

Approximation Algorithms for Stochastic Combinatorial Optimization

Part II: sampling theorems, multistage problems and other extensions

Anupam Gupta
Carnegie Mellon University

two-stage with recourse

There are two stages of decision-making

- initially we perform some **anticipatory actions**
- then the instance is revealed to us, drawn from distribution π
- we may take some more **recourse actions** at this point

Want to minimize:

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

representations of π

- “Explicit scenarios” model
 - Complete listing of the sample space
- “Black box” access to probability distribution
 - generates an independent sample from probability distribution π

Solving doubly-exponential LPs: the Shmoys—Swamy method

the vertex cover LP

$$\text{minimize } F(x) = \sum_v c(v) x(v) + \sum_k p_k \left[\sum_v c_k(v) y_k(v) \right]$$

subject to

$$\left[x(v) + y_k(v) \right] + \left[x(w) + y_k(w) \right] \geq 1 \quad \text{for each } k, \text{ edge } (v,w) \text{ in } E_k$$

Let's rewrite this compactly:

$$\text{minimize } F(x) = \sum_v c(v) x(v) + \sum_k p_k f_k(x) \text{ s.t. } 1 \geq x \geq 0 \text{ for all } x.$$

where:

$$f_k(x) = \min \sum_v c_k(v) y_k(v)$$

$$\text{s.t. } \left[x(v) + y_k(v) \right] + \left[x(w) + y_k(w) \right] \geq 1 \quad \text{for each } k, (v,w) \text{ in } E_k$$

Each $f_k(x)$ is convex, so $F(x)$ is a convex objective function.

but we don't know how to evaluate $F(x)$...

solving a convex program

Minimize convex function $F(x)$ s.t. x in $\mathbf{P}$
using the ellipsoid method.

Given a point y :

if y not in $\mathbf{P}$, return a violated inequality

if y in $\mathbf{P}$, add new constraint $F(x) \leq F(y)$ to
get new polytope $\mathbf{P}'$

(enough to cut using the “subgradient”)

In our case, computing subgradients is hard!
and computing F is hard.

solving a convex program

Minimize convex function $F(x)$ s.t. x in $\mathbf{P}$
using the ellipsoid method.

Given a point y :

if y not in $\mathbf{P}$, return a violated inequality

if y in $\mathbf{P}$, add new constraint $F(x) \leq F(y)$ to
get new polytope $\mathbf{P}'$

(enough to cut using the “subgradient”)

Shmoys and Swamy show another way how to compute
approximate subgradients using sampling, and that
approximate subgradients suffice for $(1+\epsilon)$ -approx

Scenario Reduction: the Sample Average Approximation method

scenario reduction

Question: What happens if we

- sample N scenarios from the black box for π
- then use the explicit-scenario techniques to get a good approximation for this sampled problem?

Answer: It almost suffices if N is large enough.

Question: what is “large enough”?

a first idea you'd try

Want to minimize

$$F(x) = c(x) + \mathbf{E}_{\omega \leftarrow \pi} [g(x, \omega)]$$

over x in some set X . Say x^* is the minimizer.

Sample N times, and define

$$F'(x) = c(x) + \frac{\sum_k g(x, \omega_k)}{N}$$

Clearly, if $F(x) \approx F'(x)$ for all x in X , a minimizer for one is also an (almost)-minimizer for the other.

of course, N depends on the various parameters of the problem...

don't want "large" to depend on these properties...

some “nice” conditions

Want to minimize

$$F(x) = c(x) + \mathbf{E}_{\omega \leftarrow \pi} [g(x, \omega)]$$

over x in some set X . Say x^* is the minimizer.

Suppose the functions satisfy:

1. [Non-negativity] both functions $c()$ and $g()$ are non-negative.
2. [“Lipschitz”-condition] for all ω and for x, x'

$$| g(x, \omega) - g(x', \omega) | \leq \lambda (c(x) + c(x'))$$

Both satisfied as long as all inflations bounded by some universal factor λ .

a sampling theorem

Want to minimize

$$F(x) = c(x) + \mathbf{E}_{\omega \leftarrow \pi} [g(x, \omega)]$$

over x in some set X . Say x^* is the minimizer.

