Approximation Algorithms for Stochastic Combinatorial Optimization

R. Ravi
Carnegie Mellon University

Joint work with:
Kedar Dhamdhere, CMU
Anupam Gupta, CMU
Martin Pal, DIMACS
Mohit Singh, CMU
Amitabh Sinha, U. Michigan

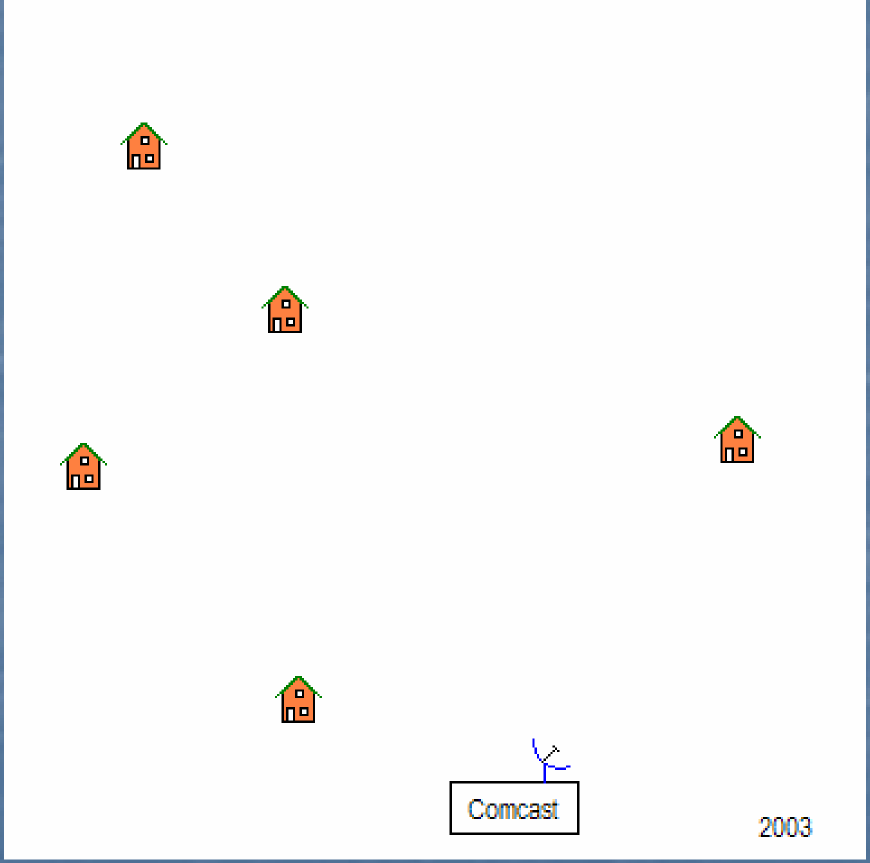
Sources: [RS IPCO 04, GPRS STOC 04, GRS FOCS 04, DRS IPCO 05]

Outline

- Motivation: The cable company problem
- Model and literature review
- Solution to the cable company problem
- General covering problem
- Scenario dependent cost model

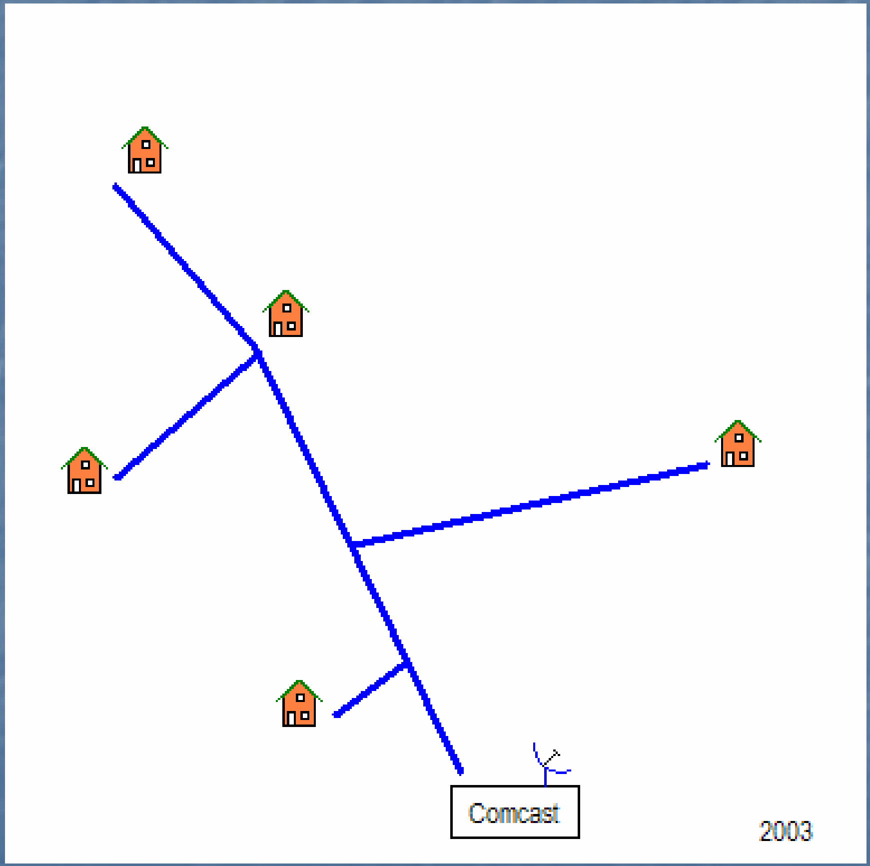
The cable company problem

- Cable company plans to enter a new area
- Currently, low population
- Wants to install cable infrastructure in anticipation of future demand



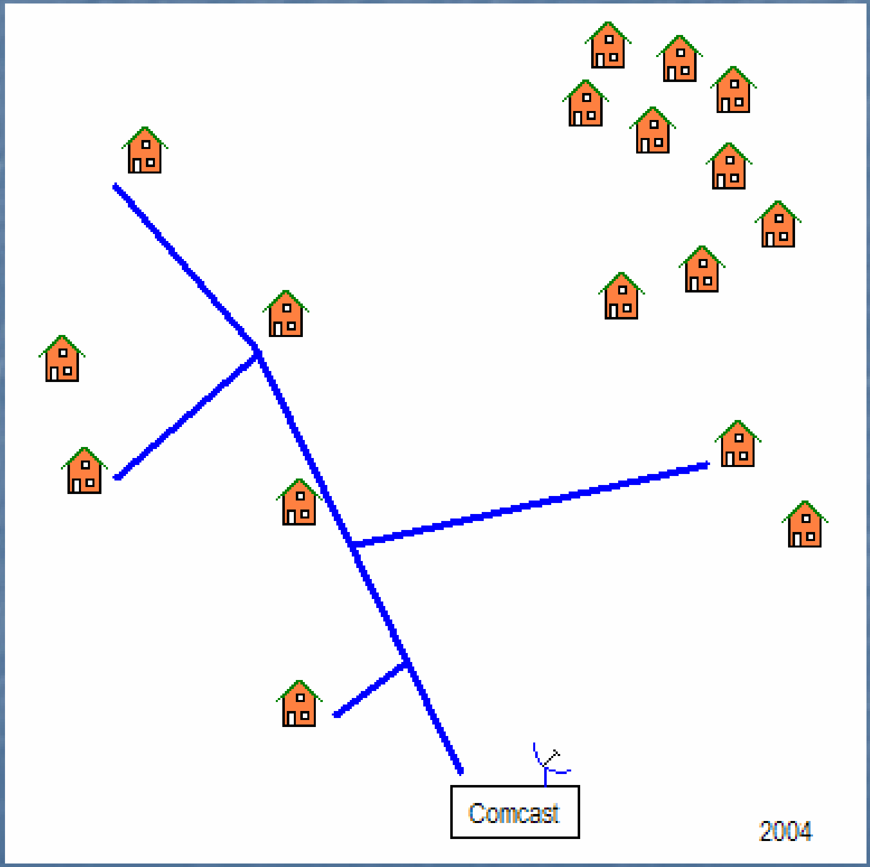
The cable company problem

- Future demand unknown, yet cable company needs to build now
- Where should cable company install cables?



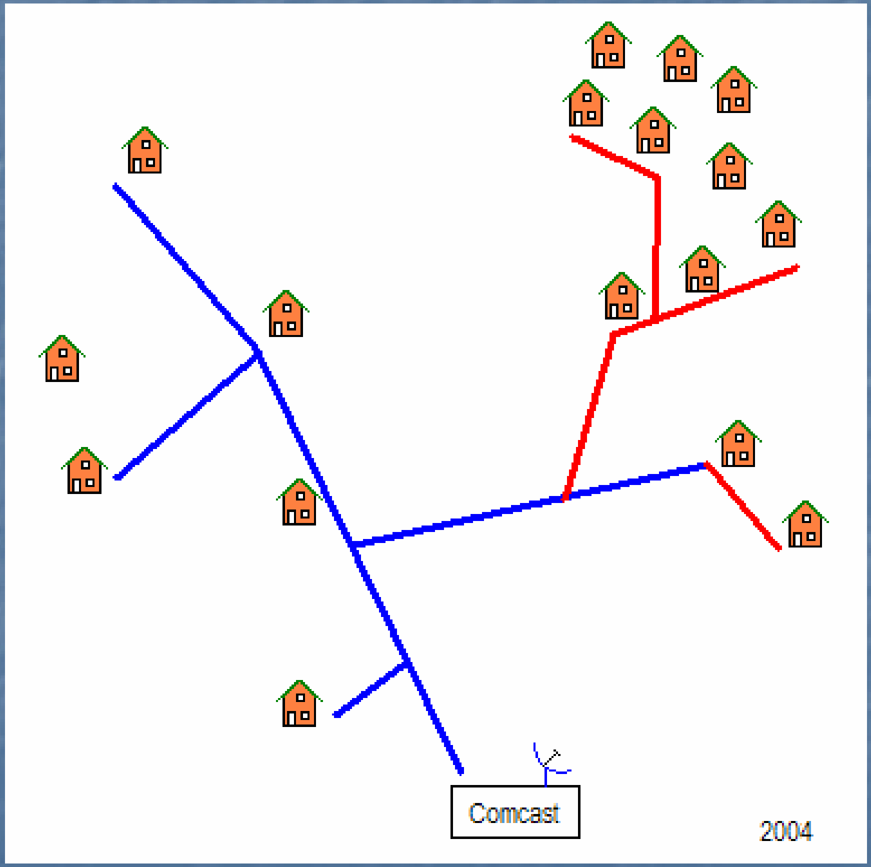
The cable company problem

- Future demand unknown, yet cable company needs to build now
- Where should cable company install cables?



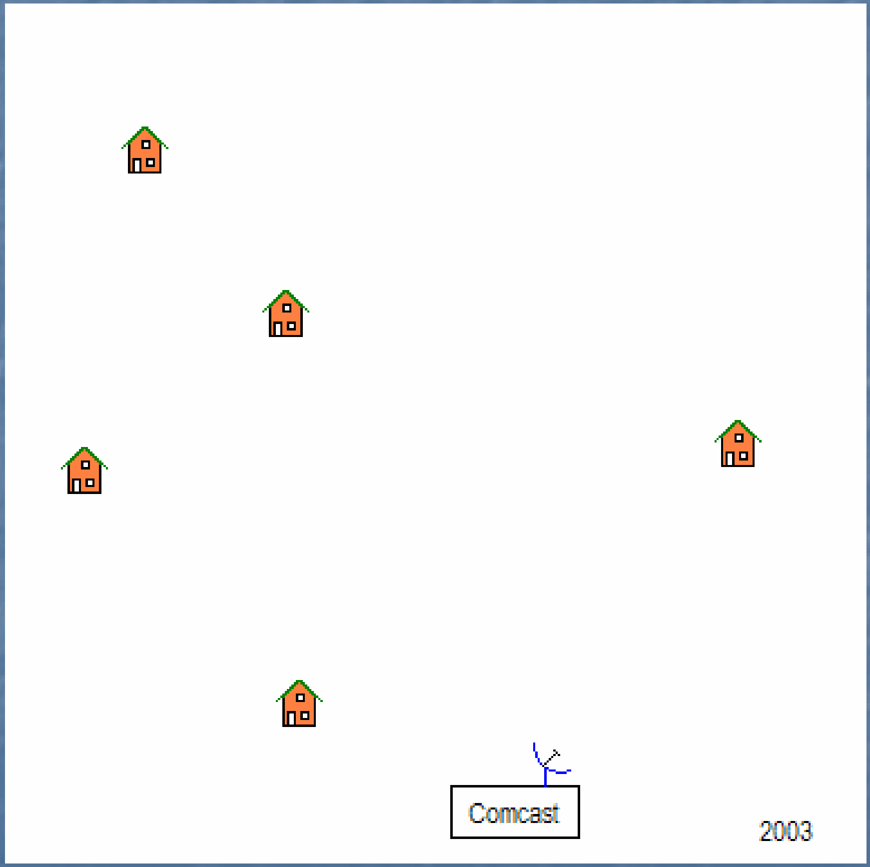
The cable company problem

- Future demand unknown, yet cable company needs to build now
- Where should cable company install cables?



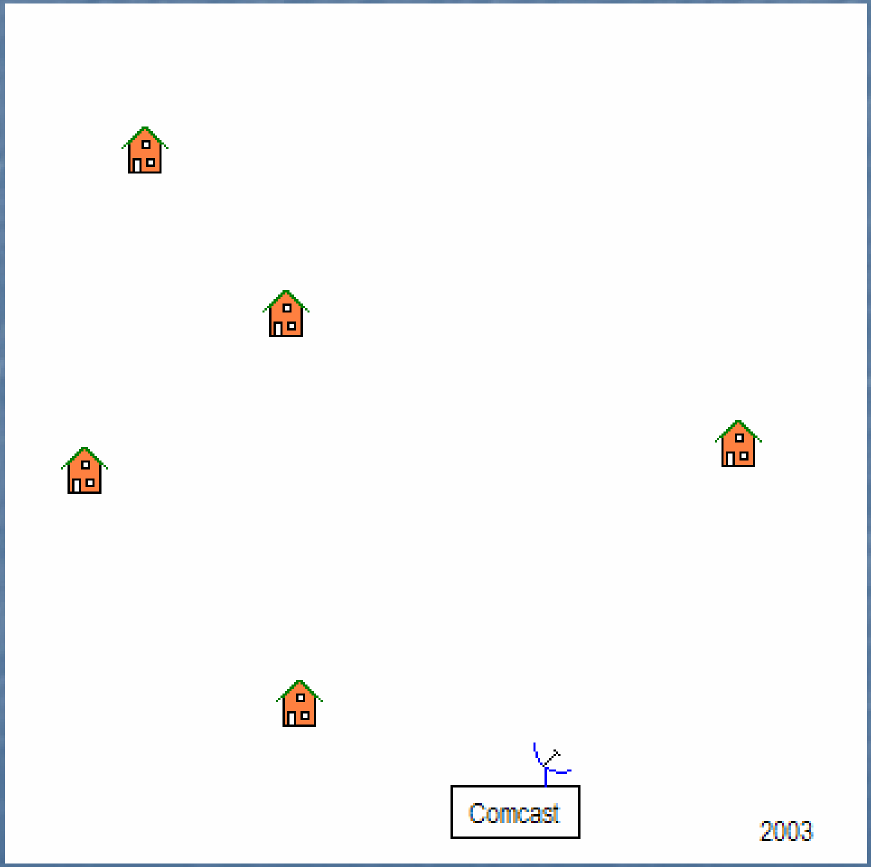
The cable company problem

- Future demand unknown, yet cable company needs to build now
- Where should cable company install cables?



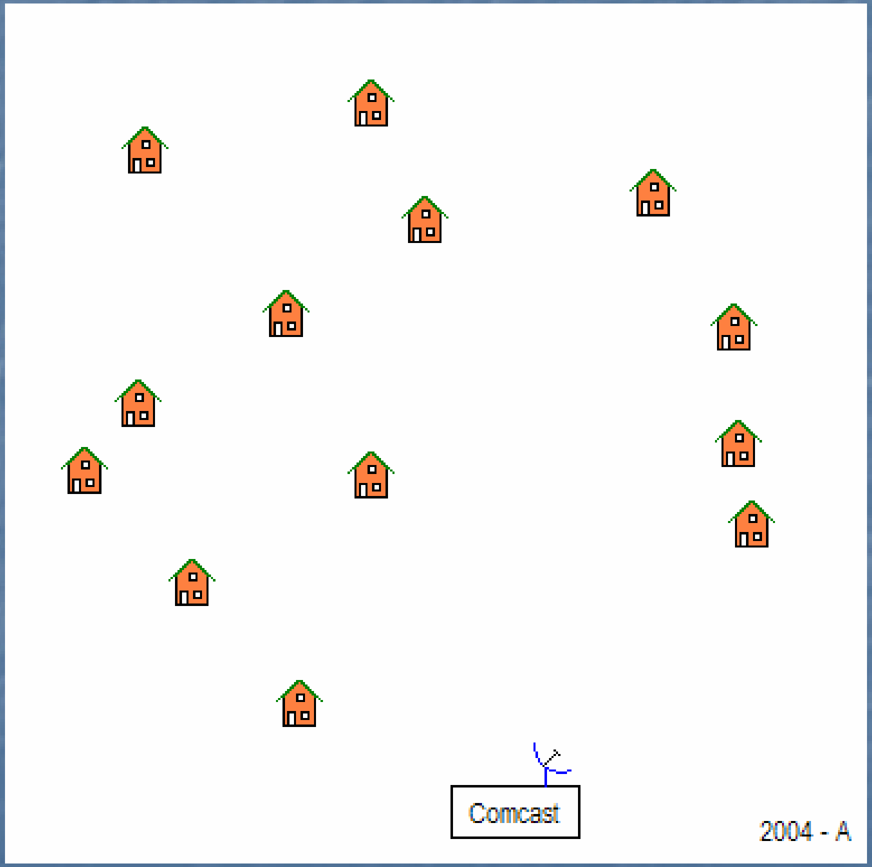
The cable company problem

- Future demand unknown, yet cable company needs to build now
- Forecasts of possible future demands exist
- Where should cable company install cables?



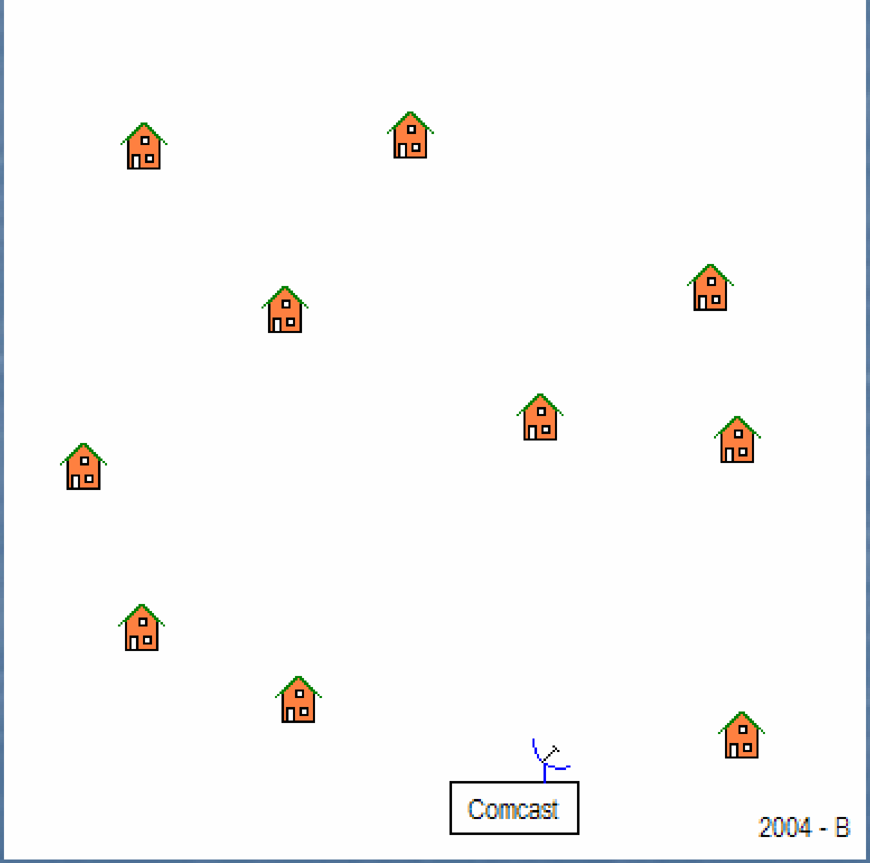
The cable company problem

- Future demand unknown, yet cable company needs to build now
- Forecasts of possible future demands exist
- Where should cable company install cables?



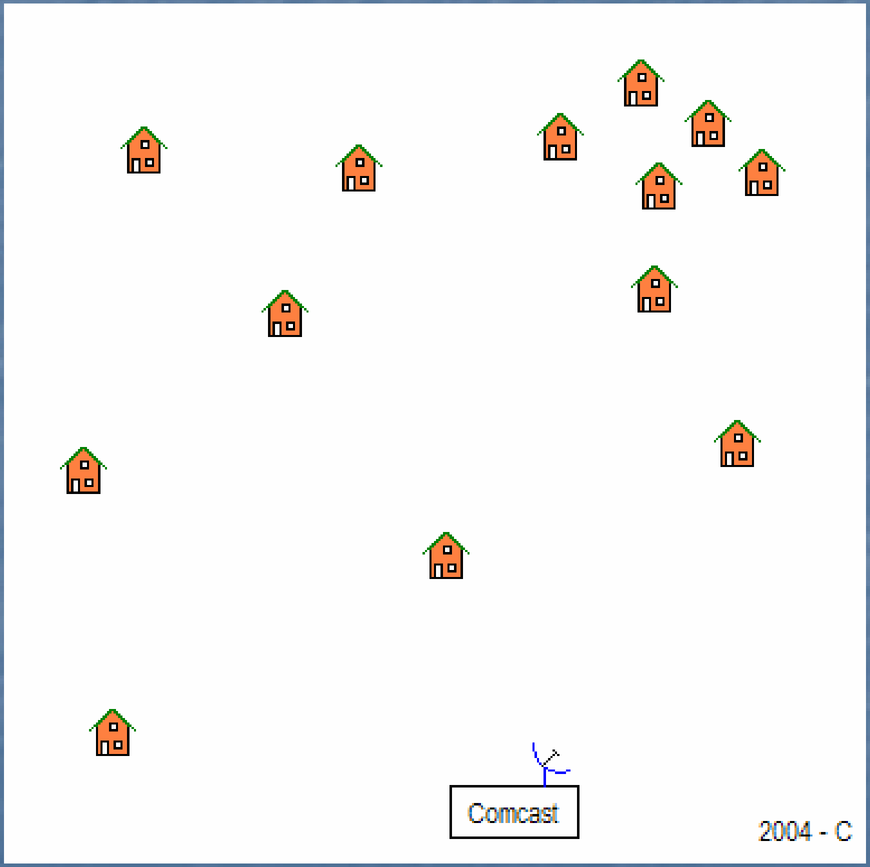
The cable company problem

- Future demand unknown, yet cable company needs to build now
- Forecasts of possible future demands exist
- Where should cable company install cables?



The cable company problem

- Future demand unknown, yet cable company needs to build now
- Forecasts of possible future demands exist
- Where should cable company install cables?



