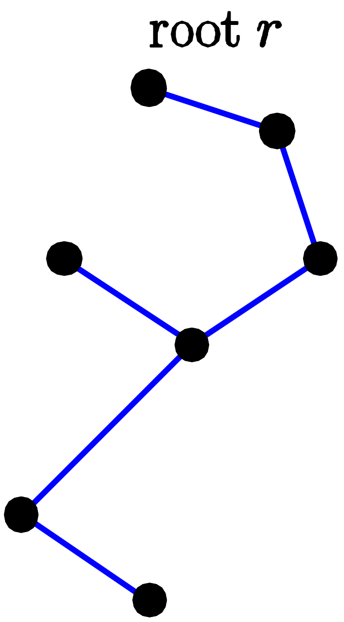
   cost of MST(S) = $\sum_{v \in S} c(\text{parent}_v)$.

2. ["Late-comers OK"]

   If $S = B \cup G$, then

   spanning-tree(B) $\cup \{\text{parent}_v \mid v \in G\}$

   spans S.



a useful "cost sharing" scheme

Associate each node v with its parent edge parent_v

1. ["Budget Balance"]

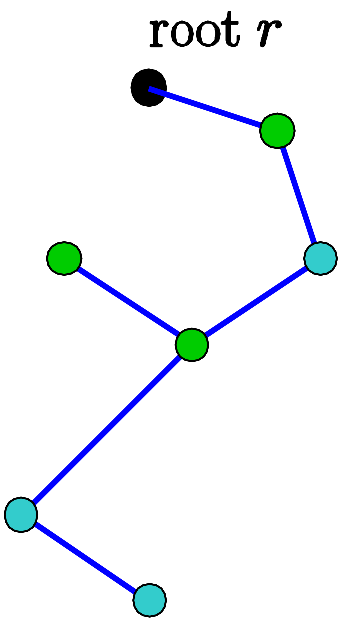
   cost of MST(S) = $\sum_{v \in S} c(\text{parent}_v)$.

2. ["Late-comers OK"]

   If $S = B \cup G$, then

   spanning-tree(B) $\cup \{\text{parent}_v \mid v \in G\}$

   spans S.



a useful "cost sharing" scheme

Associate each node v with its parent edge parent_v

1. ["Budget Balance"]

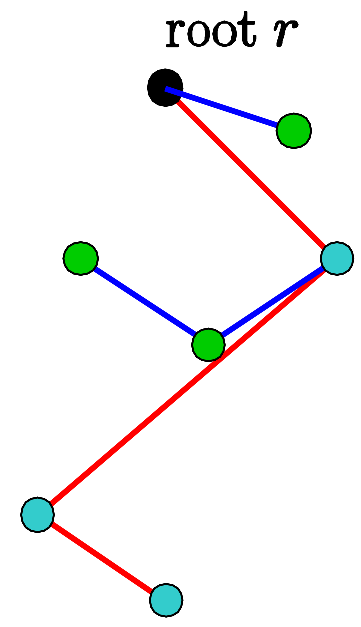
   cost of MST(S) = $\sum_{v \in S} c(\text{parent}_v)$.

2. ["Late-comers OK"]

   If $S = B \cup G$, then

   spanning-tree(B) $\cup \{\text{parent}_v \mid v \in G\}$

   spans S.



Let $\text{parent}(X) = \{\text{parent}_v \mid v \in X\}$.

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] $\leq \dfrac{2(\lambda+1)\,\mathrm{OPT}}{\lambda}$

- 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 parent(R) $\leq \dfrac{\mathrm{MST}(R \cup S)}{\lambda+1} \leq \dfrac{2\ \mathrm{OPT}}{\lambda}$

analysis of 2nd stage cost

Expected cost of parent(R) $\leq \frac{2\ \text{OPT}}{\lambda}$

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

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

Hence, **E**[cost of second stage] $\leq$ **E**[**λ** c(parent(R))] $\leq$ 2 OPT.

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

boosted sampling framework

Idea: take λ samples from the distribution,
build a solution on those as first stage solution.

for proof: show "strict" cost-sharing scheme for problem/algorithm

Pros: take very few samples (only λ), very simple algorithm

Cons: works only for uniform inflation, need to prove cost-sharing scheme

Can currently show for Steiner tree/forest, vertex cover, facility location

Good cost-shares do not exist for set cover.

Question: for what other problems do such good cost-shares exist?

results so far

Stochastic Vertex Cover

Theorem: 4-approximation for explicit scenarios. (2-approx in problems.)

Theorem: better than 2-approx refutes the unique games conjecture.

Stochastic MST

Theorem: O(log nm)-approximation for explicit scenarios.

Theorem: better than O(log n)-approx even for m = poly(n) => P=NP.

Stochastic Steiner Tree

Theorem: 4-approximation for black-box, uniform inflation.

Theorem: 1.00001-hard in the classical (non-stochastic) setting.

some citations(1)

Scheduling with stochastic data

- Substantial work [Pinedo '95]
- Also on approximation algorithms ← see Marc's and Vish's talks
  [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] ← see problem set
- Facility location, vertex cover, etc using LP rounding
  [Ravi Sinha IPCO '04] ← vertex cover example from this one

background(2)

the linear programming approach:

two stage problems (ES) [Ravi Sinha IPCO ’04]

vertex cover example from this one

two stage problems (BB) [Shmoys Swamy, FOCS ’04]

next lecture

The combinatorial approach:

two-stage problems [Gupta Pal Ravi Sinha, STOC ’04]

all “reduce” stochastic case to deterministic case in different ways