Theorem

[Charikar Chekuri Pal]

Sample $N = \text{poly}(\log |X|, \lambda)$ times, and find the minimizer x' of the sample average

$$F'(x) = c(x) + \frac{\sum_k g(x, \omega_k)}{N}$$

Then $F(x') \leq 1.001 F(x^*)$ with probability .99999.

Note: other SAA theorems known previously, this is one example.

intuition for proof

Bad case for naïve approach:
some of the ω 's incur a huge cost.

Define: ω is “high” if $g(x_0, \omega) \geq M$ for some $M \approx \lambda \text{ OPT} / \epsilon$.

On these **high** ω 's, doesn't matter what **x** you choose,
the probability of getting high omega is small ($\approx \epsilon / \lambda$), and

$$| g(x, \omega) - g(x^*, \omega) | \leq \lambda c(x) + \lambda c(x^*)$$

On **low** ω 's, the sample average is a good approximation
to the real average (upto additive $\pm \epsilon \text{ OPT}$)

extensions

- What about NP-hard problems?
 - If we can only find the approximate minimizer of F' , just repeat the process a few times and take the best one.
- What about maximization problems?
 - No worries, it works...
- What about linear programs?
 - No worries, it works.
 - Essentially replace $\log |X|$ by the dimension n of the LP.

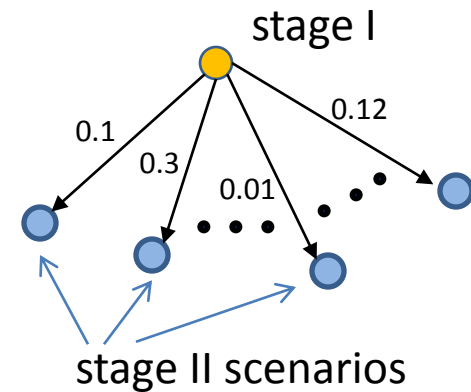
Approximations for Multistage Stochastic Programs

k stages of decision making

k-stages of decision-making

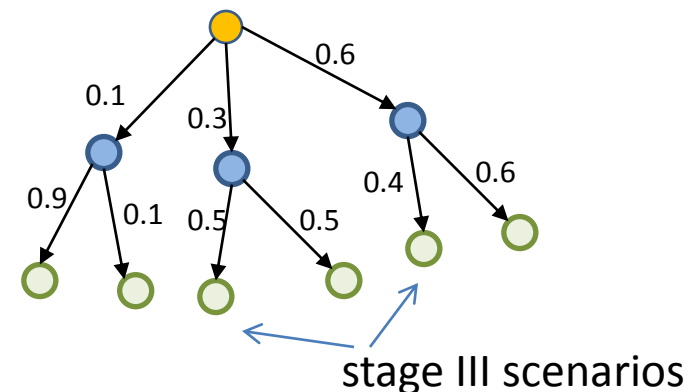
[Gupta Pal Ravi Sinha '05]

$O(k)$ approximation for
k-stage Steiner tree.

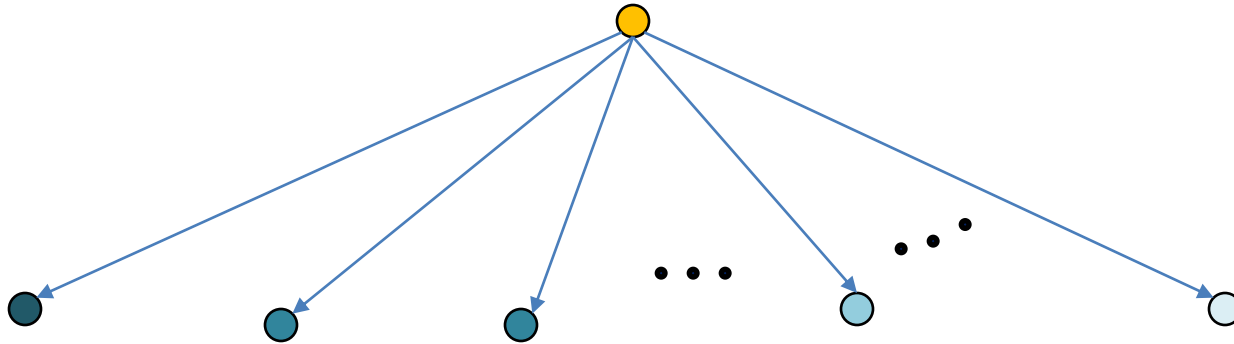


[Swamy Shmoys '05]

$O(k)$ for k-stage facility location,
vertex cover.