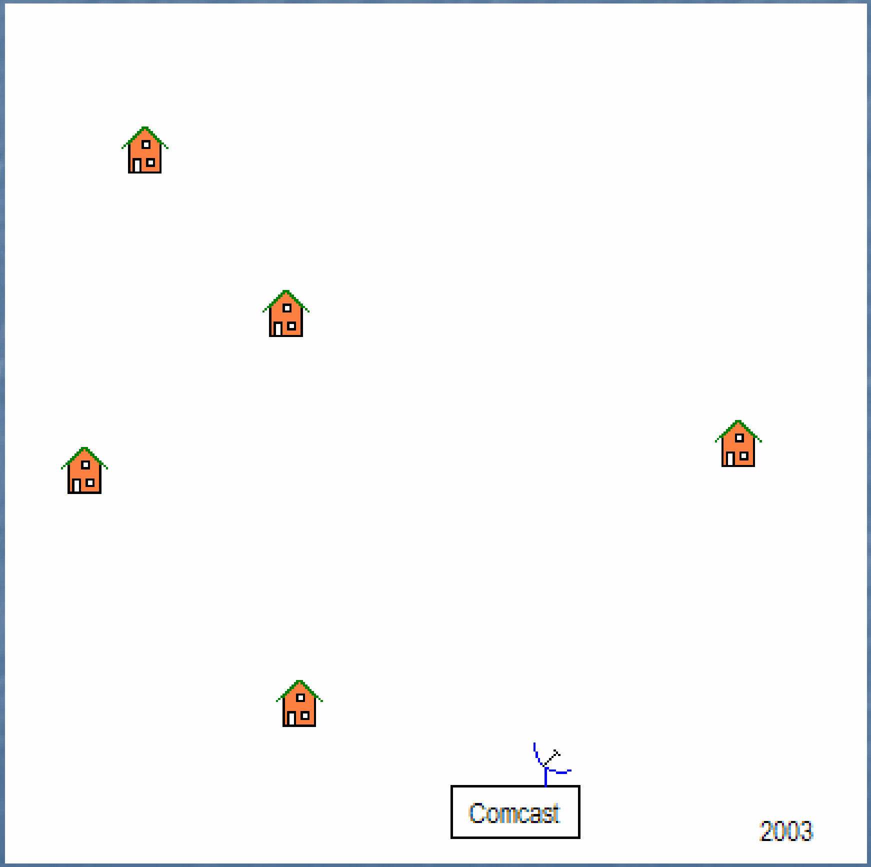
The cable company problem

- cable company wants to use demand forecasts, to

Minimize

Today's install. costs

+ Expected future costs



Outline

- Motivation: The cable company problem
- Model and literature review
- Solution to the cable company problem
- General covering problem
- Scenario dependent cost model

Stochastic optimization

- Classical optimization assumed deterministic inputs

Stochastic optimization

- Classical optimization assumed deterministic inputs
- Need for modeling data uncertainty quickly realized [Dantzig ‘55, Beale ‘61]

Stochastic optimization

- Classical optimization assumes deterministic inputs
- Need for modeling data uncertainty quickly realized [Dantzig '55, Beale '61]
- [Birge, Louveaux '97, Klein Haneveld, van der Vlerk '99]

Model

Two-stage stochastic opt. with recourse

Model

Two-stage stochastic opt. with recourse

- Two stages of decision making, with limited information in first stage

Model

Two-stage stochastic opt. with recourse

- Two stages of decision making
- Probability distribution governing second-stage data and costs given in 1st stage

Model

Two-stage stochastic opt. with recourse

- Two stages of decision making
- Probability dist. governing data and costs
- Solution can always be made feasible in second stage

Mathematical model

- Ω: probability space of 2nd stage data

$$
\begin{array}{llllll}
\min & c.x & + E[c(\omega)y(\omega)] & & & \\
A.x & & & \leq & b & \\
T(\omega).x & + & W(\omega).y(\omega) & \leq & h(\omega) & \forall \omega \in \Omega \\
x \in P, & & y(\omega) \in P(\omega) & & &
\end{array}
$$

Mathematical model

- Ω: probability space of 2nd stage data

$$
\begin{array}{lllll}
\min & c.x & + E[c(\omega)y(\omega)] & & \\
A.x & & \leq & b & \\
T(\omega).x & + & W(\omega).y(\omega) \leq & h(\omega) & \forall \omega \in \Omega \\
x \in P, & & y(\omega) \in P(\omega) & &
\end{array}
$$

- Extensive form: Enumerate over all $\omega \in \Omega$

Scenario models

- Enumerating over all $\omega \in \Omega$ may lead to very large problem size
- Enumeration (or even approximation) may not be possible for continuous domains

New model: Sampling Access

- "Black box" available which generates a sample of 2nd stage data with same distribution as actual 2nd stage

- Bare minimum requirement on model of stochastic process

Computational complexity

- Stochastic optimization problems solved using Mixed Integer Program formulations
- Solution times prohibitive
- NP-hardness inherent to problem, not formulation: E.g., 2-stage stochastic versions of MST, Shortest paths are NP-hard.

Our goal

- Approximation algorithm using sampling access
  - cable company problem
  - General model – extensions to other problems

Our goal

- Approximation algorithm using sampling access
  - cable company problem
  - (General model – extensions to other problems)
- Consequences
  - Provable guarantees on solution quality
  - Minimal requirements of stochastic process

Previous work

- Scheduling with stochastic data
  - Substantial work on exact algorithms [Pinedo ’95]
  - Some recent approximation algorithms [Goel, Indyk ’99; Möhring, Schulz, Uetz ’99]
- Approximation algorithms for stochastic models
  - Resource provisioning with polynomial scenarios [Dye, Stougie, Tomasgard Nav. Res. Qtrly ’03]
  - ”Maybecast” Steiner tree: $O(\log n)$ approximation when terminals activate independently [Immorlica, Karger, Minkoff, Mirrokni ’04]

Our work

- Approximation algorithms for two-stage stochastic combinatorial optimization
  - Polynomial Scenarios model, several problems using LP rounding, incl. Vertex Cover, Facility Location, Shortest paths [R., Sinha, July ’03, appeared IPCO ’04]
  - Black-box model: Boosted sampling algorithm for covering problems with subadditivity – general approximation algorithm [Gupta, Pal, R., Sinha STOC ’04]
  - Steiner trees and network design problems: Polynomial scenarios model, Combination of LP rounding and Primal-Dual [Gupta, R., Sinha FOCS ’04]
  - Stochastic MSTs under scenario model and Black-box model with polynomially bounded cost inflations [Dhamdhere, R., Singh, To appear, IPCO ’05]

Related work

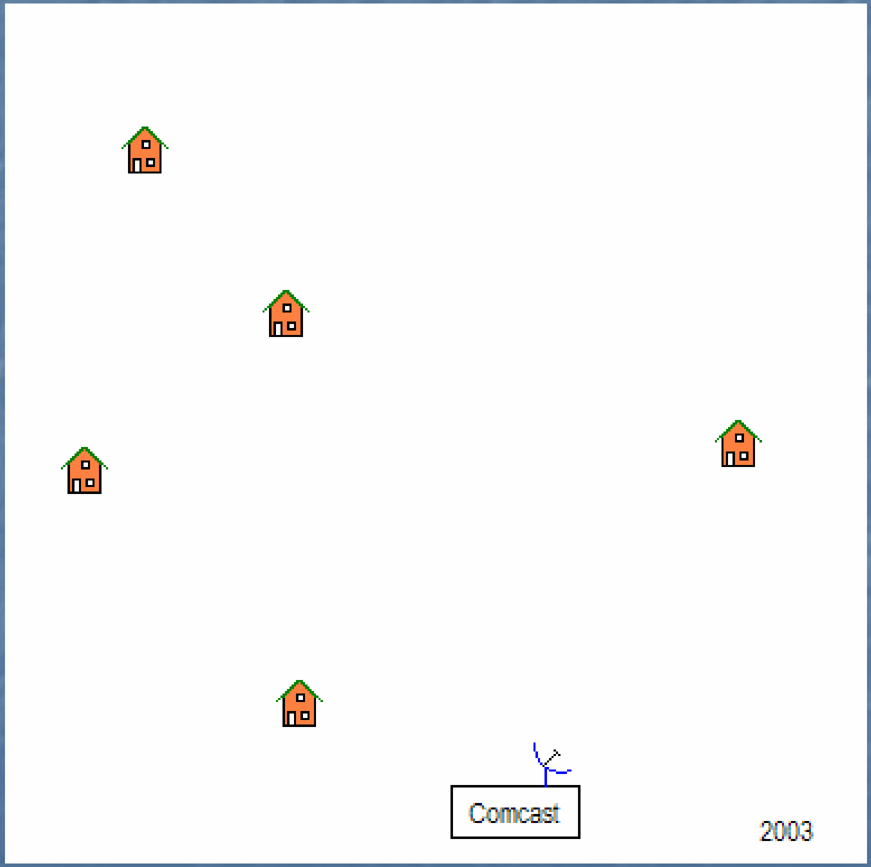
- Approximation algorithms for Stochastic Combinatorial Problems
  - Vertex cover and Steiner trees in restricted models studied by [Immorlica, Karger, Minkoff, Mirrokni SODA '04]
  - Rounding for stochastic Set Cover, FPRAS for #P hard Stochastic Set Cover LPs [Shmoys, Swamy FOCS '04]
  - Multi-stage stochastic Steiner trees [Hayrapetyan, Swamy, Tardos SODA '05]
  - Multi-stage Stochastic Set Cover [Shmoys, Swamy, manuscript '04]
  - Multi-stage black box model – Extension of Boosted sampling with rejection [Gupta, Pal, R., Sinha manuscript '05]

Outline

- Motivation: The cable company problem
- Model and literature review
- Solution to the cable company problem
- General covering problem
- Scenario dependent cost model

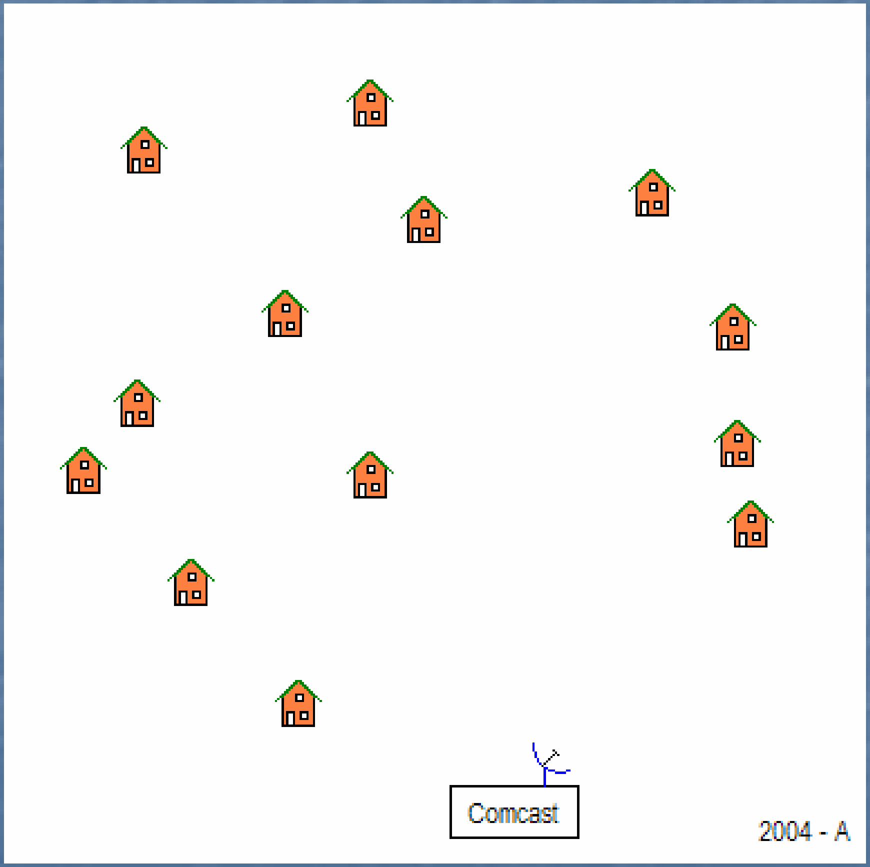
The cable company problem

- Cable company wants to install cables to serve future demand



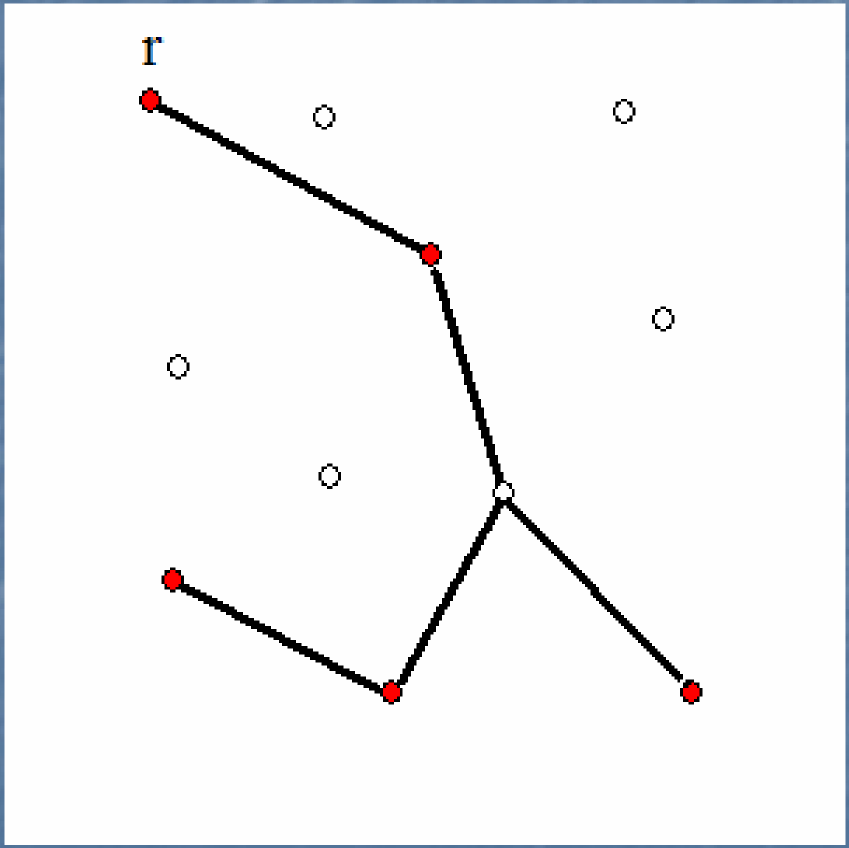
The cable company problem

- Cable company wants to install cables to serve future demand
- Future demand stochastic, cables get expensive next year
- What cables to install this year?



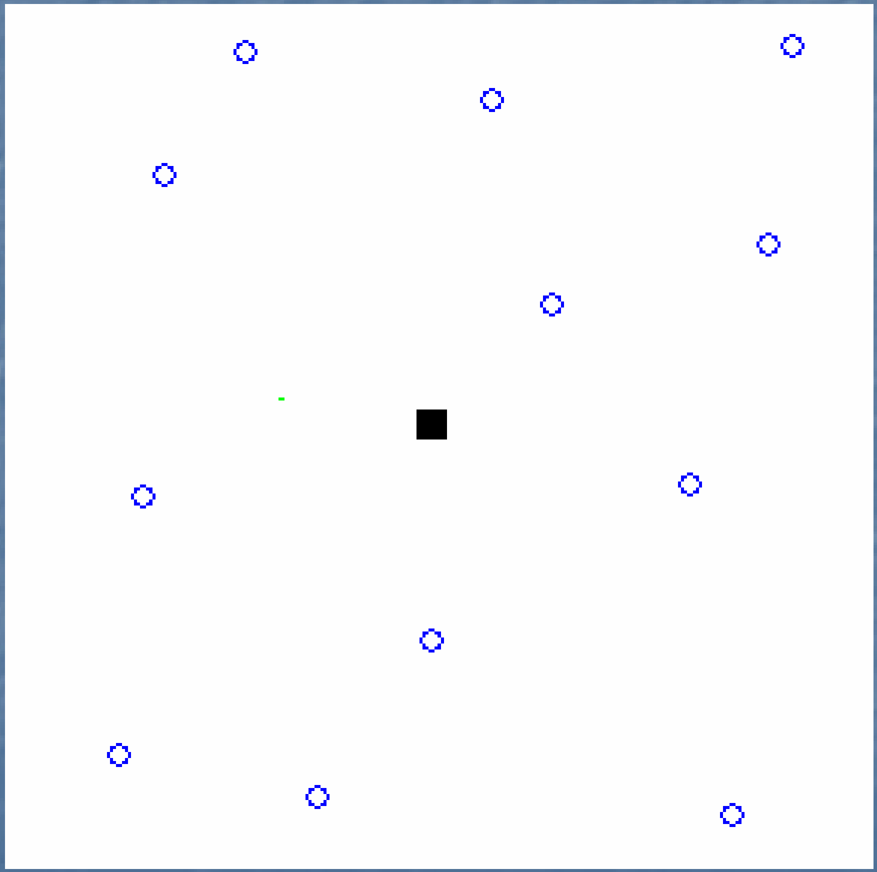
Steiner Tree - Background

- Graph *G=(V,E,c)*
- Terminals *S*, root $r \in S$
- Steiner tree: Min cost tree spanning *S*
- NP-hard, MST is a 2-approx, Current best 1.55-approx (Robins, Zelikovsky '99)
- Primal-dual 2-approx (Agrawal, Klein, R. '91; Goemans, Williamson '92)



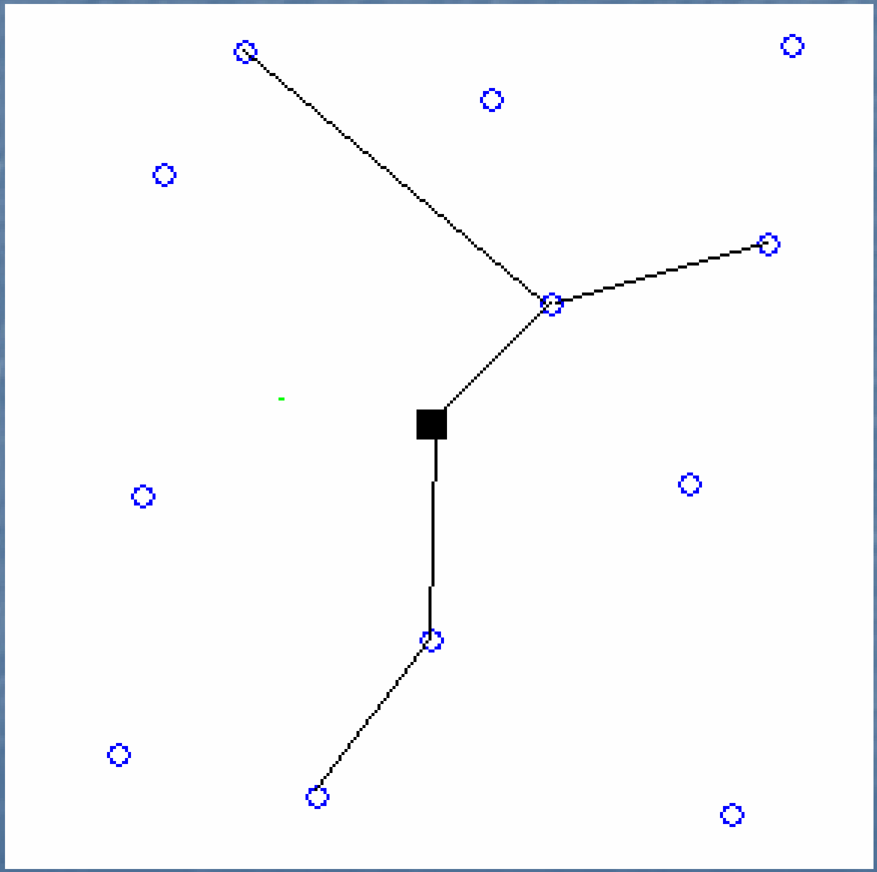
Stochastic Min. Steiner Tree

- Given a metric space of points, distances c_e
- Points: possible locations of future demand
- Wlog, simplifying assumption: no 1st stage demand



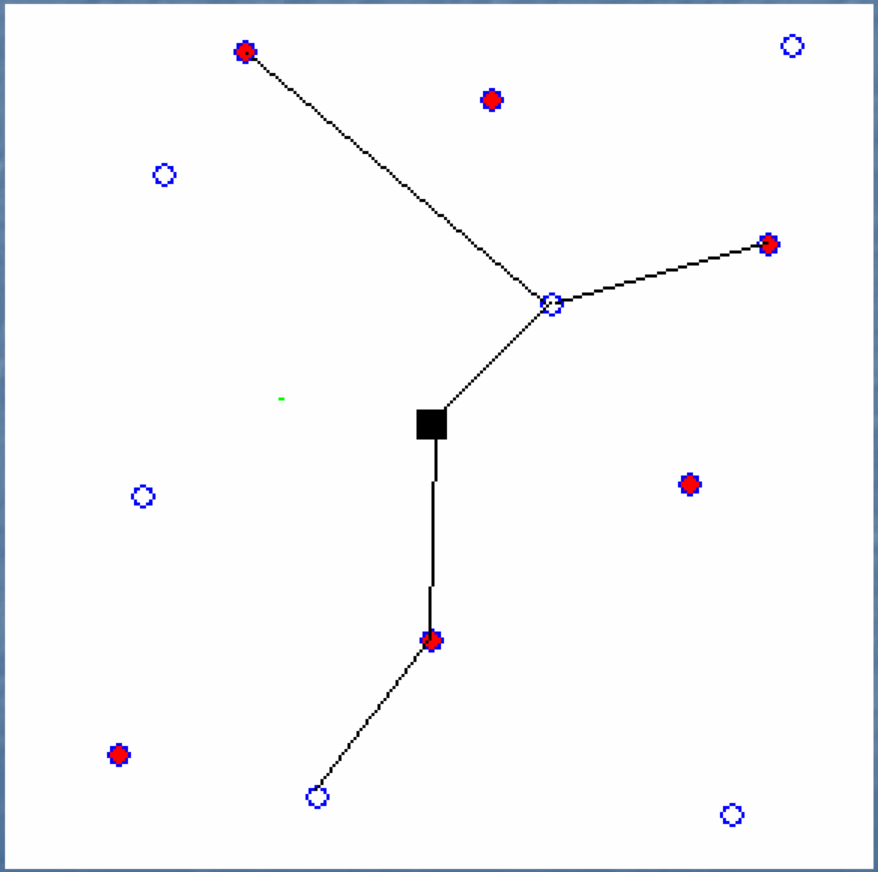
Stochastic Min. Steiner Tree

- Given a metric space of points, distances c_e
- 1st stage: buy edges at costs c_e



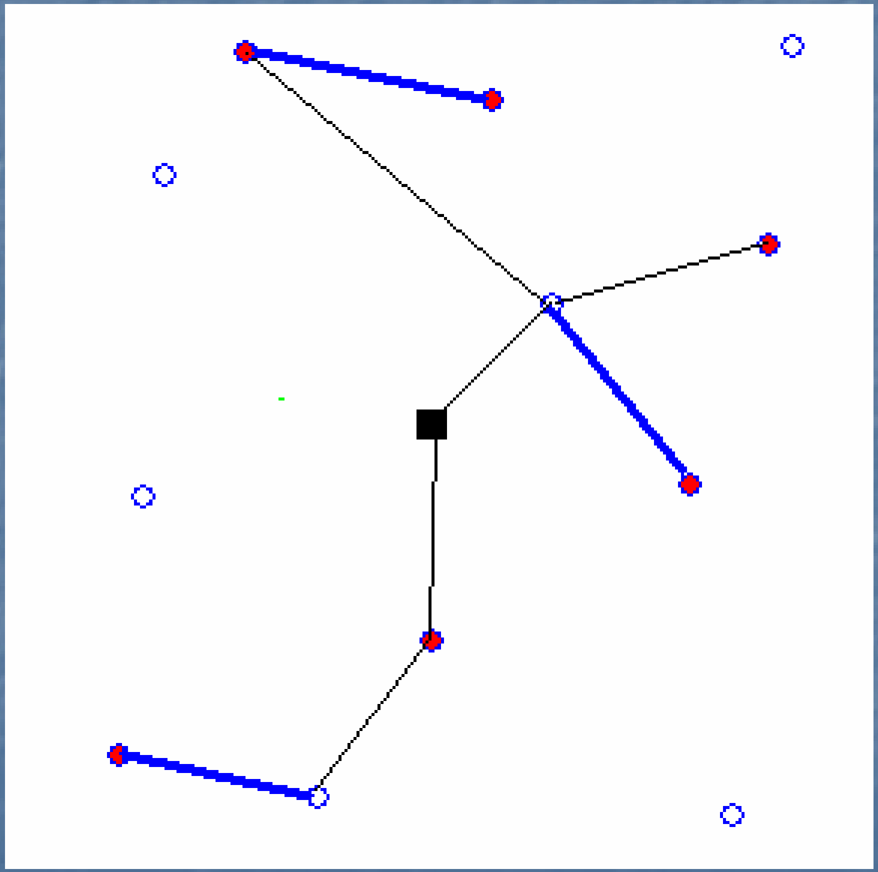
Stochastic Min. Steiner Tree

- Given a metric space of points, distances c_e
- 1st stage: buy edges at costs c_e
- 2nd stage: Some clients "realized", buy edges at cost $\sigma.c_e$ to serve them ($\sigma > 1$)