Say, the cost increases by a factor of $\lambda = 2$ from the first to the second stage



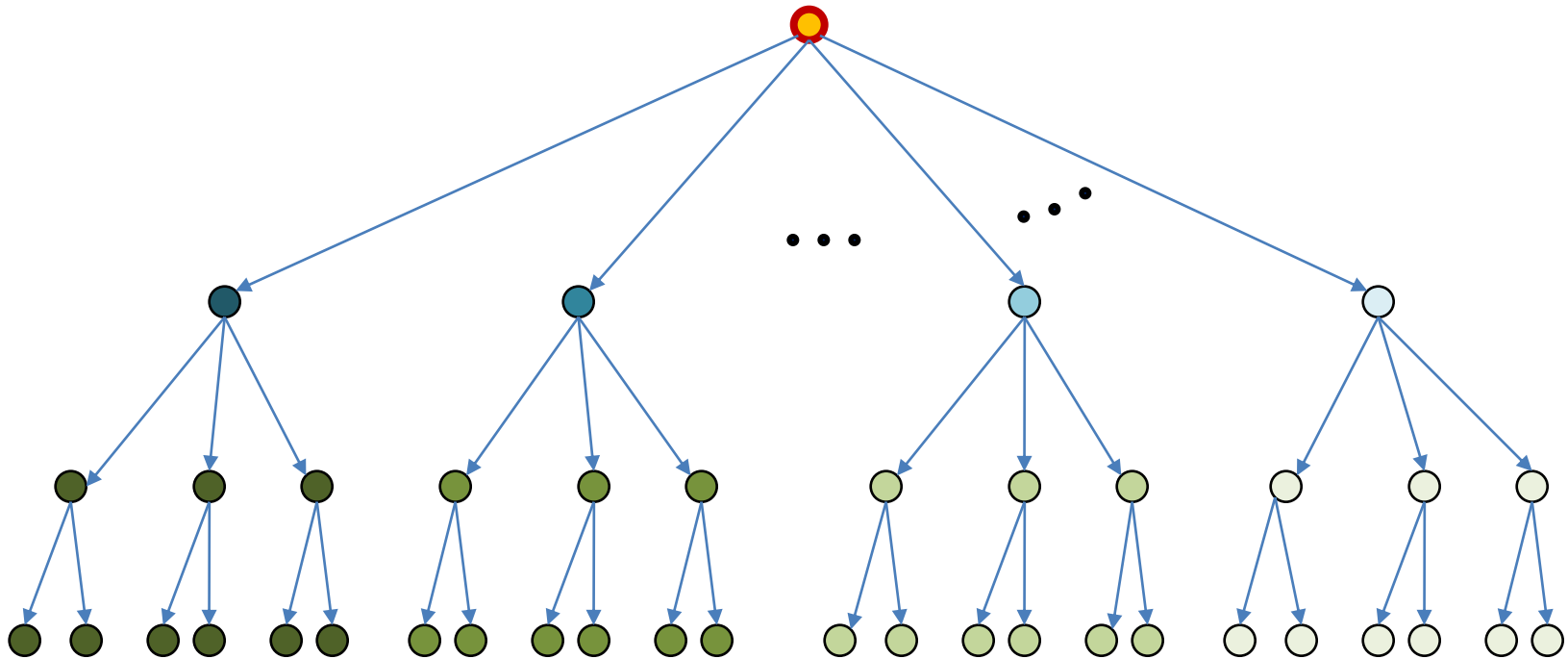
Boosted Sampling Algorithm:

Sample λ scenarios.

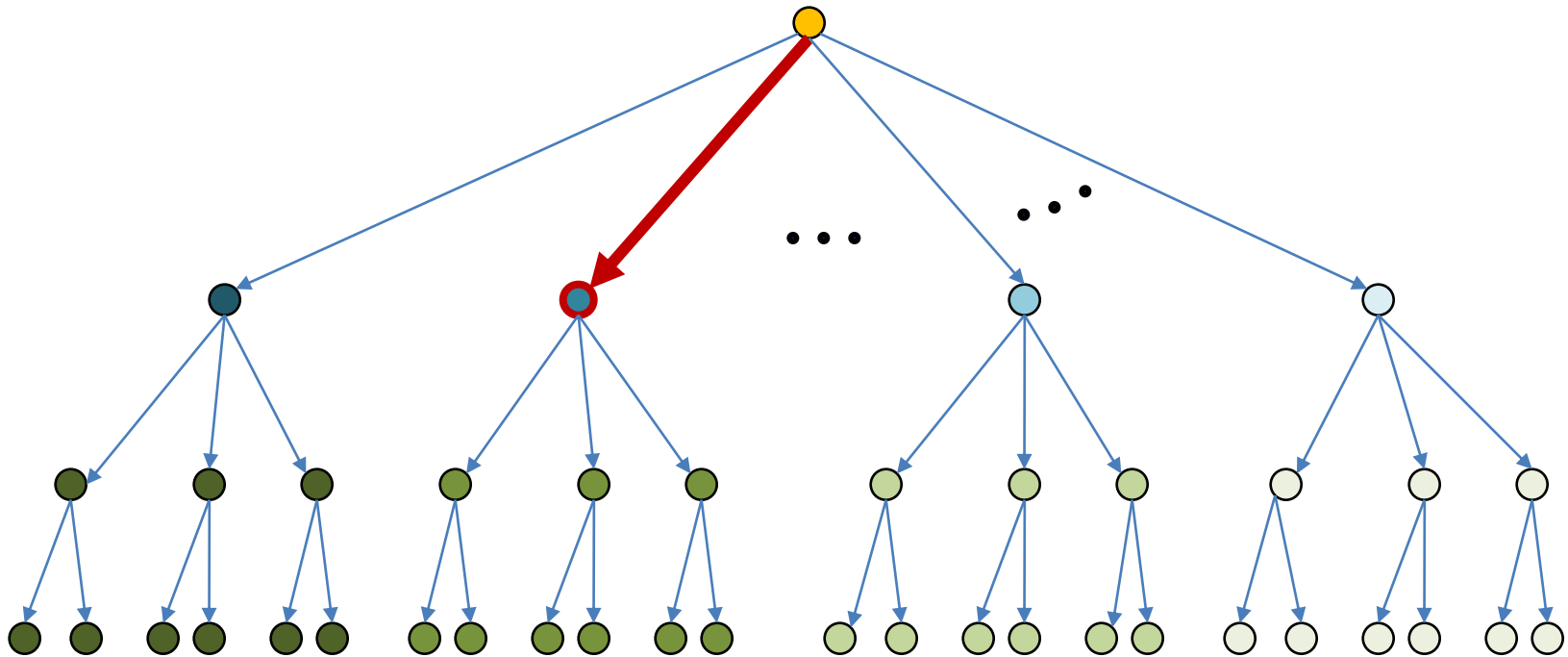
Build a first-stage solution on the union of these scenarios.

When you get the actual scenario,
augment the solution in the obvious way.