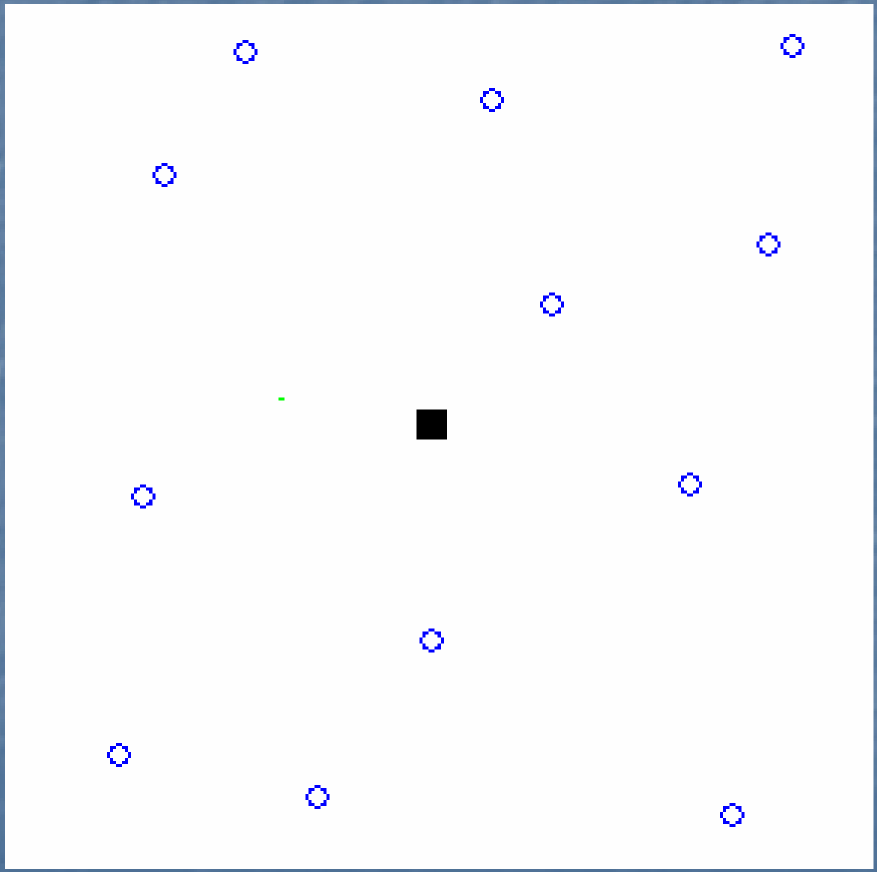
Stochastic Min. Steiner Tree

- Given a metric space of points, distances c_e
- 1st stage: buy edges at costs c_e
- 2nd stage: Some clients "realized", buy edges at cost $\sigma.c_e$ to serve them ($\sigma > 1$)



Stochastic Min. Steiner Tree

- Given a metric space of points, distances c_e
- 1st stage: buy edges at costs c_e
- 2nd stage: Some clients "realized", buy edges at cost $\sigma.c_e$ to serve them ($\sigma > 1$)
- Minimize exp. cost



Algorithm Boosted-Sample

- Sample from the distribution of clients σ times (sampled set S)

Algorithm Boosted-Sample

- Sample from the distribution of clients σ times (sampled set S)
- Build minimum spanning tree T_0 on S
  - Recall: Minimum spanning tree is a 2-approximation to Minimum Steiner tree

Algorithm Boosted-Sample

- Sample from the distribution of clients σ times (sampled set S)
- Build minimum spanning tree T_0 on S
- 2nd stage: actual client set realized (R)

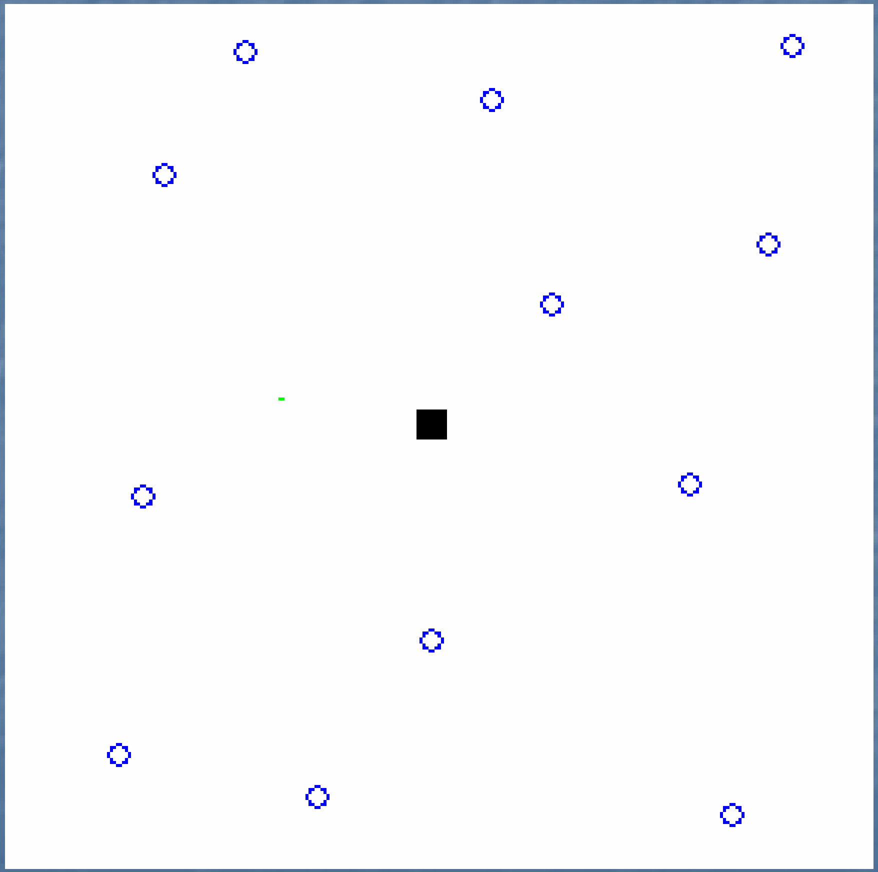
  \- Extend T_0 to span R

Algorithm Boosted-Sample

- Sample from the distribution of clients σ times (sampled set S)
- Build minimum spanning tree T_0 on S
- 2nd stage: actual client set realized (R)
  - Extend T_0 to span R
- Theorem: 4-approximation!

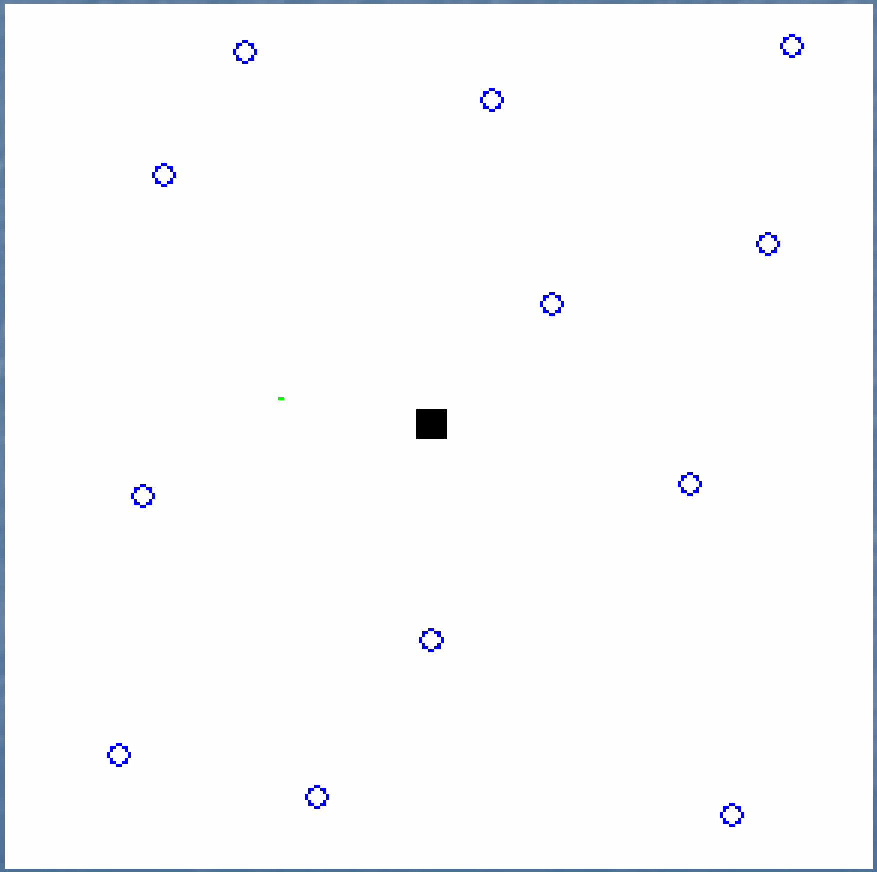
Algorithm: Illustration

- Input, with *σ=3*



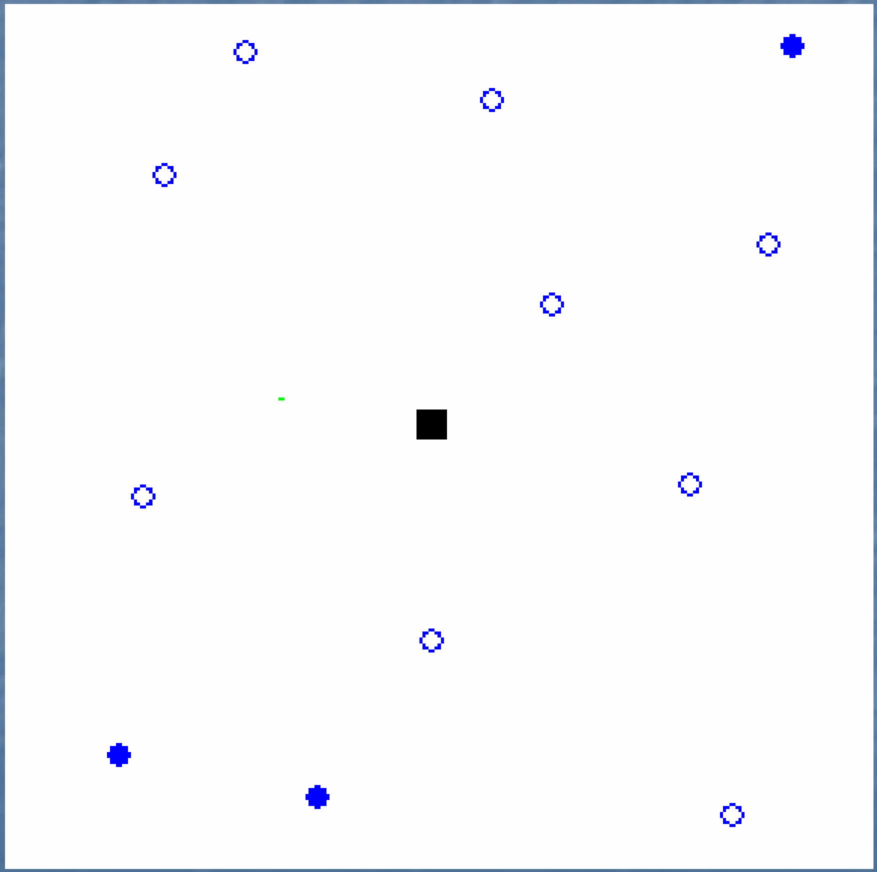
Algorithm: Illustration

- Input, with *σ=3*
- Sample *σ* times from client distribution



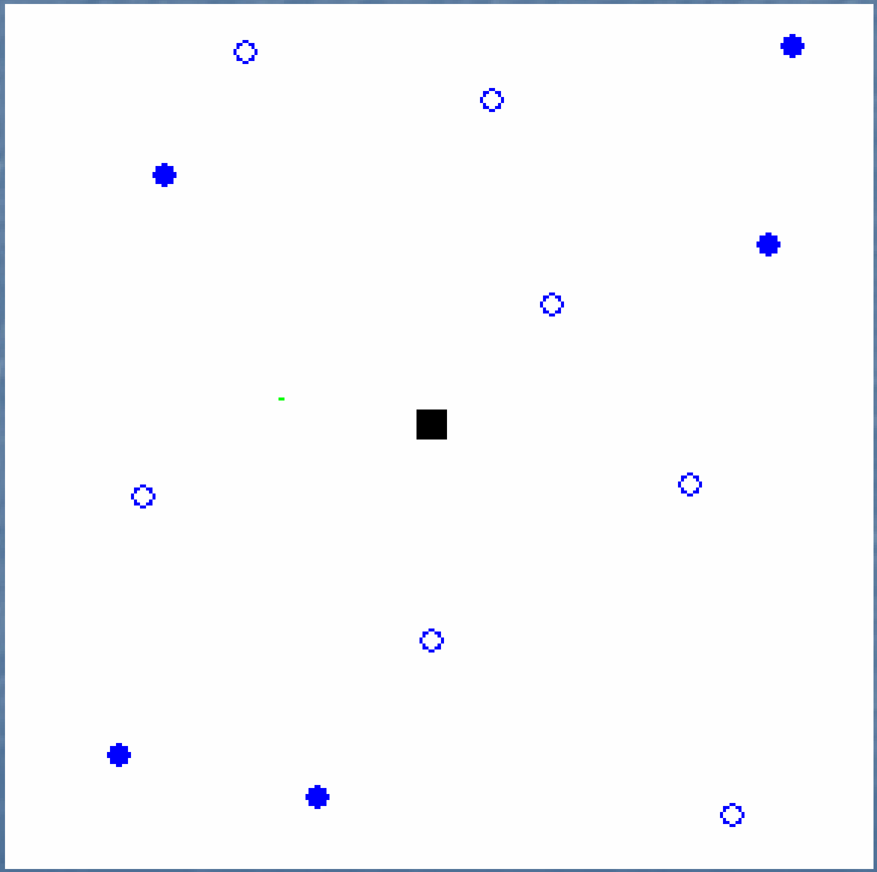
Algorithm: Illustration

- Input, with *σ=3*
- Sample σ times from client distribution



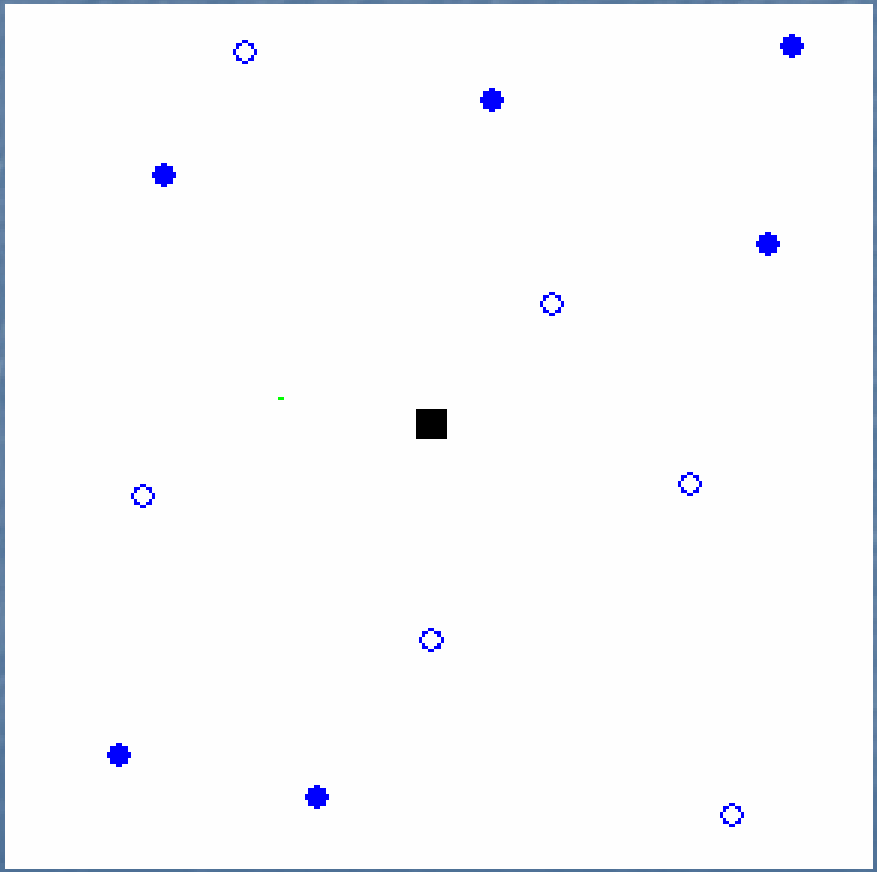
Algorithm: Illustration

- Input, with $\sigma=3$
- Sample σ times from client distribution



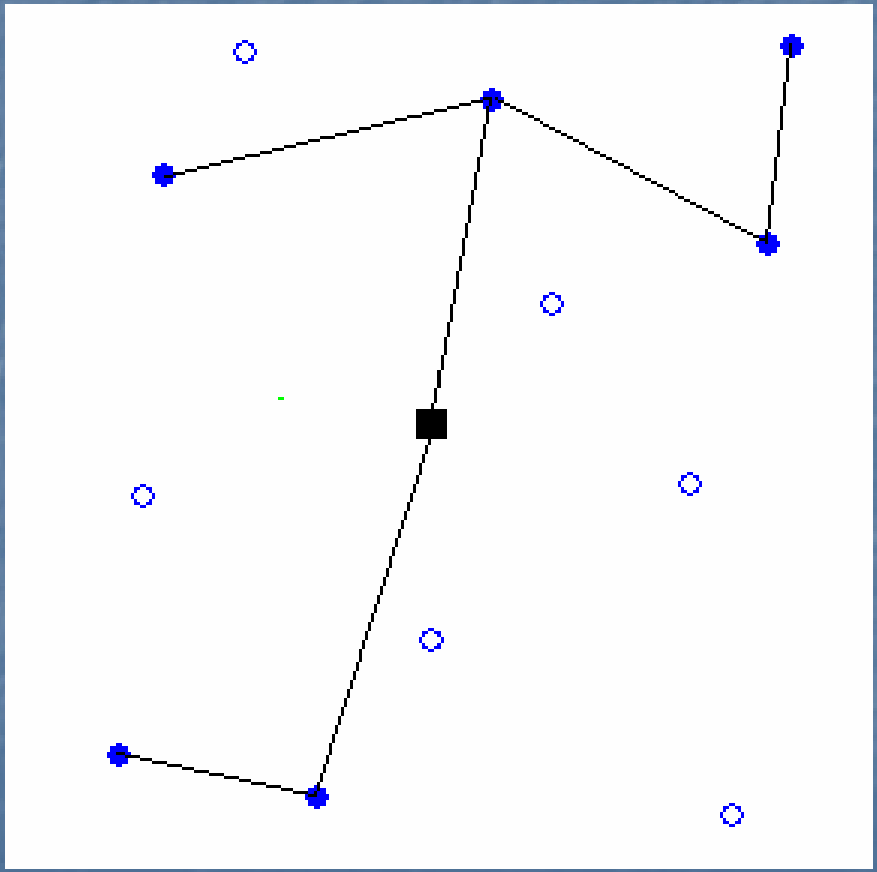
Algorithm: Illustration

- Input, with *σ=3*
- Sample σ times from client distribution



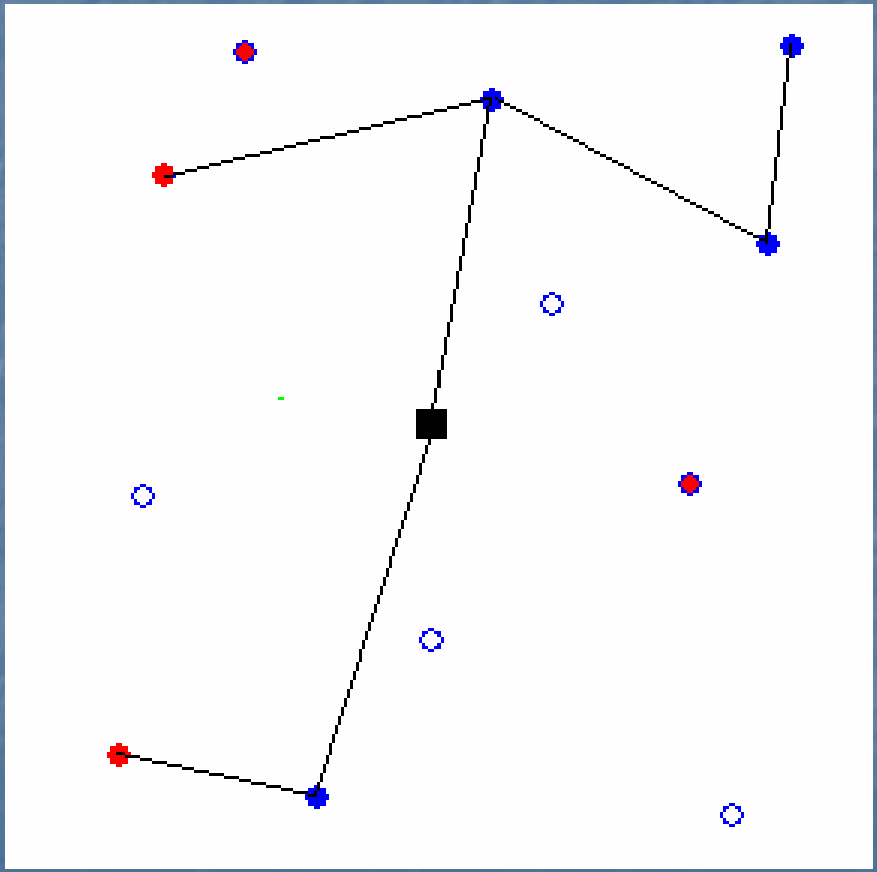
Algorithm: Illustration

- Input, with $\sigma=3$
- Sample σ times from client distribution
- Build MST T_0 on S



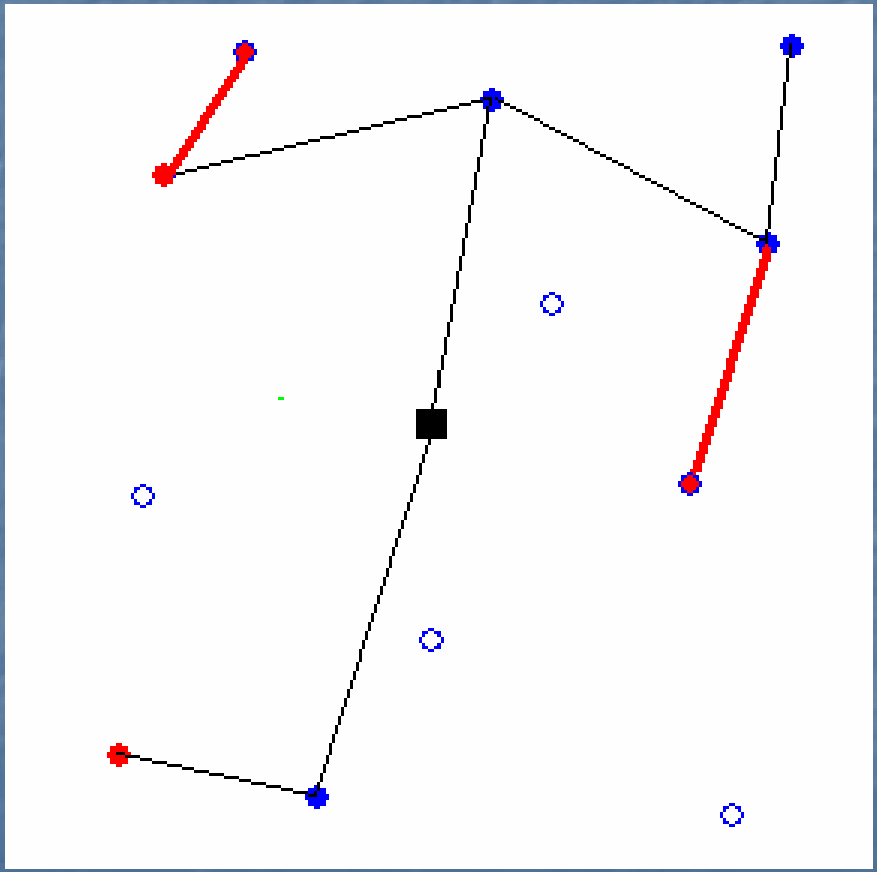
Algorithm: Illustration

- Input, with σ*=3*
- Sample σ times from client distribution
- Build MST T_0 on S
- When actual scenario (R) is realized …



Algorithm: Illustration

- Input, with $\sigma=3$
- Sample σ times from client distribution
- Build MST T_0 on S
- When actual scenario (R) is realized …
- Extend T_0 to span R



Analysis of 1st stage cost

- Let $OPT = c(T_0^*) + \sum_X p_X \sigma c(T_X^*)$

Analysis of 1st stage cost

- Let $OPT = c(T_0^*) + \sum_X p_X \sigma c(T_X^*)$
- Claim: $E[c(T_0)] \leq 2.OPT$

Analysis of 1st stage cost

- Let $OPT = c(T_0^*) + \sum_X p_X \sigma c(T_X^*)$
- Claim:

$$E[c(T_0)] \le 2.OPT$$

- Our *σ* samples: *S={S₁,* *S₂, …, S_σ}*

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

Analysis of 1st stage cost

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

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

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

Analysis of 1st stage cost

- Let $OPT = c(T_0^*) + \sum_X p_X \sigma c(T_X^*)$
- Claim:

$$E[c(T_0)] \le 2.OPT$$

- Our σ samples: *S={S₁, S₂, …, S_σ}*

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

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

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

Analysis of 2nd stage cost

- Intuition:
  - 1st stage: σ samples at cost c_e
  - 2nd stage: *1* sample at cost $\sigma.c_e$

Analysis of 2nd stage cost

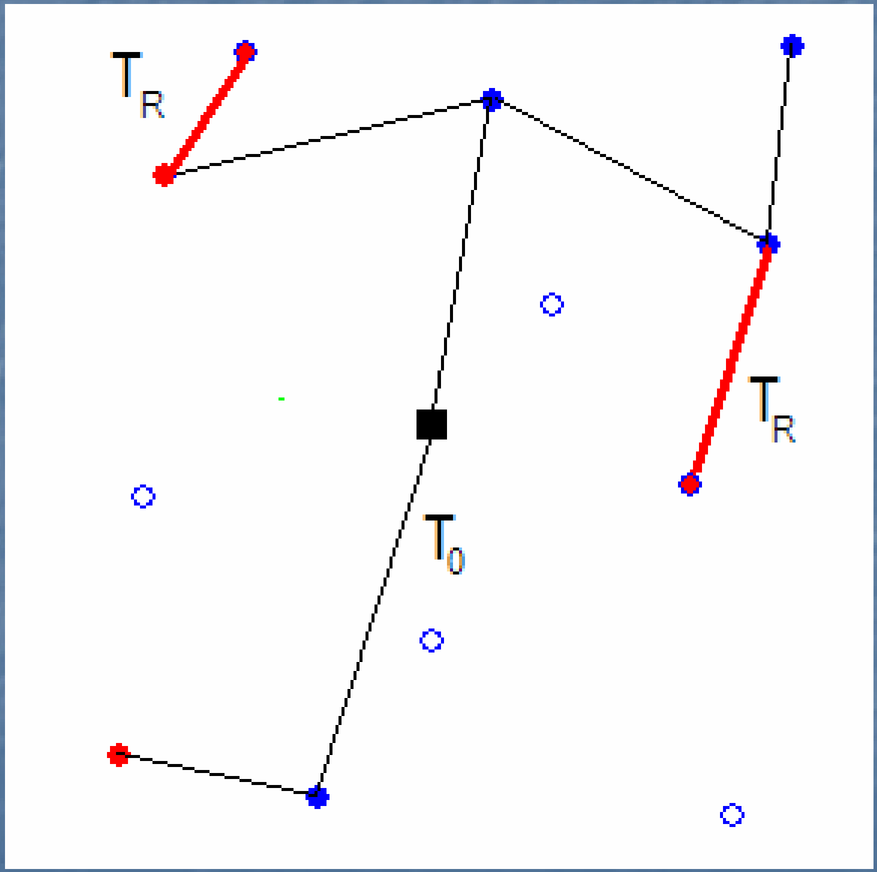
- Intuition:
  - 1st stage: σ samples at cost c_e
  - 2nd stage: *1* sample at cost $\sigma.c_e$
- In expectation,
  - 2nd stage cost $\leq$ 1st stage cost

Analysis of 2nd stage cost

- Intuition:
  - 1st stage: σ samples at cost c_e
  - 2nd stage: *1* sample at cost $\sigma.c_e$
- In expectation,
  - 2nd stage cost $\leq$ 1st stage cost
- But we've already bounded 1st stage cost!

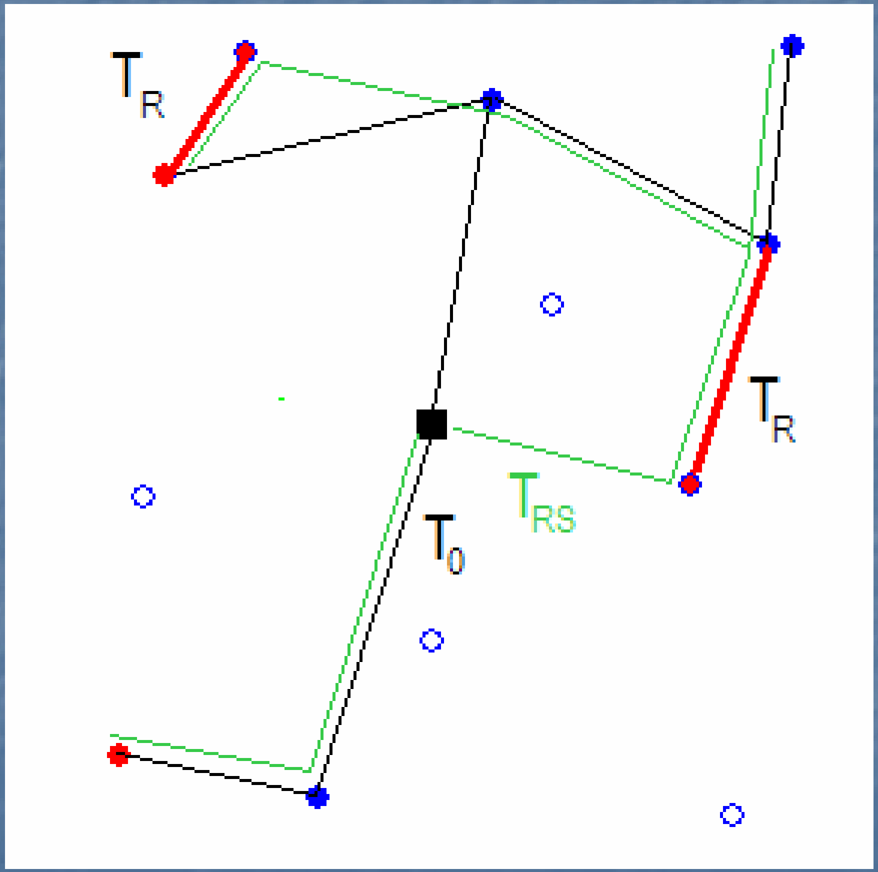
Analysis of 2nd stage cost

- Claim: *$E[\sigma c(T_R)] \leq E[c(T_0)]$*
- Proof using an auxiliary structure



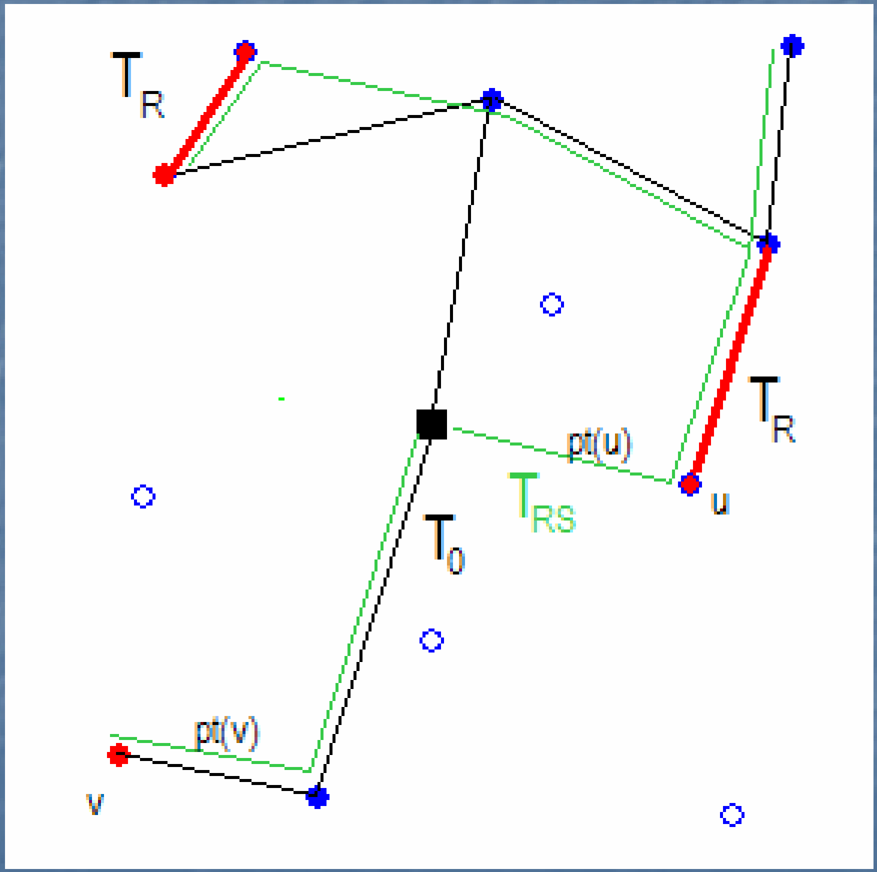
Analysis of 2nd stage cost

- Claim: $E[\sigma c(T_R)] \leq E[c(T_0)]$
- Let T_{RS} be an MST on $R \cup S$



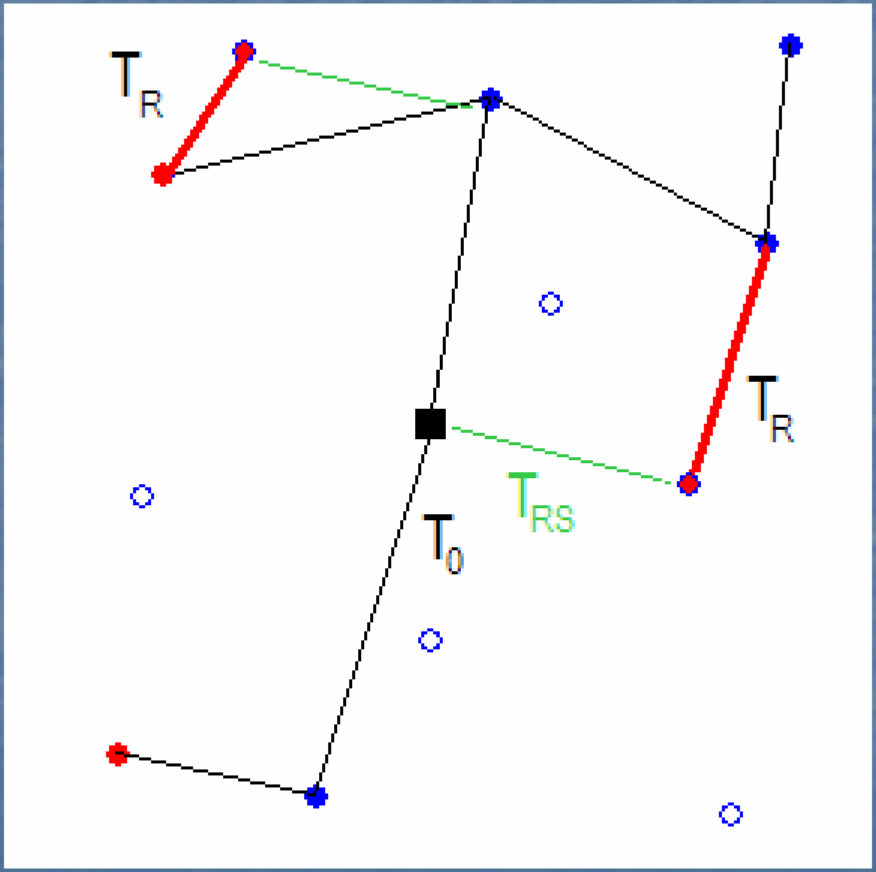
Analysis of 2nd stage cost

- Claim: $E[\sigma c(T_R)] \leq E[c(T_0)]$
- Let T_{RS} be an MST on $R \cup S$
- Associate each node $v \in T_{RS}$ with its parent edge $pt(v)$; $c(T_{RS}) = c(pt(R)) + c(pt(S))$



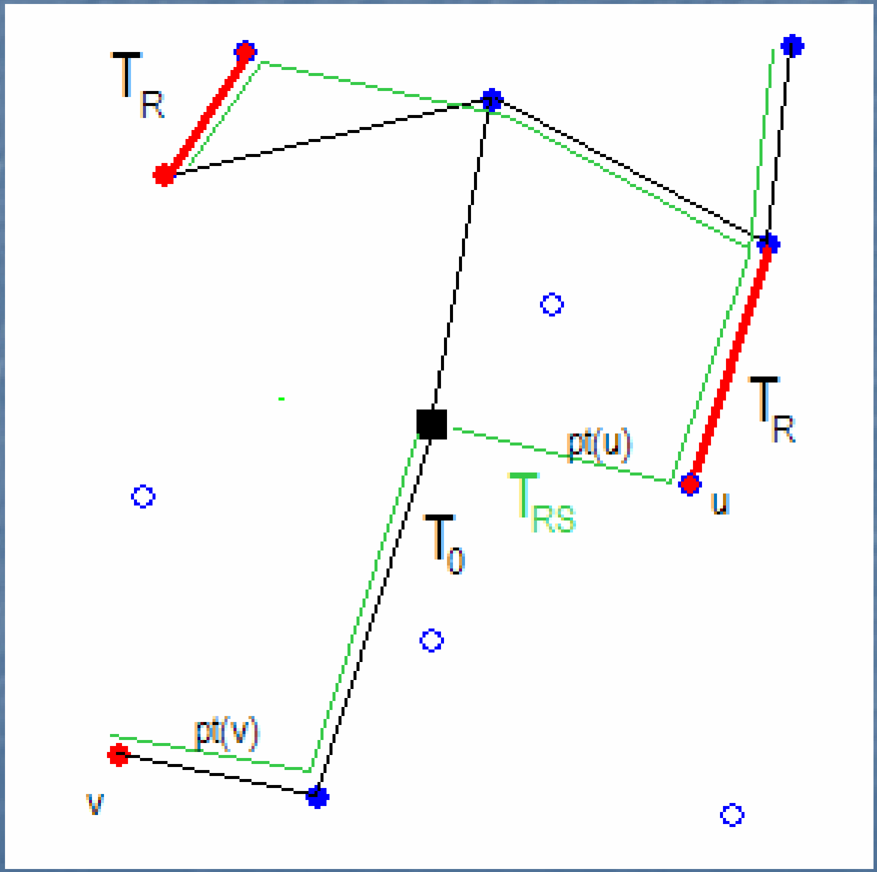
Analysis of 2nd stage cost

- Claim: $E[\sigma c(T_R)] \leq E[c(T_0)]$
- Let T_{RS} be an MST on $R \cup S$
- Associate each node $v \in T_{RS}$ with its parent edge $pt(v)$; $c(T_{RS}) = c(pt(R)) + c(pt(S))$
- $c(T_R) \leq c(pt(R))$, since T_R was the cheapest possible way to connect R to T_0



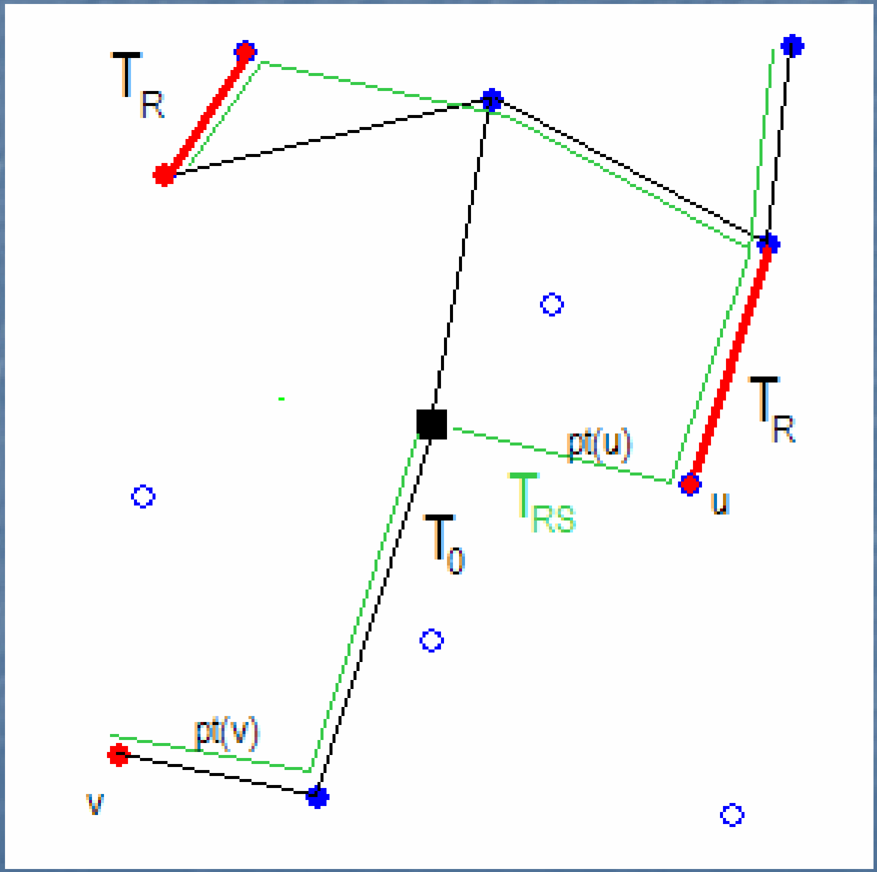
Analysis of 2nd stage cost

- Claim: $E[\sigma c(T_R)] \le E[c(T_0)]$
- Let T_{RS} be an MST on $R \cup S$
- Associate each node $v \in T_{RS}$ with its parent edge $pt(v)$; $c(T_{RS})=c(pt(R)) + c(pt(S))$
- $c(T_R) \le c(pt(R))$
- $E[c(pt(R))] \le E[c(pt(S))]/\sigma$, since R is 1 sample and S is σ samples from same process



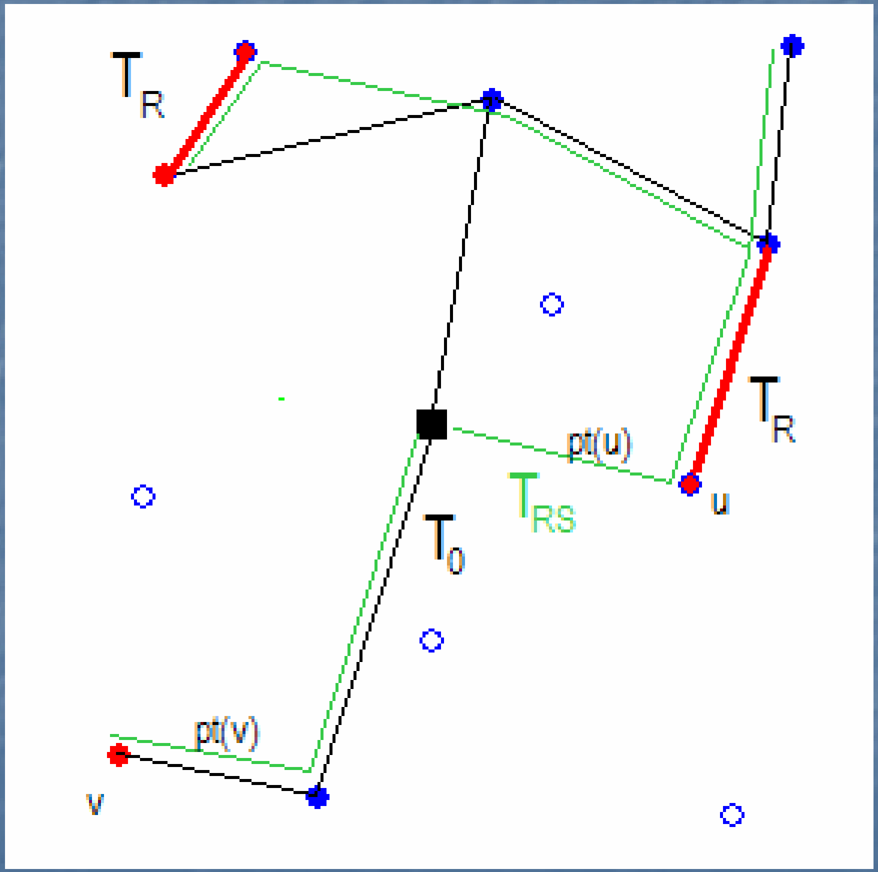
Analysis of 2nd stage cost

- Claim: $E[\sigma c(T_R)] \leq E[c(T_0)]$
- Let T_{RS} be an MST on $R \cup S$
- Associate each node $v \in T_{RS}$ with its parent edge $pt(v)$; $c(T_{RS}) = c(pt(R)) + c(pt(S))$
- $c(T_R) \leq c(pt(R))$
- $E[c(pt(R))] \leq E[c(pt(S))]/\sigma$
- $c(pt(S)) \leq c(T_0)$, since $pt(S) \cup pt(R)$ is a MST while adding $pt(R)$ to T_0 spans $R \cup S$



Analysis of 2nd stage cost

- Claim: $E[\sigma c(T_R)] \leq E[c(T_0)]$
- Let T_{RS} be an MST on $R \cup S$
- Associate each node $v \in T_{RS}$ with its parent edge $pt(v)$; $c(T_{RS})=c(pt(R)) + c(pt(S))$
- $c(T_R) \leq c(pt(R))$
- $E[c(pt(R))] \leq E[c(pt(S))]/\sigma$
- $c(pt(S)) \leq c(T_0)$
- Chain inequalities and claim follows



Recap

- Algorithm for Stochastic Steiner Tree:
  - 1^{st} stage: Sample σ times, build MST
  - 2^{nd} stage: Extend MST to realized clients

Recap

- Algorithm for Stochastic Steiner Tree:
  - 1st stage: Sample σ times, build MST
  - 2nd stage: Extend MST to realized clients
- Theorem: Algorithm BOOST-AND-SAMPLE is a 4-approximation to Stochastic Steiner Tree

Recap

- Algorithm for Stochastic MST:
  - 1st stage: Sample σ times, build MST
  - 2nd stage: Extend MST to realized clients
- Theorem: Algorithm BOOST-AND-SAMPLE is a 4-approximation to Stochastic Steiner Tree
- Shortcomings:
  - Specific problem, in a specific model
  - Cannot adapt to scenario model with non-correlated cost changes across scenarios

Coping with shortcomings

Specific problem, in a specific model

- Boosted Sampling works for more general covering problems with subadditivity - Solves Facility location, vertex cover

Skip general model (details in STOC 04 paper)

Cannot adapt to scenario model with scenario-dependent cost inflations

- A combination of LP-rounding and primal-dual methods solves the scenario model with scenario-dependent cost inflations; Also handles risk-bounds on more general network design.