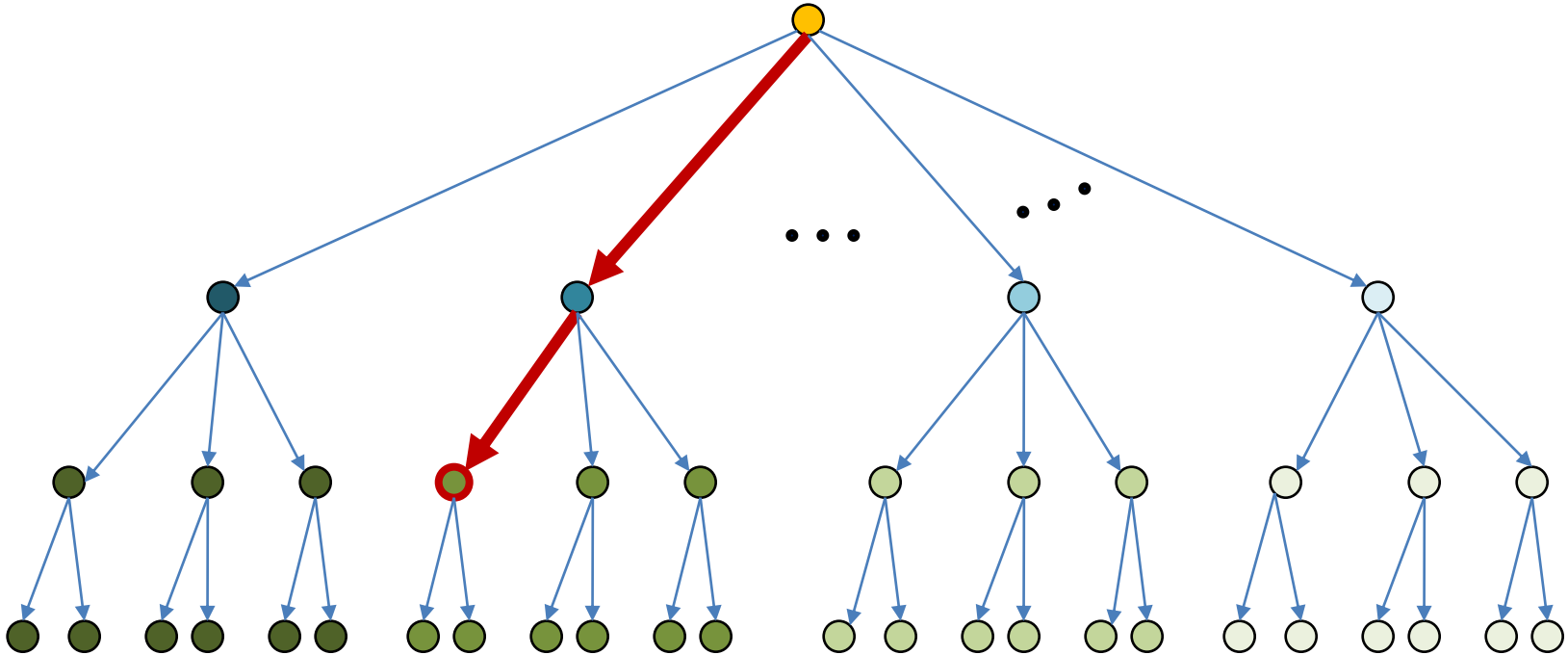
Say, the cost of every edge increases by a factor of 2 at every stage.



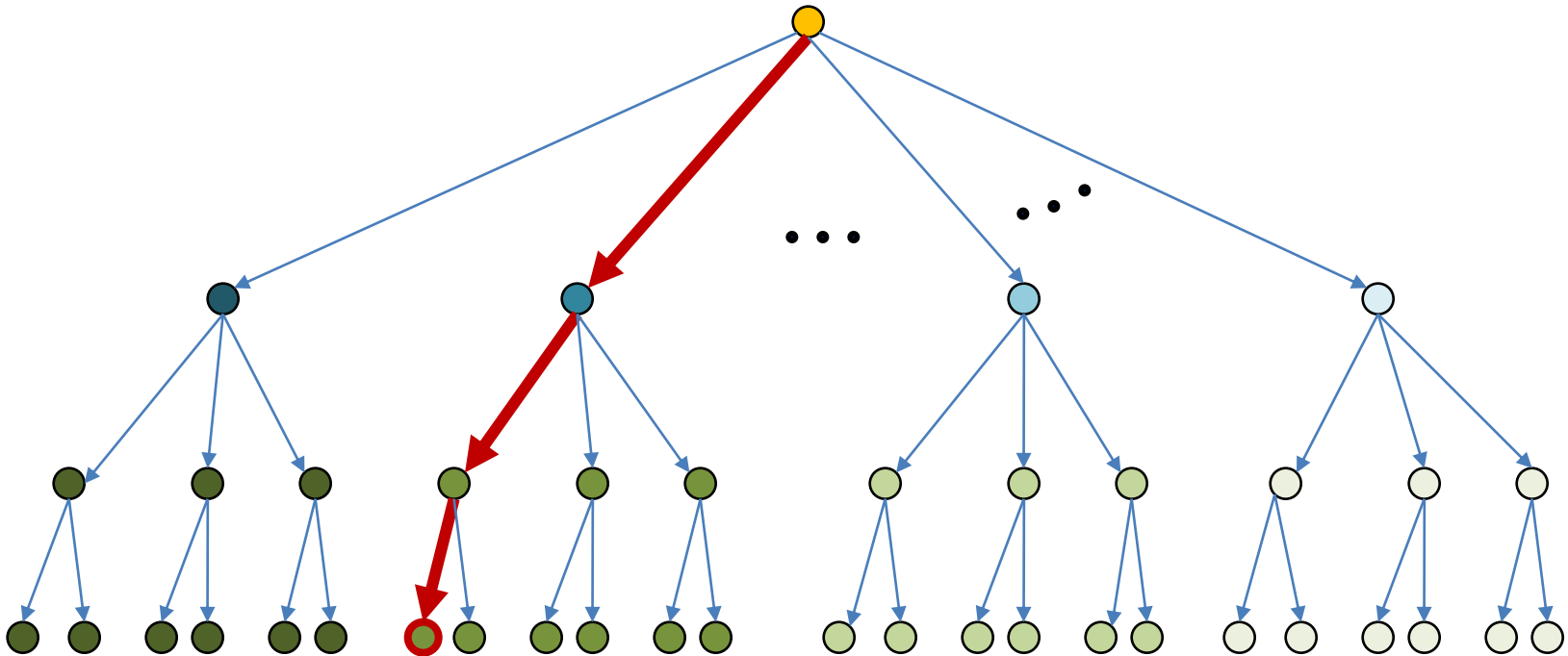
Say, the cost of every edge increases by a factor of 2 at every stage.