Skip scenario model (details in FOCS 04 paper)

Skip both

Outline

- Motivation: The cable company problem
- Model and literature review
- Solution to the cable company problem
- General covering problem
- Scenario dependent cost model

General Model

- U : universe of potential clients (e.g., terminals)
- X : elements which provide service, with element costs c_x (e.g., edges)
- Given $S \subseteq U$, set of feasible sol'ns is Sols(S) $\subseteq 2^X$
- Deterministic problem: Given S, find minimum cost $F \in$ Sols(S)

Model: details

- Element costs are c_x in first stage and $\sigma.c_x$ in second stage
- In second stage, client set $S \subseteq U$ is realized with probability $p(S)$
- Objective: Compute F_0 and F_S to minimize

$$c(F_0) + E[\sigma\, c(F_S)]$$

where $F_0 \cup F_S \in \text{Sols}(S)$ for all S

Sampling access model

- Second stage: Client set S appears with probability $p(S)$
- We only require sampling access:
  - Oracle, when queried, gives us a sample scenario D
  - Identically distributed to actual second stage

Main result: Preview

- Given stochastic optimization problem with cost inflation factor σ :
  - Generate σ samples: $D_1, D_2, \ldots, D_\sigma$
  - Use deterministic approximation algorithm to compute $F_0 \in \mathrm{Sols}(\cup D_i)$
  - When actual second stage S is realized, augment by selecting F_S
- Theorem: Good approximation for stochastic problem!

Requirement: Sub-additivity

- If S and S' are legal sets of clients, then:
  - $S \cup S'$ is also a legal client set
  - For any $F \in \mathrm{Sols}(S)$ and $F' \in \mathrm{Sols}(S')$, we also have $F \cup F' \in \mathrm{Sols}(S \cup S')$

Requirement: Approximation

- There is an α-approximation algorithm for deterministic problem
  - Given any $S \subseteq U$, can find $F \in \mathrm{Sols}(S)$ in polynomial time such that:

    $c(F) \leq \alpha.\min \{c(F'): F' \in \mathrm{Sols}(S)\}$

Crucial ingredient: Cost shares

- Recall Stochastic Steiner Tree:
  - Bounding 2nd stage cost required allocating the cost of an MST to the client nodes, and summing up carefully (auxiliary structure)
- Cost sharing function: way of distributing solution cost to clients
- Originated in game theory [Young, ’94], adapted to approximation algorithms [Gupta, Kumar, Pal, Roughgarden FOCS ’03]

Requirement: Cost-sharing

- $\xi : 2^U \times U \rightarrow R$ is a β -strict cost sharing function for α-approximation A if:
  - $\xi(S,j) > 0$ only if $j \in S$
  - $\sum_{j \in S} \xi(S,j) \leq c(\mathrm{OPT}(S))$
  - If $S' = S \cup T$, $\mathrm{A}(S)$ is an α-approx. for S, and $\mathrm{Aug}(S,T)$ provides a solution for augmenting $\mathrm{A}(S)$ to also serve T, then

    $\sum_{j \in T} \xi(S',j) \geq (1/\beta)\, c(\mathrm{Aug}(S,T))$

Main theorem: Formal

- Given a sub-additive problem with α-approximation algorithm A and β-strict cost sharing function, the following is an $(\alpha+\beta)$-approximation algorithm for stochastic variant:
  - Generate σ samples: $D_1, D_2, \ldots, D_\sigma$
  - First stage: Use algorithm A to compute F_0 as an α-approximation for $\cup D_i$
  - Second stage: When actual set S is realized, use algorithm Aug($\cup D_i$, S) to compute F_S

First-stage cost

- Samples D_i, Algo A generates $F_0 \in$ Sols($\cup\ D_i$)
- Define optimum: $Z^* = c(F_0^*) + \sum_S p(S).\sigma.c(F_S^*)$
- By sub-additivity,

  $F_0^* \cup F_{D1}^* \cup \ldots \cup F_{D\sigma}^* \in$ Sols($\cup\ D_i$)
- Since A is α-approximation,

  $c(F_0)/\alpha \quad \leq \quad c(F_0^*) + \sum_i c(F_{Di}^*)$
- E[$c(F_0)$]/$\alpha \quad \leq \quad c(F_0^*) + \sum_i$ E[$c(F_S^*)$]

  $\leq \quad c(F_0^*) + \sigma \sum_S p(S)\, c(F_S^*) \;=\; Z^*$
- Therefore, first-stage cost E[$c(F_0)$] $\leq\ \alpha.Z^*$

Second-stage cost

- D_i: samples, S: actual 2nd stage, define $S' = S \cup D_i$
- $c(F_S) \leq \beta.\xi(S',S)$, by cost-sharing function defn.
- $\xi(S',D_1) + \ldots + \xi(S',D_\sigma) + \xi(S',S) \leq c\,(\mathrm{OPT}(S'))$
- S' has $\sigma+1$ client sets, identically distributed:

  $\mathrm{E}[\xi(S',S)] \leq \mathrm{E}[c\,(\mathrm{OPT}(S'))] \,/\, (\sigma+1)$
- $c\,(\mathrm{OPT}(S')) \leq c(F_0{}^*) + c(F_{D1}{}^*) + \ldots + c(F_{D\sigma}{}^*) + c(F_S{}^*)$,

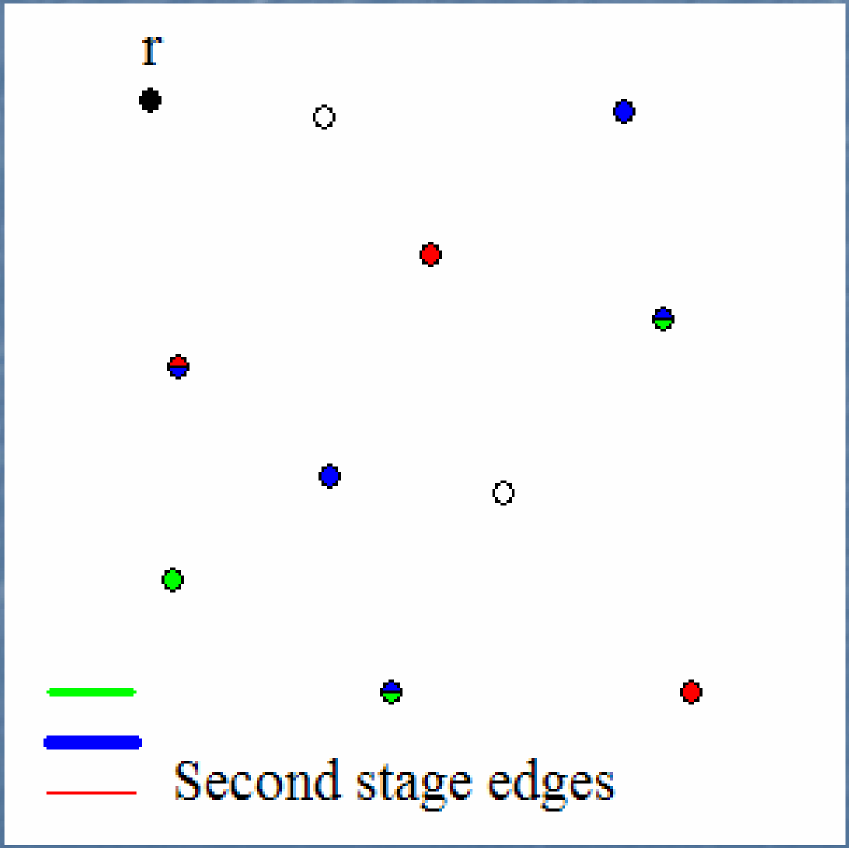
  by sub-additivity
- $\mathrm{E}[c\,(\mathrm{OPT}(S'))] \leq c(F_0{}^*) + (\sigma+1)\,\mathrm{E}[c(F_s)] \leq (\sigma+1)Z^*/\sigma$
- $\mathrm{E}[\sigma.c(F_S)] \leq \beta.Z^*$, bounding second-stage cost

Outline

- Motivation: The cable company problem
- Model and literature review
- Solution to the cable company problem
- General covering problem
- Scenario dependent cost model

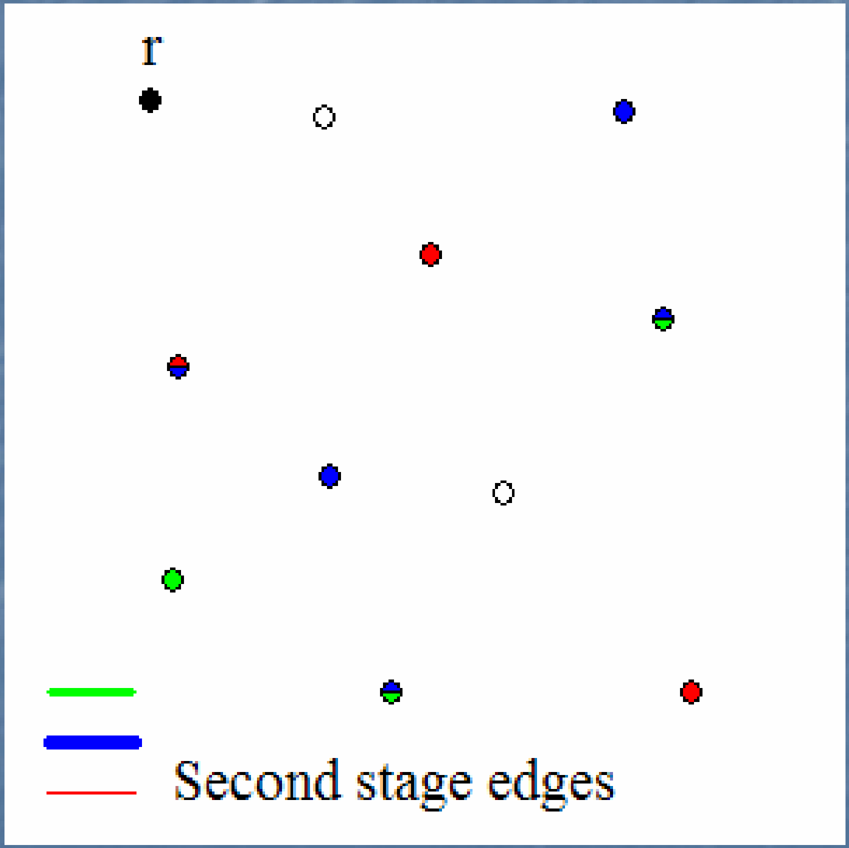
Stochastic Steiner Tree

- First stage: *G*, *r* given
- 2nd stage: one of *m* scenarios occurs:
  - Terminals S_k
  - Probability p_k
  - Edge cost inflation factor σ_k



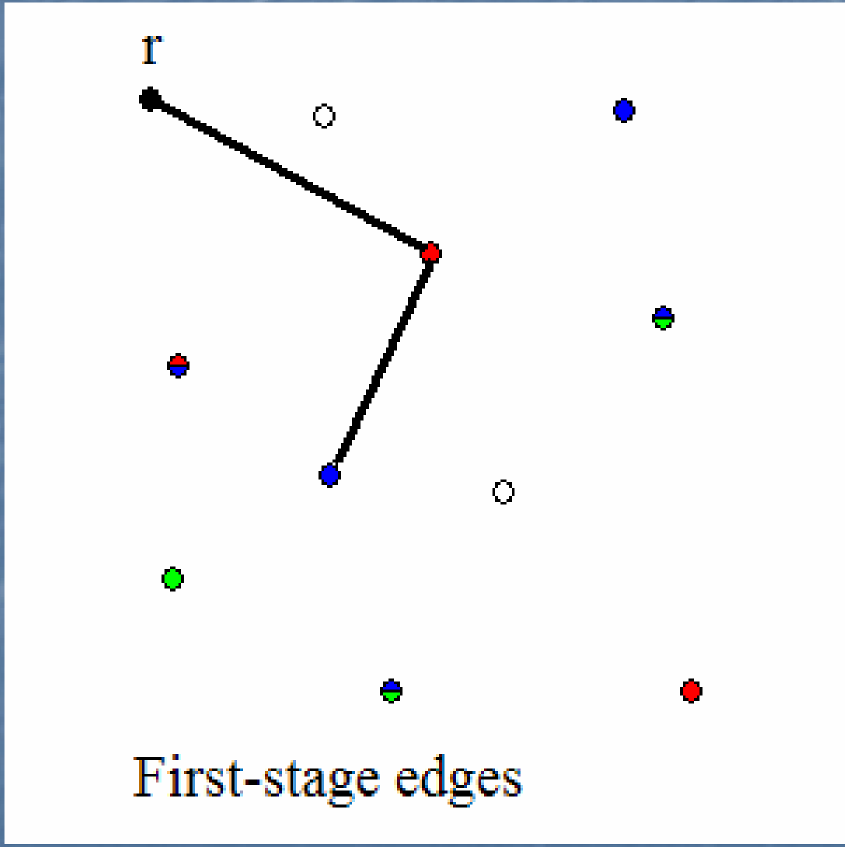
Stochastic Steiner Tree

- First stage: G, r given
- 2^{nd} stage: one of m scenarios occurs:
  - Terminals S_k
  - Probability p_k
  - Edge cost inflation factor σ_k
- Objective: 1^{st} stage tree T^0, 2^{nd} stage trees T^k s.t. $T^0 \cup T^k$ span S_k



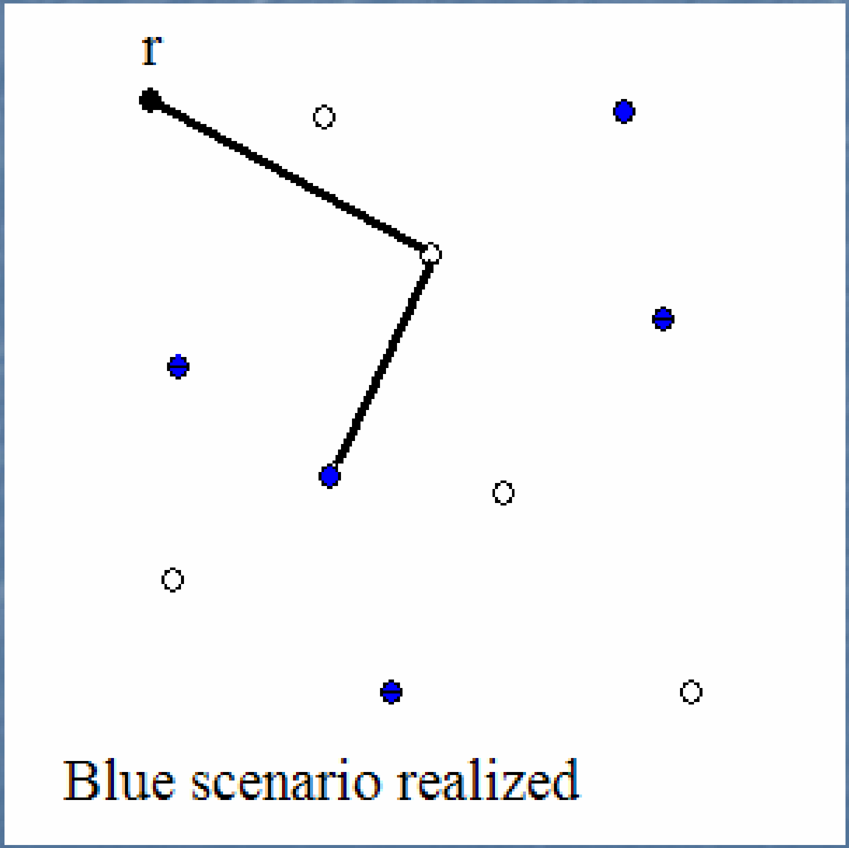
Stochastic Steiner Tree

- First stage: G, r given
- 2nd stage: one of m scenarios occurs:
  - Terminals S_k
  - Probability p_k
  - Edge cost inflation factor σ_k
- Objective: 1st stage tree T^0, 2nd stage trees T^k s.t. $T^0 \cup T^k$ span S_k



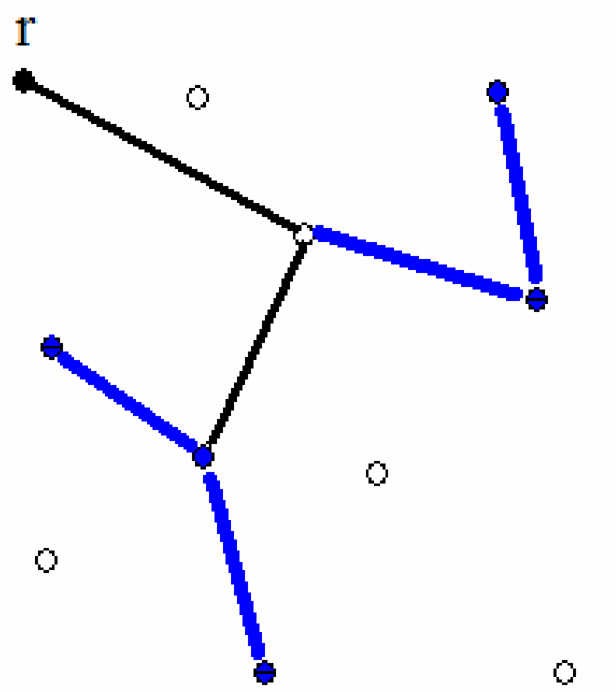
Stochastic Steiner Tree

- First stage: G, r given
- 2nd stage: one of m scenarios occurs:
  - Terminals S_k
  - Probability p_k
  - Edge cost inflation factor σ_k
- Objective: 1st stage tree T^0, 2nd stage trees T^k s.t. $T^0 \cup T^k$ span S_k



Stochastic Steiner Tree

- First stage: G, r given
- 2^{nd} stage: one of m scenarios occurs:
  - Terminals S_k
  - Probability p_k
  - Edge cost inflation factor σ_k
- Objective: 1^{st} stage tree T^0, 2^{nd} stage trees T^k s.t. $T^0 \cup T^k$ span S_k

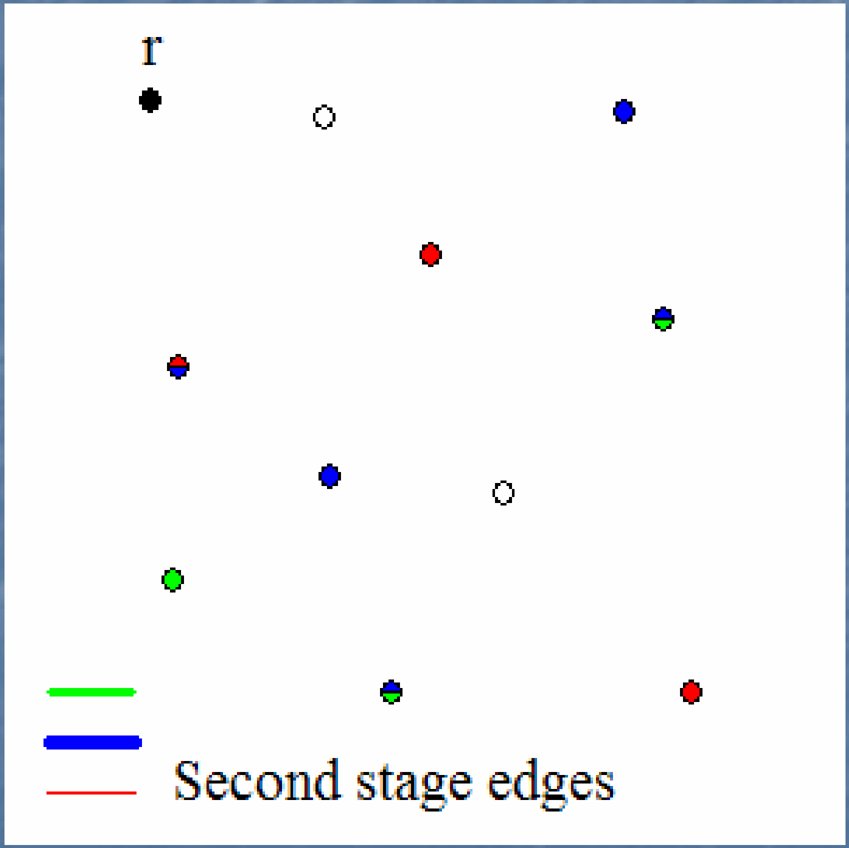


Second stage for realized scenario

Stochastic Steiner Tree

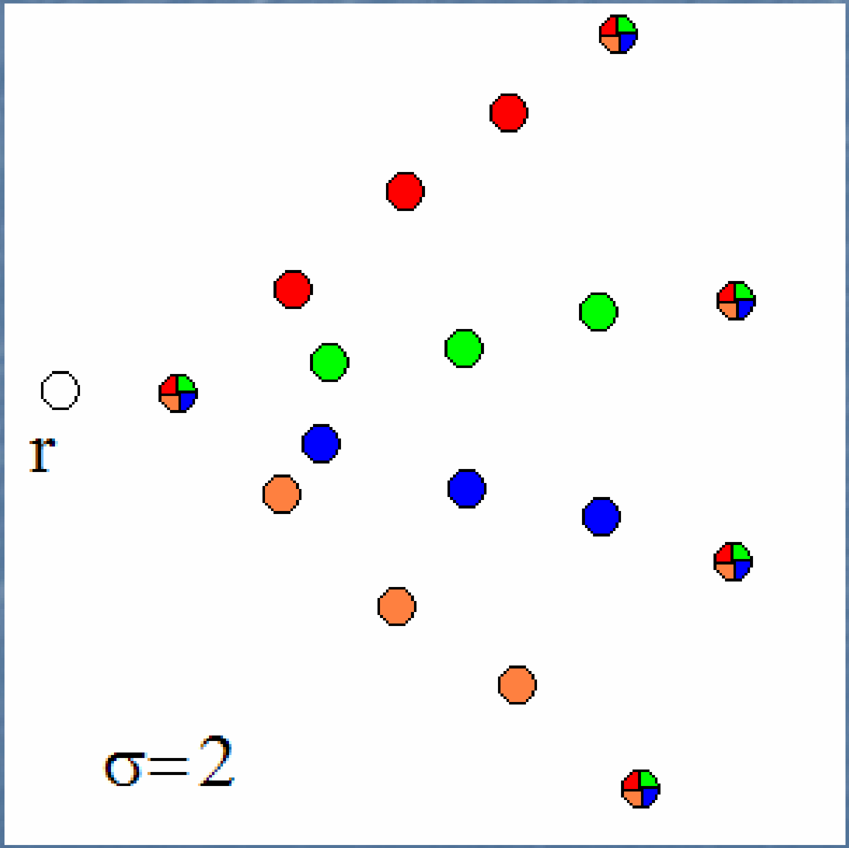
- First stage: G, r given
- 2nd stage: one of m scenarios occurs:
  - Terminals S_k
  - Probability p_k
  - Edge cost inflation factor σ_k
- Objective: 1st stage tree T^0, 2nd stage trees T^k s.t. $T^0 \cup T^k$ span S_k
- Minimize $c(T^0)+E[c(T')]$

[Skip Algorithm]



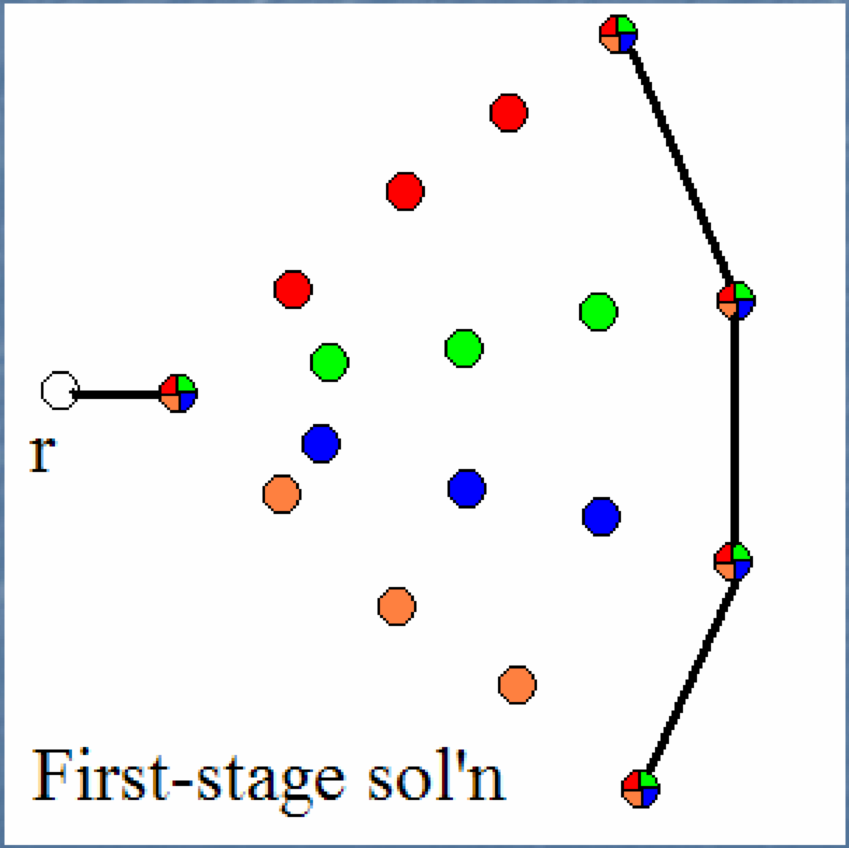
Tree solutions

- Example with 4 scenarios and *$\sigma=2$*



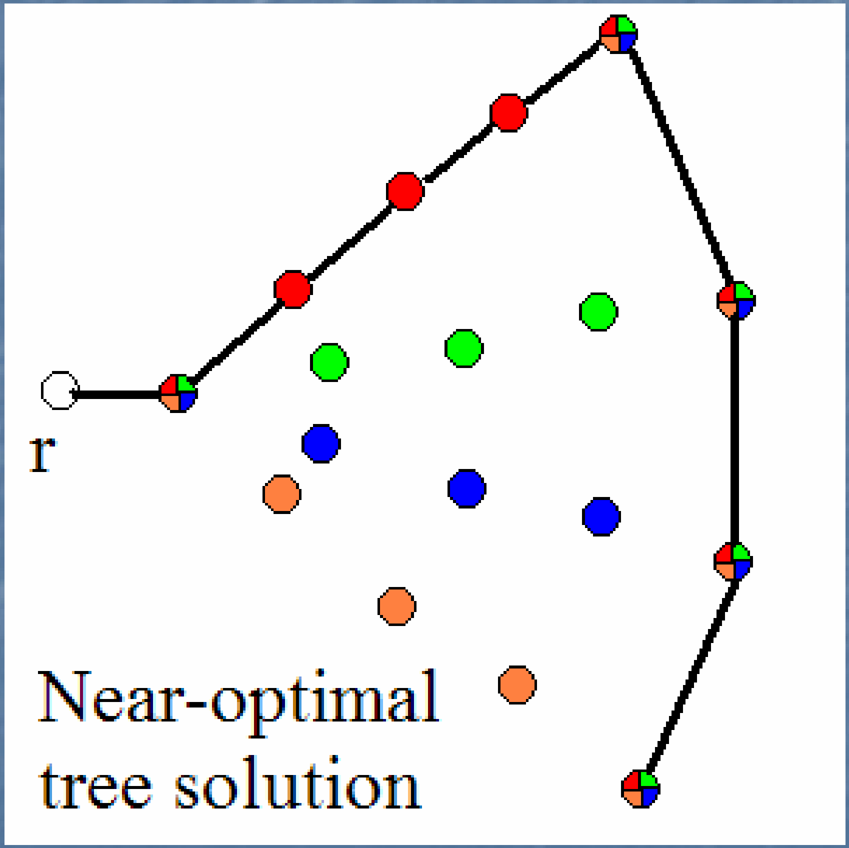
Tree solutions

- Example with 4 scenarios and $\sigma=2$
- Optimal solution may have lots of components!



Tree solutions

- Example with 4 scenarios and $\sigma=2$
- Optimal solution may have lots of components!
- Lemma: There exists a solution where 1st stage is a tree and overall cost is no more than 3 times the optimal cost
- Restrict to tree solutions



IP formulation

- Tree solution: From any (2^{nd}-stage) terminal, path to root consists of exactly two parts: strictly 2^{nd}-stage, followed by strictly 1^{st}-stage
- IP: Install edges to support unit flow along such paths from each terminal to root

IP formulation

$$\min \sum_{e\in E} c(e)x_e^0 + \sum_{k=1}^{m} p_k \sigma_k \sum_{e\in E} c(e) x_e^k$$

$$\begin{aligned}
\sum_{e\in\delta_+(t)} (r_e^0(t) + r_e^k(t)) &\geq 1 \\
\sum_{e\in\delta_+(v)} (r_e^0(t) + r_e^k(t)) - \sum_{e\in\delta_-(v)} (r_e^0(t) + r_e^k(t)) &= 0 \\
\sum_{e\in\delta_-(v)} r_e^0(t) - \sum_{e\in\delta_+(v)} r_e^0(t) &\leq 0 \\
r_e^k(t) - x_e^k &\leq 0
\end{aligned}$$

---

x_e^k: edge *e* installed in scenario *k*;
$r_e^k(t)$: flow on edge *e* of type *k* from terminal *t* ;
for $k = 0$ (1st stage) and $i=1,2,\ldots,m$ (2nd stage)

IP formulation

$$\min \sum_{e\in E} c(e)x_e^0 + \sum_{k=1}^{m} p_k\sigma_k \sum_{e\in E} c(e)x_e^k$$

$$\begin{array}{rcl}
\displaystyle\sum_{e\in\delta_+(t)} (r_e^0(t)+r_e^k(t)) & \geq & 1 \\
\displaystyle\sum_{e\in\delta_+(v)} (r_e^0(t)+r_e^k(t)) - \sum_{e\in\delta_-(v)} (r_e^0(t)+r_e^k(t)) & = & 0 \\
\displaystyle\sum_{e\in\delta_-(v)} r_e^0(t) - \sum_{e\in\delta_+(v)} r_e^0(t) & \leq & 0 \\
r_e^k(t) - x_e^k & \leq & 0
\end{array}$$

Objective: minimize expected cost

IP formulation

$$\min \sum_{e\in E} c(e)x_e^0 + \sum_{k=1}^{m} p_k\sigma_k \sum_{e\in E} c(e)x_e^k$$

$$\sum_{e\in\delta_+(t)} (r_e^0(t) + r_e^k(t)) \quad \geq \quad 1$$

$$\sum_{e\in\delta_+(v)} (r_e^0(t) + r_e^k(t)) - \sum_{e\in\delta_-(v)} (r_e^0(t) + r_e^k(t)) \quad = \quad 0$$

$$\sum_{e\in\delta_-(v)} r_e^0(t) - \sum_{e\in\delta_+(v)} r_e^0(t) \quad \leq \quad 0$$

$$r_e^k(t) - x_e^k \quad \leq \quad 0$$

Unit out-flow from each terminal

IP formulation

$$\min \sum_{e\in E} c(e)x_e^0 + \sum_{k=1}^{m} p_k \sigma_k \sum_{e\in E} c(e)x_e^k$$

$$\sum_{e\in\delta_+(t)} (r_e^0(t) + r_e^k(t)) \quad \geq \quad 1$$

$$\boxed{\sum_{e\in\delta_+(v)} (r_e^0(t) + r_e^k(t)) - \sum_{e\in\delta_-(v)} (r_e^0(t) + r_e^k(t)) \quad = \quad 0}$$

$$\sum_{e\in\delta_-(v)} r_e^0(t) - \sum_{e\in\delta_+(v)} r_e^0(t) \quad \leq \quad 0$$

$$r_e^k(t) - x_e^k \quad \leq \quad 0$$

Flow conservation at all internal nodes ($v \neq t, r$)

IP formulation

$$\min \sum_{e \in E} c(e) x_e^0 + \sum_{k=1}^{m} p_k \sigma_k \sum_{e \in E} c(e) x_e^k$$

$$\begin{array}{rcl}
\sum_{e \in \delta_+(t)} (r_e^0(t) + r_e^k(t)) & \geq & 1 \\
\sum_{e \in \delta_+(v)} (r_e^0(t) + r_e^k(t)) - \sum_{e \in \delta_-(v)} (r_e^0(t) + r_e^k(t)) & = & 0 \\
\boxed{\sum_{e \in \delta_-(v)} r_e^0(t) - \sum_{e \in \delta_+(v)} r_e^0(t) \quad \leq \quad 0} & & \\
r_e^k(t) - x_e^k & \leq & 0
\end{array}$$

Flow monotonicity: enforces "First-stage must be a tree"

IP formulation

$$\min \sum_{e\in E} c(e)x_e^0 + \sum_{k=1}^{m} p_k\sigma_k \sum_{e\in E} c(e)x_e^k$$

$$\begin{array}{rcl}
\displaystyle\sum_{e\in\delta_+(t)} (r_e^0(t) + r_e^k(t)) & \geq & 1 \\
\displaystyle\sum_{e\in\delta_+(v)} (r_e^0(t) + r_e^k(t)) - \sum_{e\in\delta_-(v)} (r_e^0(t) + r_e^k(t)) & = & 0 \\
\displaystyle\sum_{e\in\delta_-(v)} r_e^0(t) - \sum_{e\in\delta_+(v)} r_e^0(t) & \leq & 0 \\
r_e^k(t) - x_e^k & \leq & 0
\end{array}$$

Flow support: If an edge has flow, it must be accounted for in the objective function

IP formulation

$$
\begin{array}{rcr}
\min \sum_{e\in E} c(e)x_e^0 + \sum_{k=1}^{m} p_k\sigma_k \sum_{e\in E} c(e)x_e^k & & \\
\sum_{e\in\delta_+(t)} (r_e^0(t) + r_e^k(t)) & \geq & 1 \\
\sum_{e\in\delta_+(v)} (r_e^0(t) + r_e^k(t)) - \sum_{e\in\delta_-(v)} (r_e^0(t) + r_e^k(t)) & = & 0 \\
\sum_{e\in\delta_-(v)} r_e^0(t) - \sum_{e\in\delta_+(v)} r_e^0(t) & \leq & 0 \\
r_e^k(t) - x_e^k & \leq & 0 \\
r_e^k(t), x_e^k & \geq & 0 \\
r_e^k(t), x_e^k & \in & \{0,1\}
\end{array}
$$

Algorithm overview

- $(x,r) \leftarrow$ Optimal solution to LP relaxation

Algorithm overview

- $(x,r) \leftarrow$ Optimal solution to LP relaxation
- 1st stage solution:
  - Obtain a new graph G' where $2x^0$ forms a fractional Steiner tree
  - Round using primal-dual algorithm; this is T^0

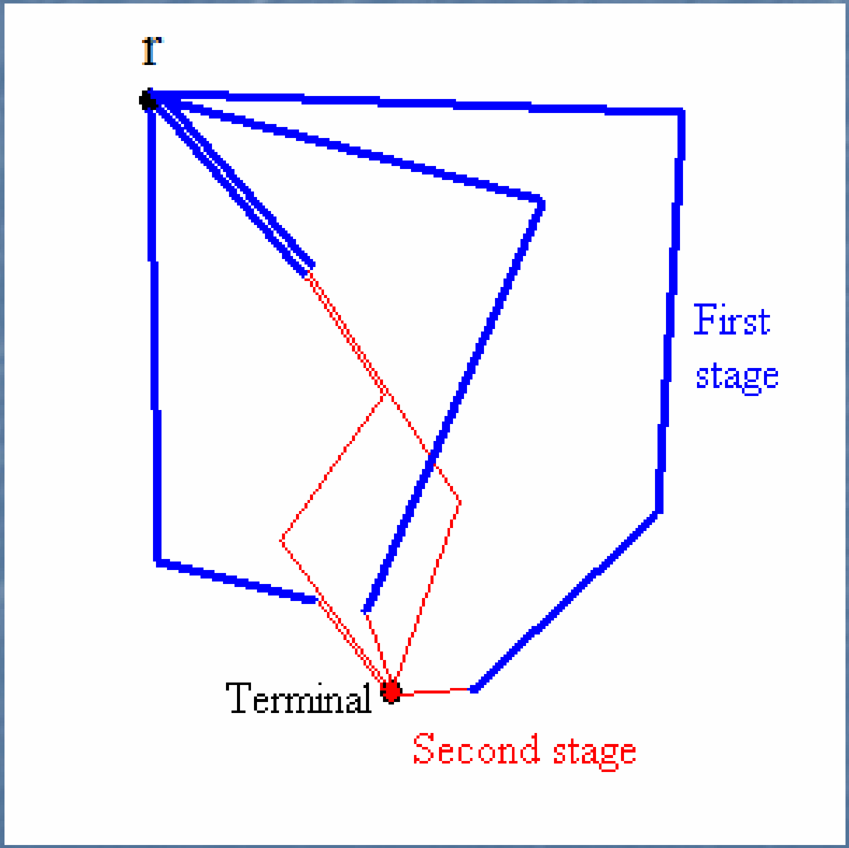
Algorithm overview

- $(x,r) \leftarrow$ Optimal solution to LP relaxation
- 1st stage solution:
  - Obtain a new graph G' where $2x^0$ forms a fractional Steiner tree
  - Round using primal-dual algorithm; this is T^0
- 2nd stage solution:
  - Examine remaining terminals in each scenario
  - Use modified primal-dual method to obtain T^k

Skip Analysis

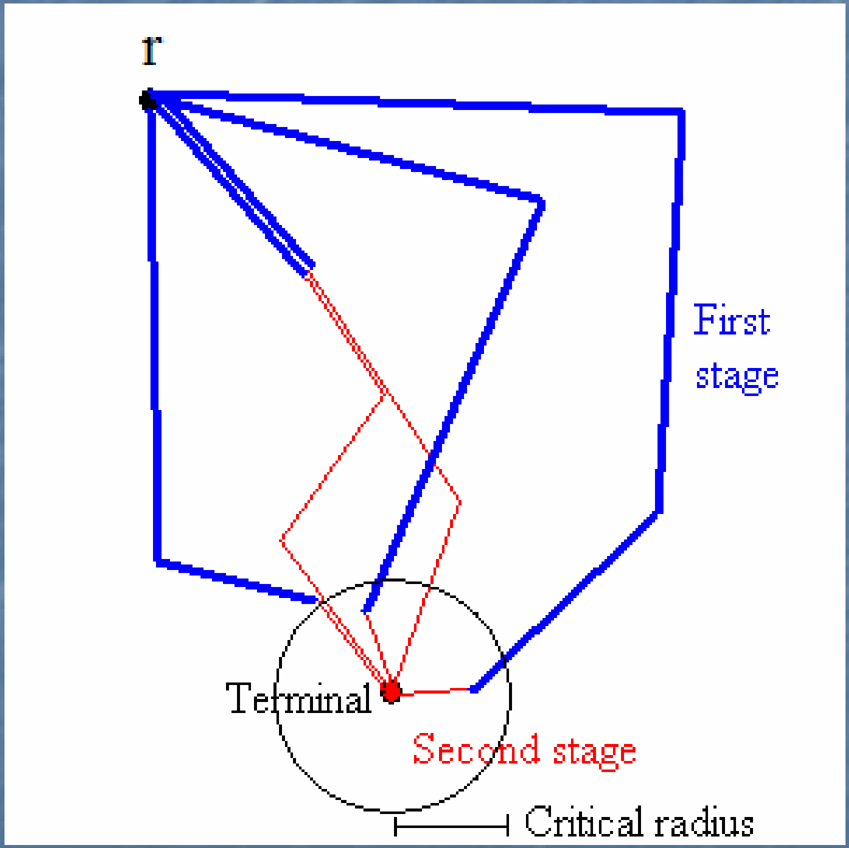
First stage

- Examine fractional paths for each terminal



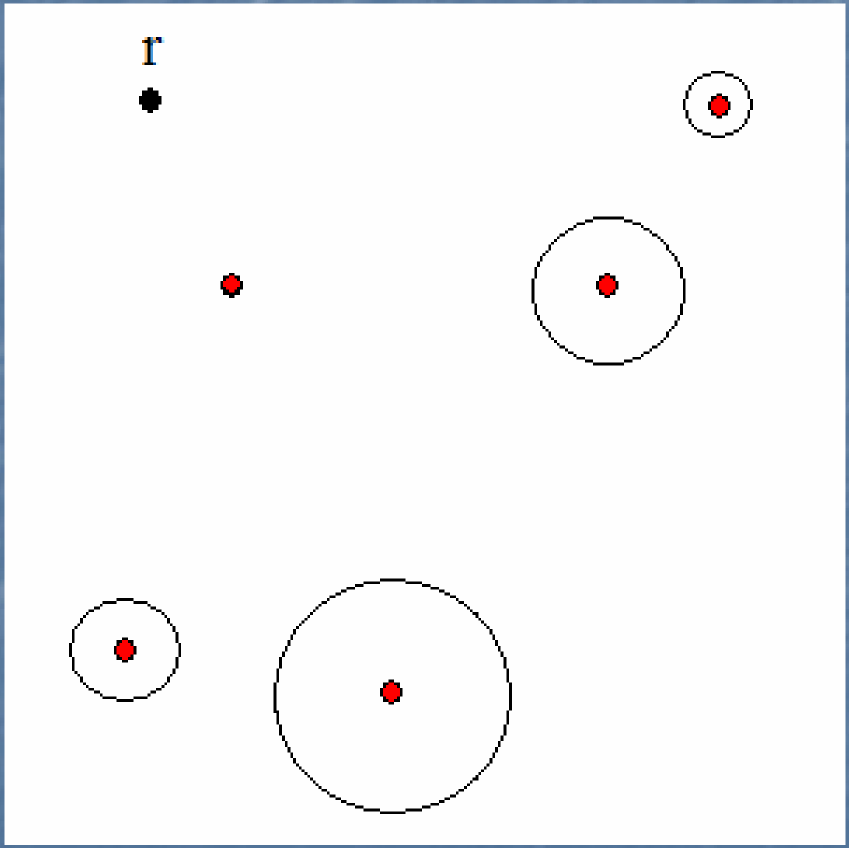
First stage

- Examine fractional paths for each terminal
- Critical radius: Flow "transitions" from 2nd-stage to 1st-stage



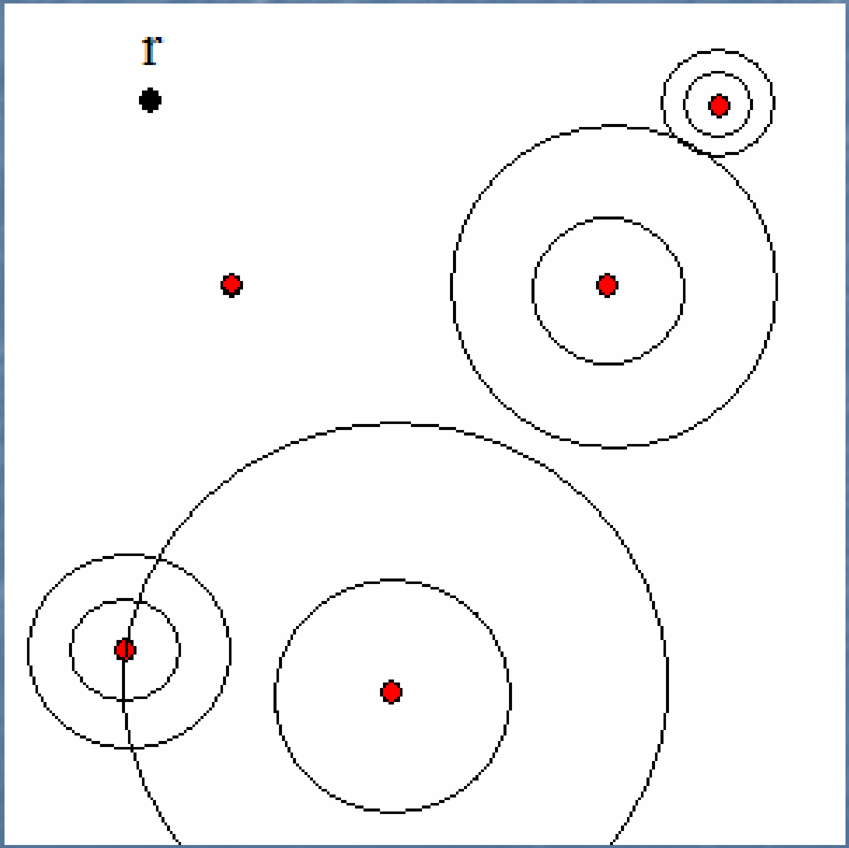
First stage

- Examine fractional paths for each terminal
- Critical radius: Flow "transitions" from 2^{nd}-stage to 1^{st}-stage
- Construct critical radii for all terminals



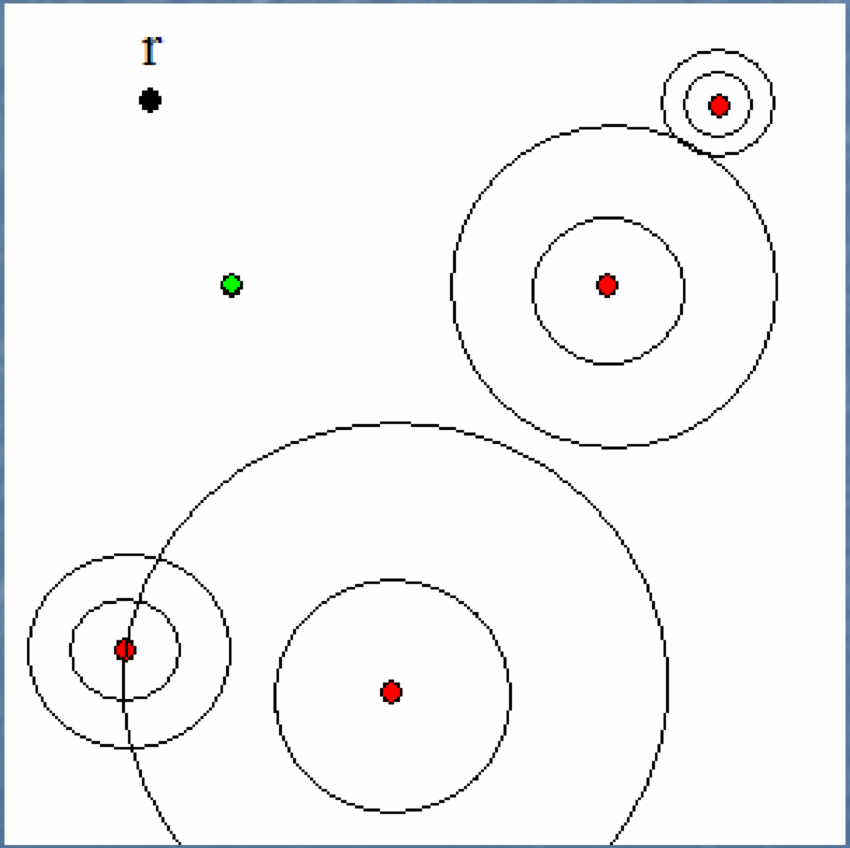
First stage

- Critical radius: Fractional flow "transitions" from 2nd-stage to 1st-stage
- Construct twice the critical radii for all terminals