Say, the cost of every edge increases by a factor of 2 at every stage.



Say, the cost of every edge increases by a factor of 2 at every stage.



Theorem: k -stage boosted-sampling is a 2^k -approximation for Steiner tree.

**“infinite horizon” problems:
stochastic meets online**

an extension

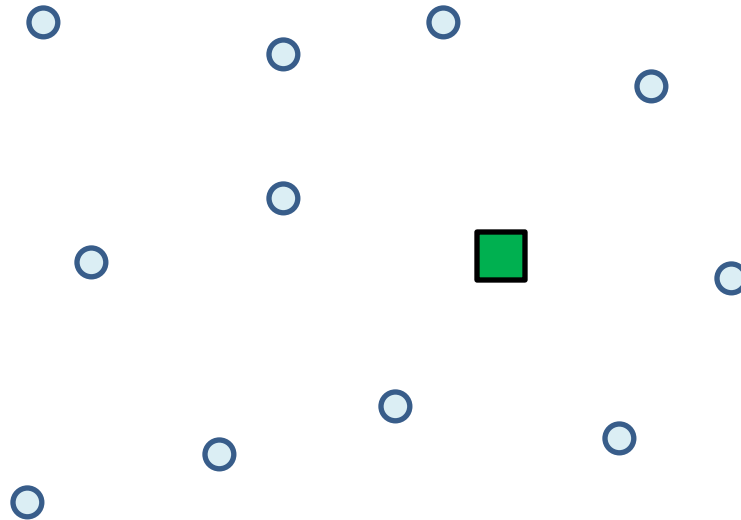
Stochastic analysis of online algorithms

Online algorithms try to model uncertainty about the future
in a different way

What if the adversary in online algorithms is replaced by a
probability distribution?

online Steiner tree

Given: graph G , edge costs $c: E \rightarrow \mathbb{R}$, root vertex r .



Online Steiner Tree

Given: graph G , edge costs $c: E \rightarrow \mathbb{R}$, root vertex r .

Competitive ratio of algorithm A :

$$\max_{\sigma} \frac{\text{Cost}(\text{algorithm } A \text{ on sequence } \sigma)}{\text{Cost}(\text{optimum Steiner tree on } \sigma)}$$

The Greedy Algorithm

[Imase Waxman '91] The greedy algorithm is $O(\log k)$ competitive for sequences of length k .

The Greedy Algorithm

[Imase Waxman '91] The greedy algorithm is $O(\log k)$ competitive for sequences of length k .