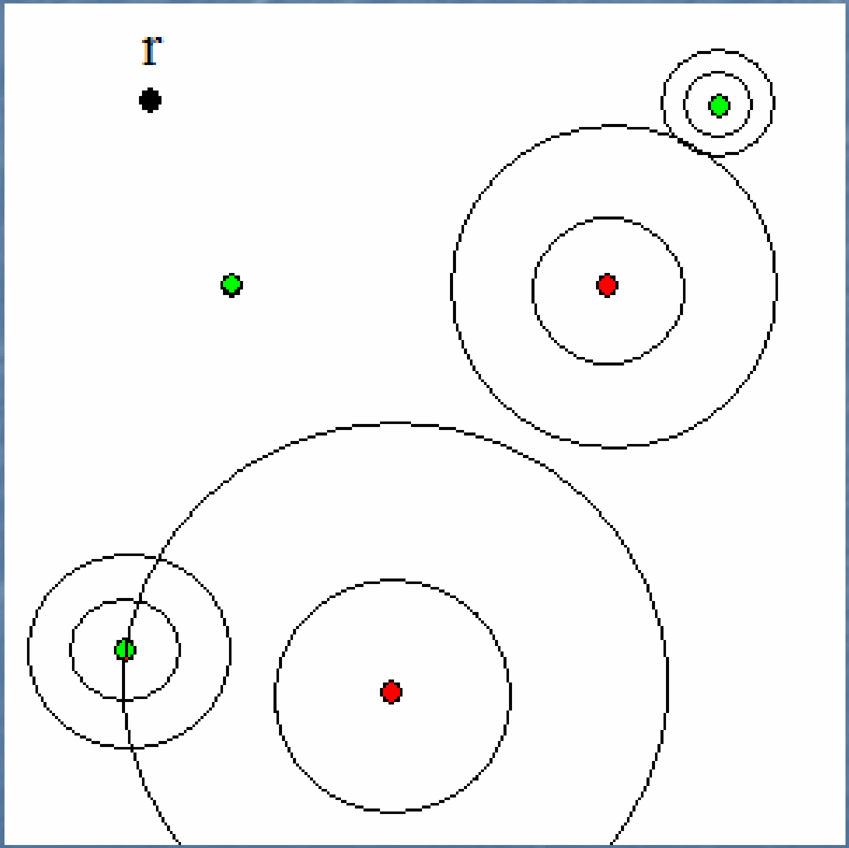
First stage

- Critical radius: Fractional flow "transitions" from 2nd-stage to 1st-stage
- Construct twice the *c.r.* for all terminals
- Examine in increasing order of *c.r.*
- $R^0 \leftarrow$ independent set based on $2 \times$ *c.r.*



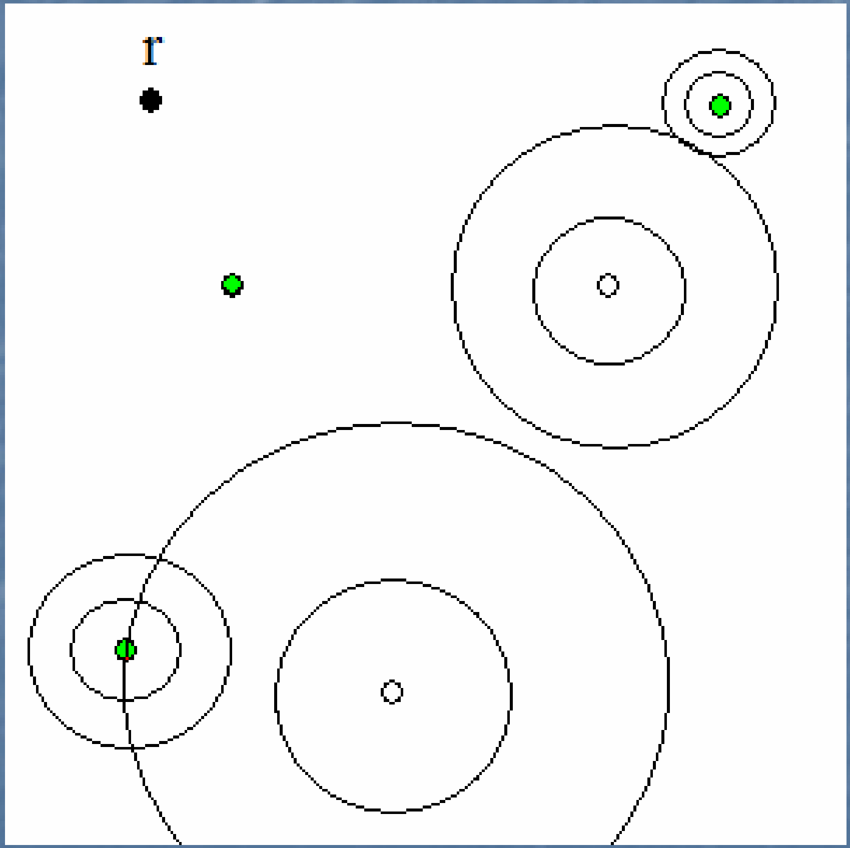
First stage

- Critical radius: Fractional flow "transitions" from 2nd-stage to 1st-stage
- Construct twice the *c.r.* for all terminals
- Examine in increasing order of *c.r.*
- $R^0 \leftarrow$ independent set based on $2 \times$ *c.r.*



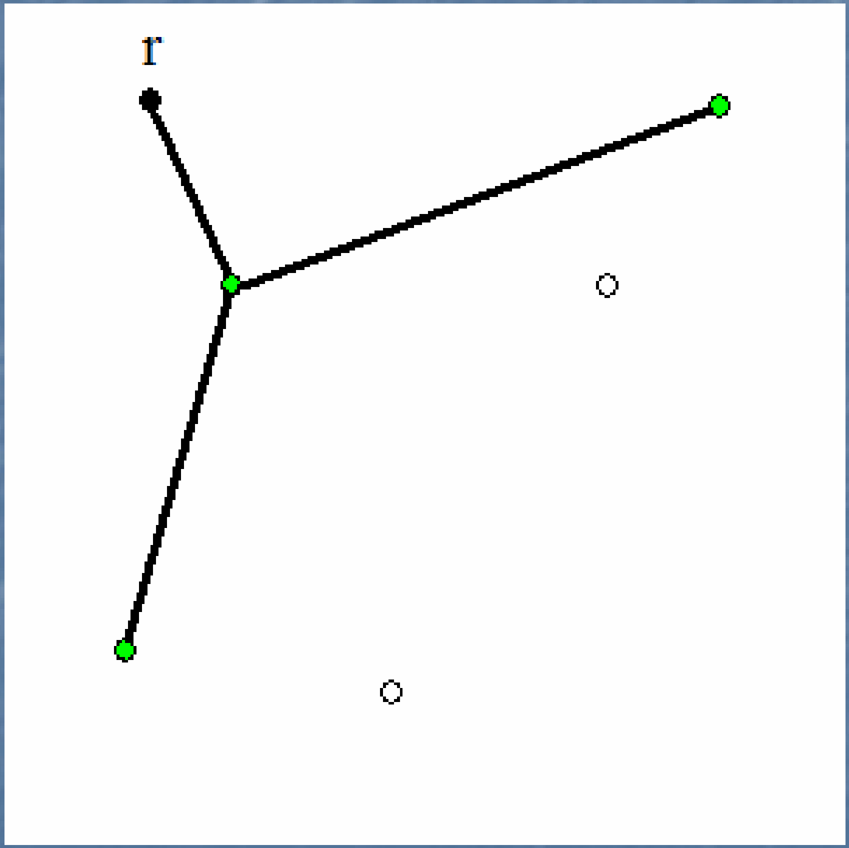
First stage

- Critical radius: Fractional flow "transitions" from 2nd-stage to 1st-stage
- Construct twice the *c.r.* for all terminals
- Examine in increasing order of *c.r.*
- $R^0 \leftarrow$ independent set based on $2 \times c.r.$



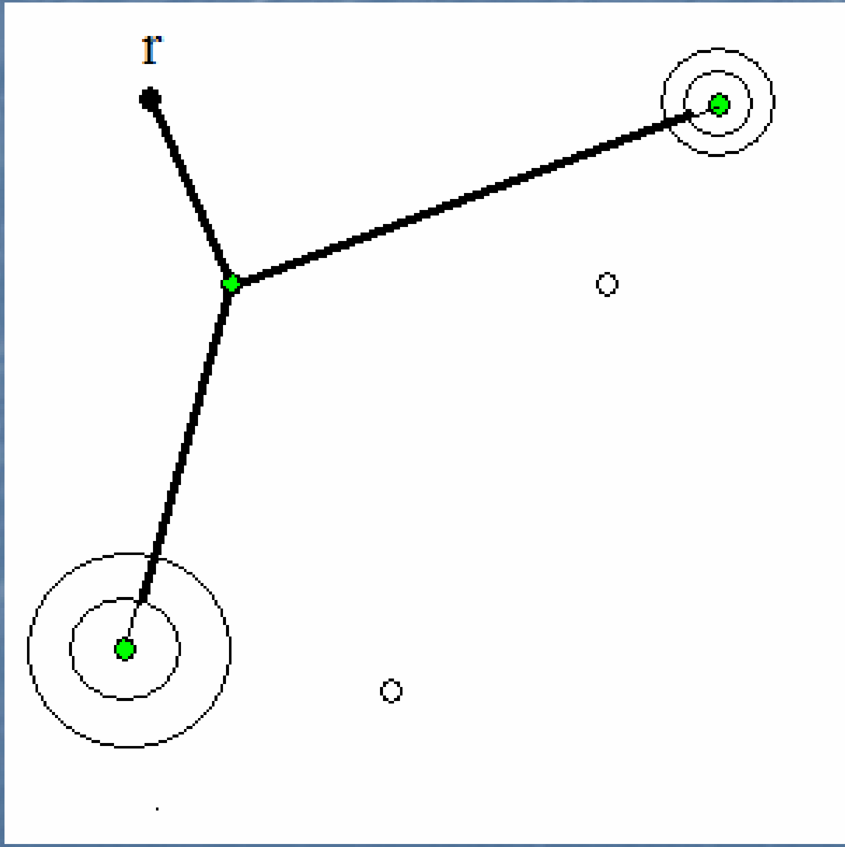
First stage

- Critical radius: Fractional flow "transitions" from 2nd-stage to 1st-stage
- Construct twice the *c.r.* for all terminals
- Examine in increasing order of *c.r.*
- $R^0 \leftarrow$ independent set based on $2 \times c.r.$
- $T^0 \leftarrow$ Steiner tree on R^0



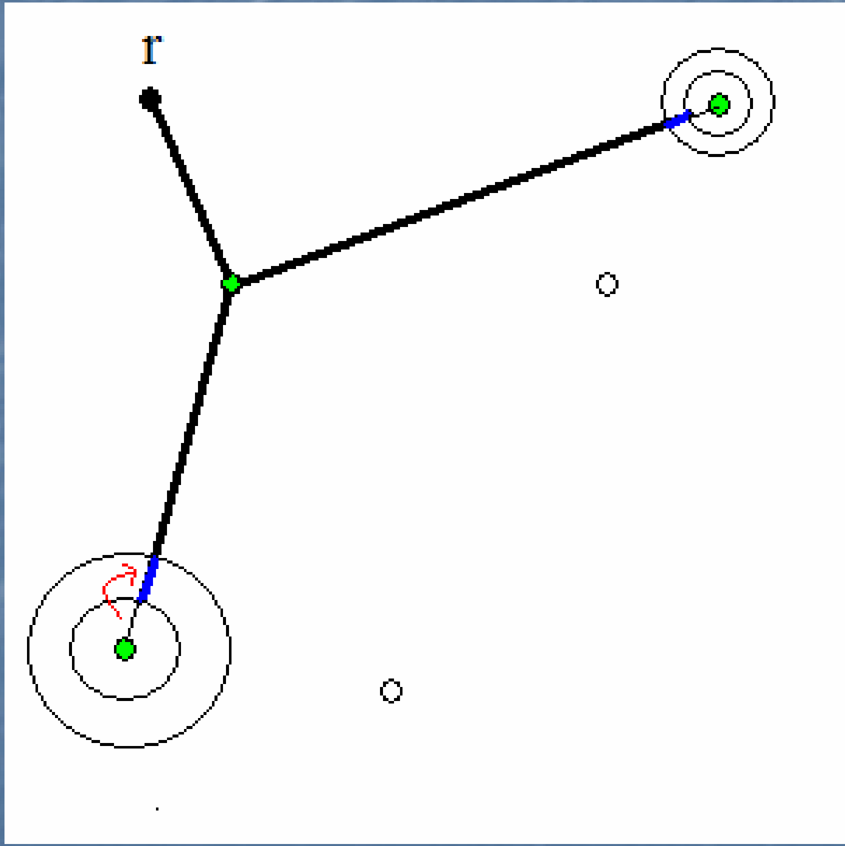
First stage analysis

- Critical radius: Fractional flow "transitions" from 2^{nd}-stage to 1^{st}-stage
- $R^0 \leftarrow$ independent set based on $2 \times c.r.$
- $T^0 \leftarrow$ Steiner tree on R^0
- $G' \leftarrow$ Contract *c.r.* balls around vertices in R^0
- $2x^0$ is feasible fractional Steiner tree for R^0 in G'



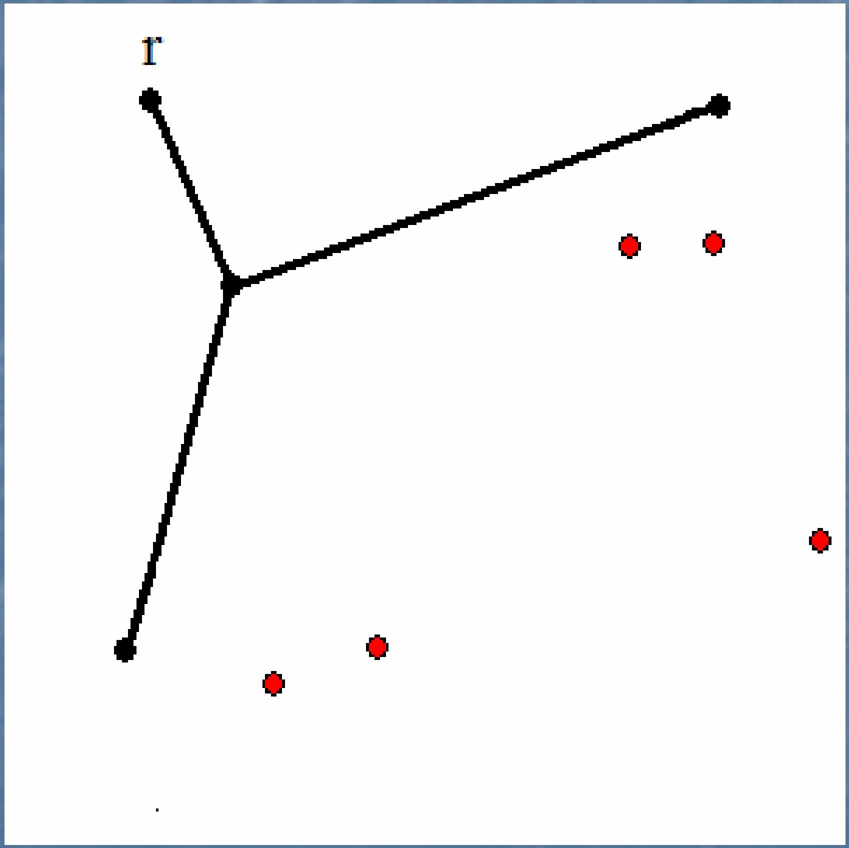
First stage analysis

- $R^0 \leftarrow$ independent set based on $2 \times c.r.$
- $T^0 \leftarrow$ Steiner tree on R^0
- $G' \leftarrow$ Contract *c.r.* balls around vertices in R^0
- $2x^0$ is feasible fractional Steiner tree for R^0 in G'
- Extension from vertex to *c.r.* charged to segment from *c.r.* to $2 \times c.r.$ (disjoint from others)



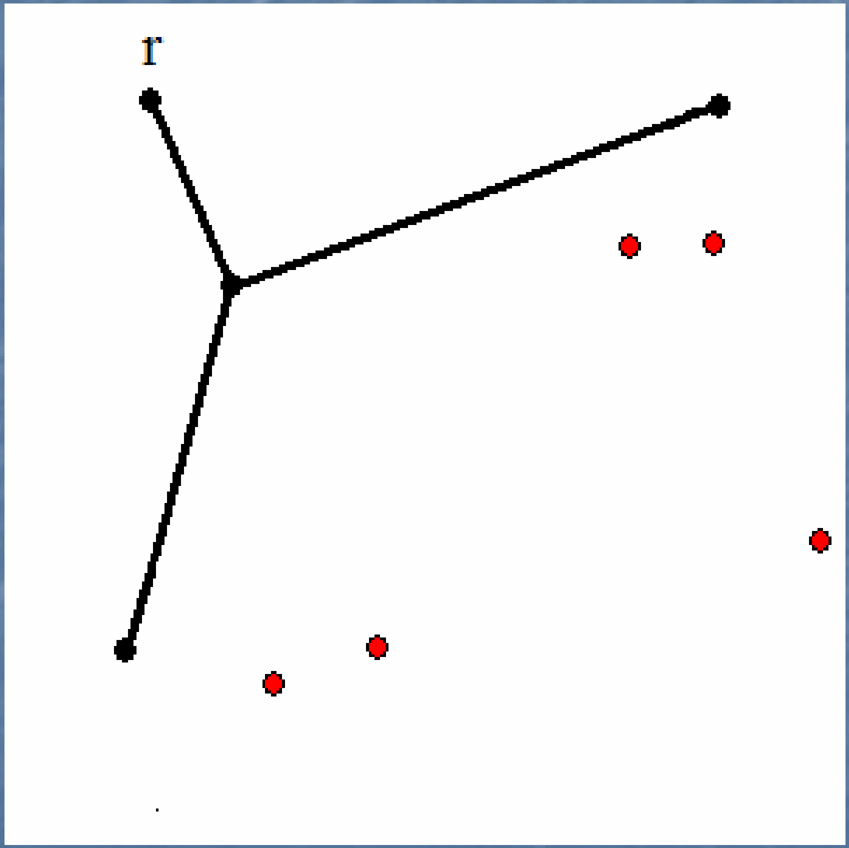
Second stage

- $T^0 \leftarrow 1^{st}$ stage tree
- Consider scenario *k*



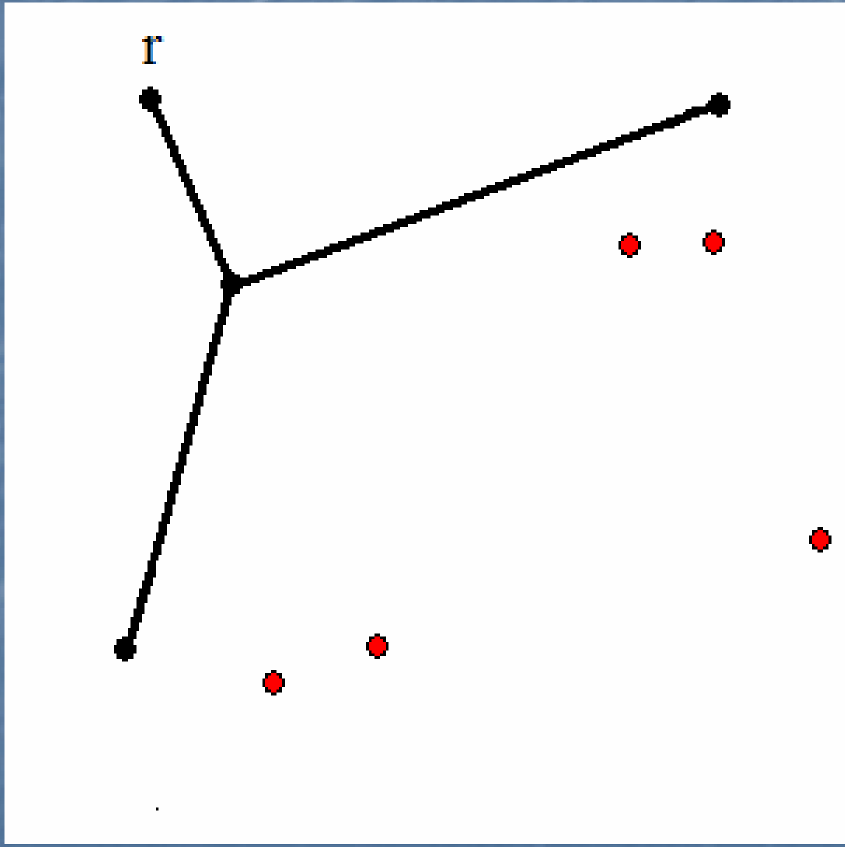
Second stage

- $T^0 \leftarrow 1^{st}$ stage tree
- Consider scenario k
- Idea: Run Steiner tree primal-dual on terminals, stopping moat M when:



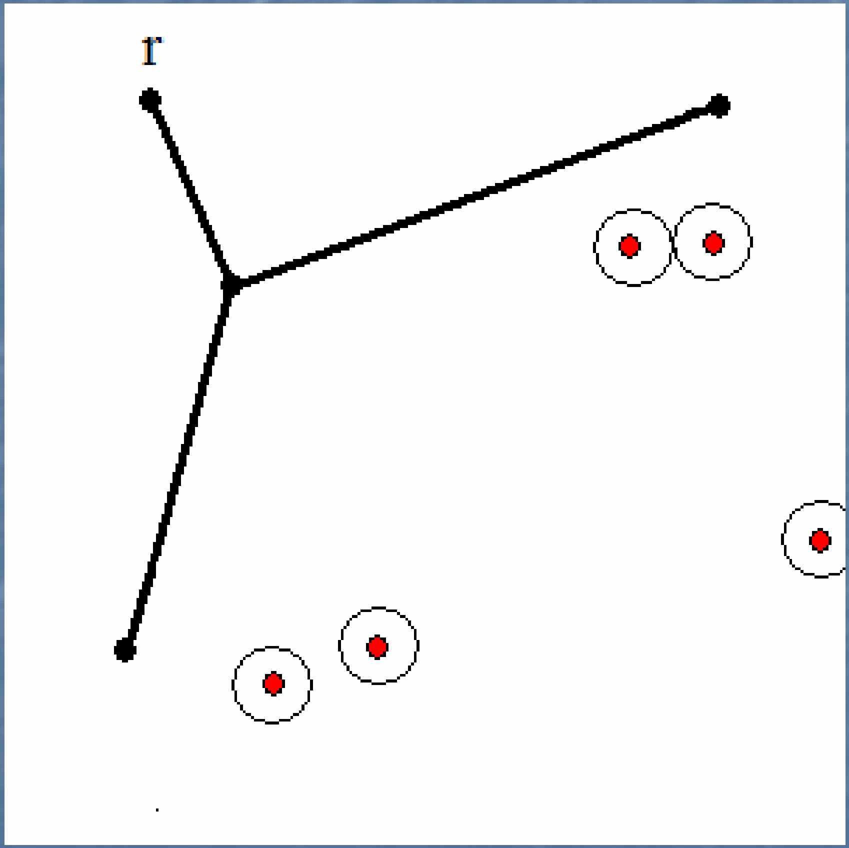
Second stage

- $T^0 \leftarrow$ 1st stage tree
- Consider scenario k
- Idea: Run Steiner tree primal-dual on terminals, stopping moat M when:
  - M hits T^0
  - M hits a stopped moat
  - For every terminal in M, less than ½ flow leaving M is 2nd-stage