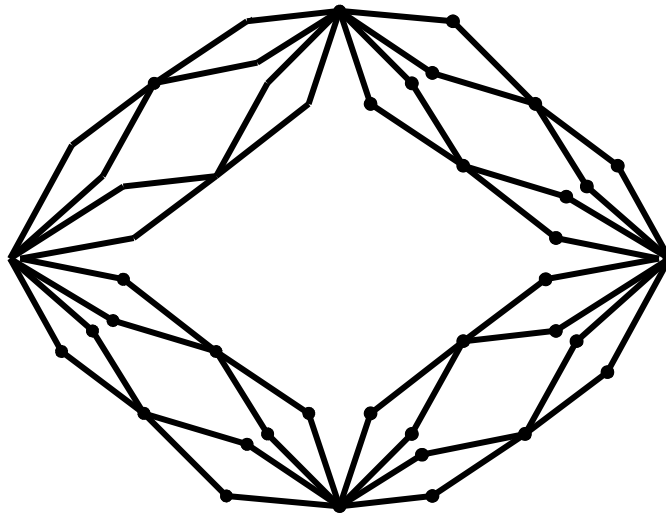
And this is tight.



The Greedy Algorithm

[Imase Waxman '91] The greedy algorithm is $O(\log k)$ competitive for sequences of length k .

And this is tight.



The Stochastic Models

Suppose demands are nodes in V drawn **uniformly** at random, **independently** of previous demands.

uniformity: not important

could have probability $p: V \rightarrow [0,1]$

but algorithm must be given this distribution...

independence: important

The Stochastic Models

Suppose demands are nodes in V drawn **uniformly** at random, **independently** of previous demands.

Suppose we know the length **k** of random sequence.

$$\begin{array}{l} \text{minimize} \\ \text{algo A} \end{array} \frac{\mathbf{E}_{\sigma \text{ in } pk} [\text{cost of algorithm A on } \sigma]}{\mathbf{E}_{\sigma \text{ in } pk} [\text{OPT}(\text{set } \sigma)]}$$

“known-length model”

greed is bad

[Imase Waxman '91] The greedy algorithm is $O(\log k)$ competitive for sequences of length k .

Tight example holds also for (uniformly) random demands.



Some results₍₁₎

Theorem 1

For stochastic online Steiner tree, “augmented greedy” gets an $O(1)$ ratio for both known and unknown lengths.

Theorem 2

Similar results for online versions of “nice” subadditive problems (e.g., facility location, vertex cover).

The Stochastic Models

Suppose demands are nodes in V drawn **uniformly** at random, **independently** of previous demands.

Instead, we could ask for a different objective function:

$$\begin{array}{l} \text{minimize} \\ \text{algo A} \end{array} \mathbf{E}_{\sigma \text{ in } pk} \left(\frac{\text{cost of algorithm A on } \sigma}{\text{OPT}(\text{set } \sigma)} \right)$$

“expected ratio”

other directions being considered

Other directions

Demand-robust models:

In two-stage stochastic problems, want to minimize

$$\text{first-stage cost} + \mathbf{E}_{\text{scenarios}} [\text{second stage cost}]$$

What about:

$$\text{first-stage cost} + \mathbf{max}_{\text{scenarios}} [\text{second stage cost}]$$

Other directions₍₂₎

Demand-robust models:

$$\min \text{ first-stage cost } + \max_{\text{scenarios}} [\text{second stage cost}]$$

[Dhamdhere Goyal Ravi Singh, FOCS 2005]

gave constant-factor approximations for a similar set of problems – Steiner tree, facility location, ...

What can we say for other functions apart from sum and max?

Other directions₍₃₎

- Most ideas apply to covering problems like Steiner tree, Vertex cover, Set cover, facility location, some flow and cut problems...
- Stochastic inventory control problems
[Levi Pal Roundy Shmoys] use cost-balancing ideas to get constant-factor approximations.
- Stochastic matching problems
[Katriel Kenyon-Mathieu Upfal] study two-stage stochastic versions of matching problems, many hardness results.
- Stochastic and Robust p-median-type problems
[Anthony Goyal Gupta Nagarajan] give poly-log approximations for many p-median type problems.

Other directions₍₃₎

- Sample complexity

If each new sample incurs a cost, how to minimize the total cost?

- Life between expectation (stochastic) and worst-case (robust)

How do we keep the variance low?

What if the adversary is allowed to make small (adversarial) changes to the random instances?

Some preliminary results are known for these problems: big questions are still open.

thank you!