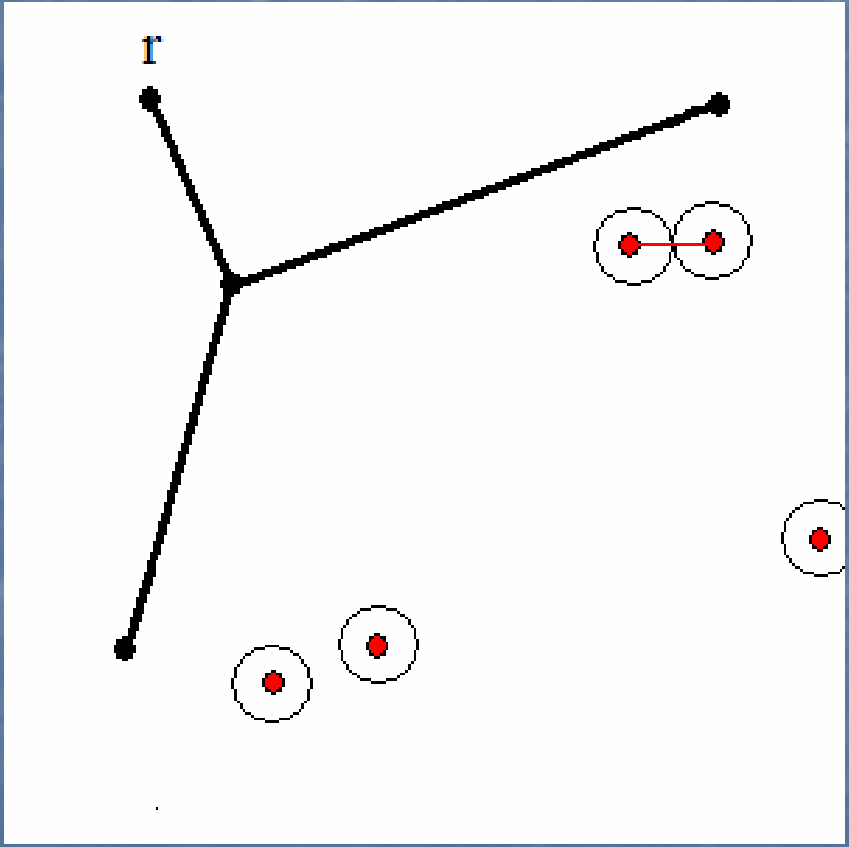
Second stage

- $T^0 \leftarrow$ 1st stage tree
- Consider scenario k
- Idea: Run Steiner tree primal-dual on terminals, stopping moat M when:
  - M hits T^0
  - M hits a stopped moat
  - For every terminal in M, less than ½ flow leaving M is 2nd-stage



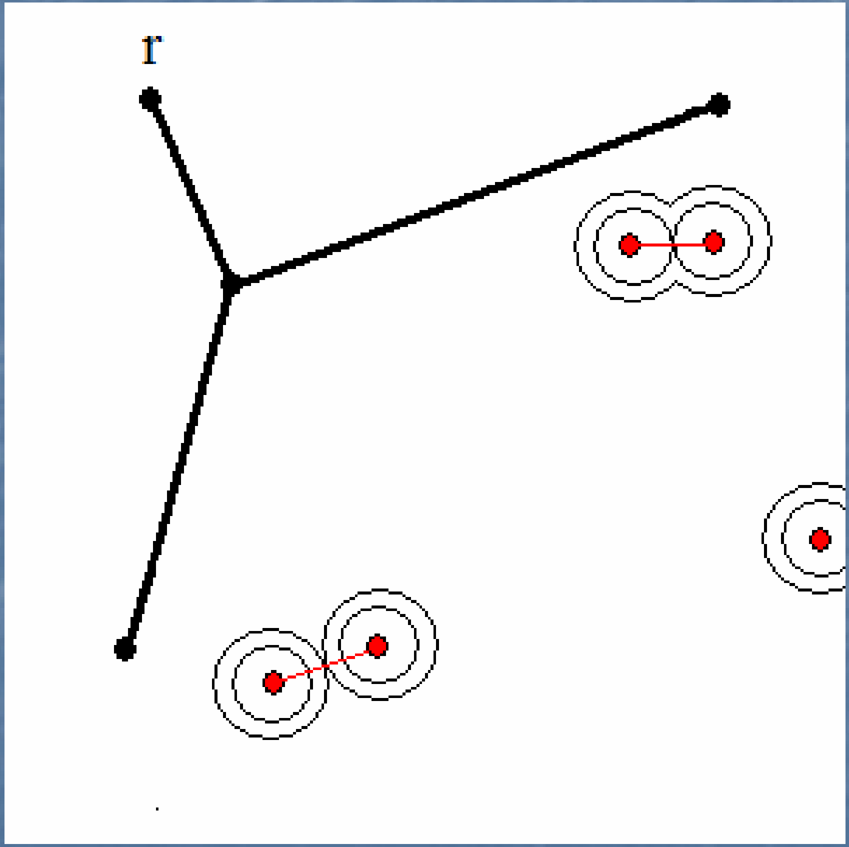
Second stage

- $T^0 \leftarrow 1^{st}$ stage tree
- Consider scenario k
- Idea: Run Steiner tree primal-dual on terminals, stopping moat M when:
  - M hits T^0
  - M hits a stopped moat
  - For every terminal in M, less than ½ flow leaving M is 2^{nd}-stage



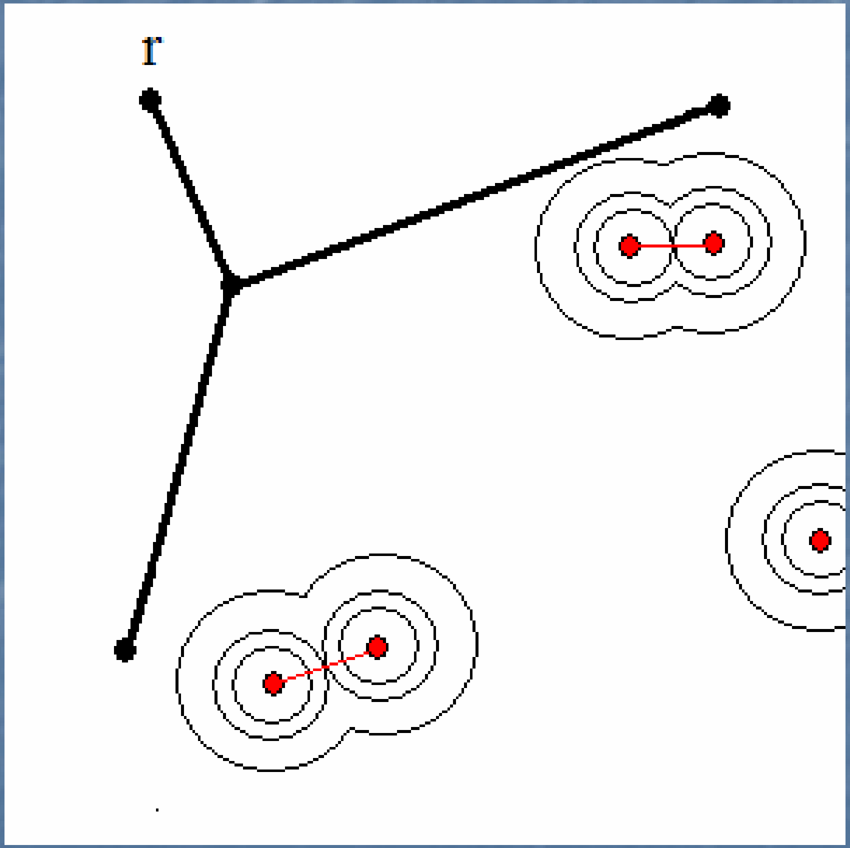
Second stage

- $T^0 \leftarrow 1^{st}$ stage tree
- Consider scenario k
- Idea: Run Steiner tree primal-dual on terminals, stopping moat M when:
  - M hits T^0
  - M hits a stopped moat
  - For every terminal in M, less than ½ flow leaving M is 2^{nd}-stage



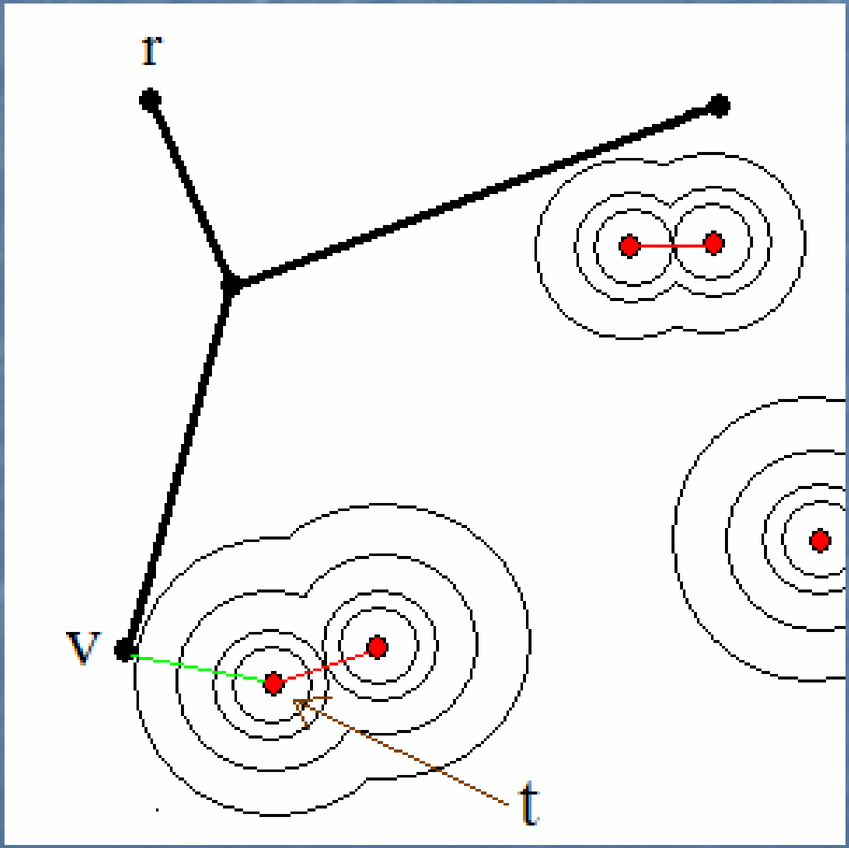
Second stage

- $T^0 \leftarrow 1^{st}$ stage tree
- Consider scenario *k*
- Idea: Run Steiner tree primal-dual on terminals, stopping moat *M* when:
  - *M* hits T^0
  - *M* hits a stopped moat
  - For every terminal in *M*, less than ½ flow leaving *M* is 2nd-stage



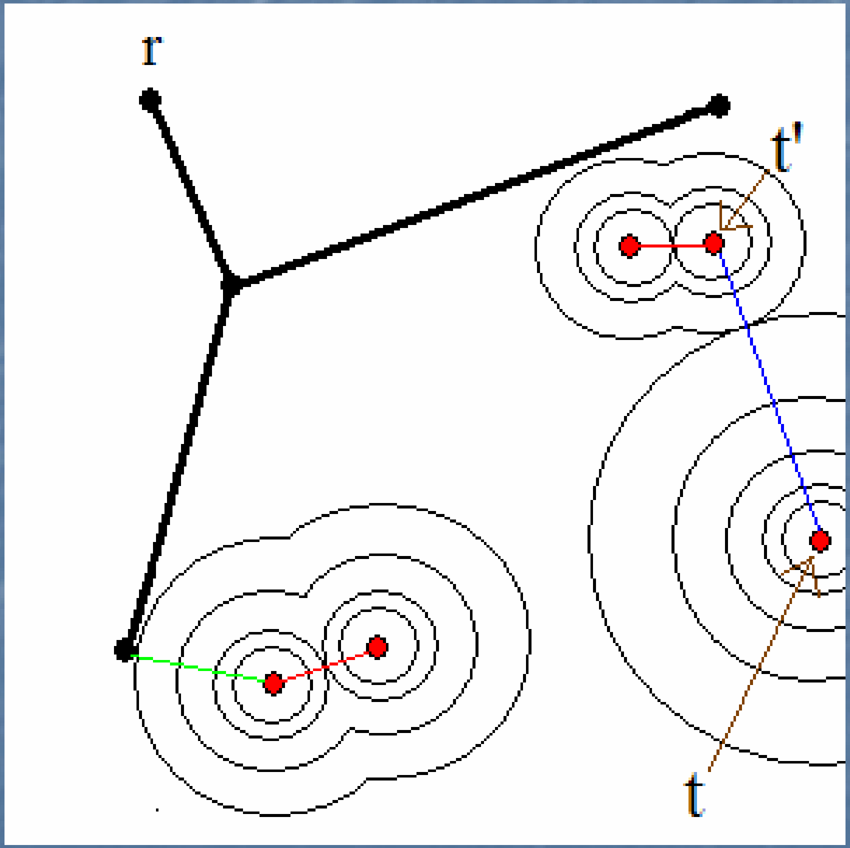
Second stage

- Idea: Run Steiner tree primal-dual on terminals, stopping moat M when:
  - M hits T^0
  - M hits a stopped moat
  - For every terminal in M, less than ½ flow leaving M is 2nd-stage
- If M hits T^0, add edge from $t \in M$ to $v \in R^0$



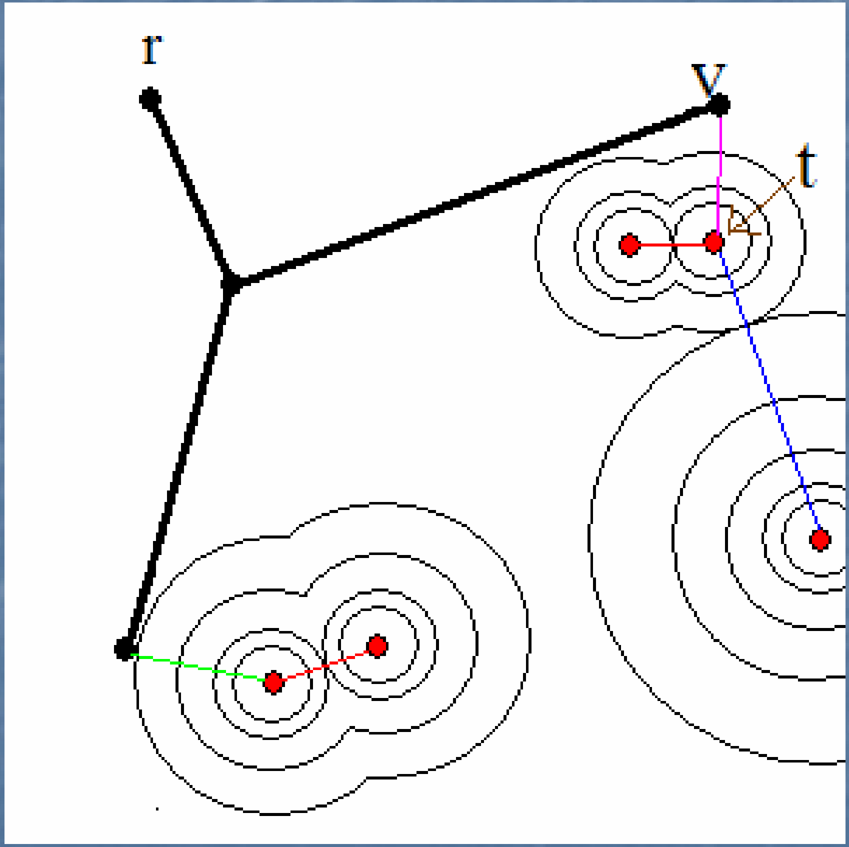
Second stage

- Idea: Run Steiner tree primal-dual on terminals, stopping moat *M* when:
  - *M* hits T^0
  - *M* hits a stopped moat
  - For every terminal in *M*, less than ½ flow leaving *M* is 2nd-stage
- If *M* hits *M′*, connect $t \in M$ with $t' \in M'$ as in Steiner tree primal-dual



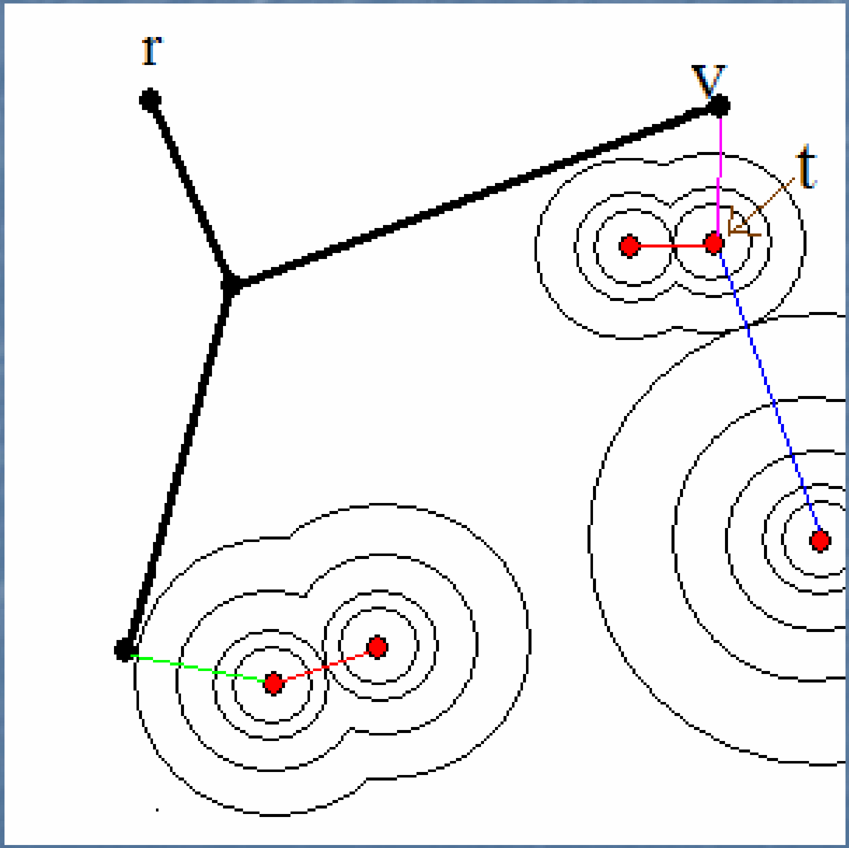
Second stage

- Idea: Run Steiner tree primal-dual on terminals, stopping moat *M* when:
  - *M* hits T^0
  - *M* hits a stopped moat
  - For every terminal in *M*, less than ½ flow leaving *M* is 2nd-stage
- There exists $t \in M$ and $v \in R^0$ s.t. *v* within $4 \times c.r.$ of *t*; connect *t* to *v*



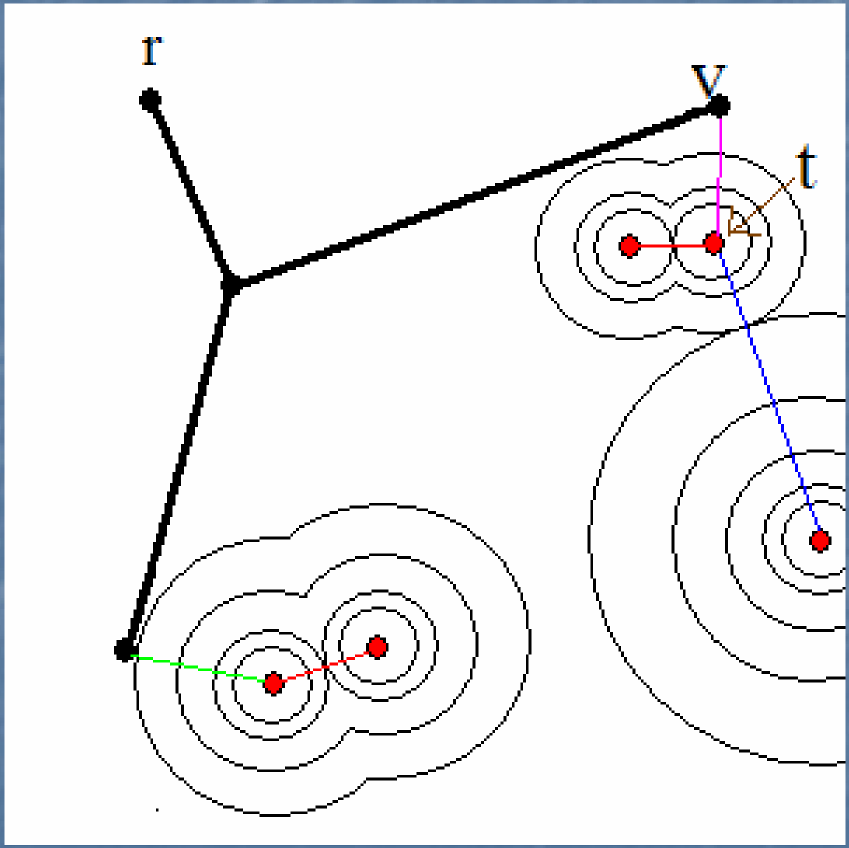
Second stage analysis

- Primal-dual accounts for edges inside moats



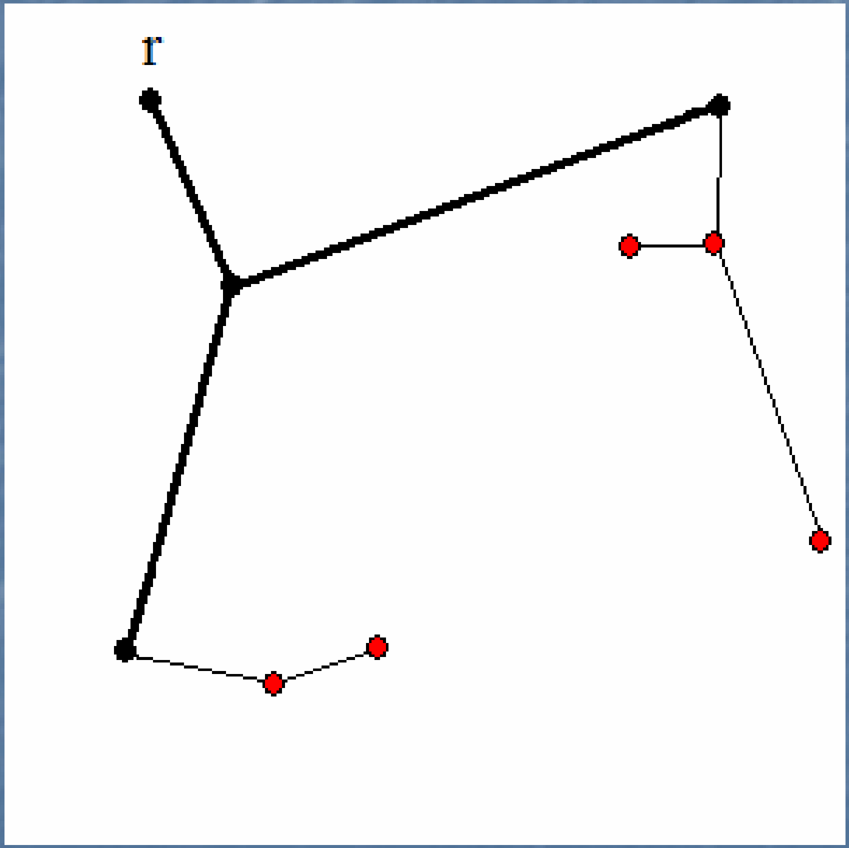
Second stage analysis

- Primal-dual accounts for edges inside moats
- Connector edges paid by carefully accounting:
  - Primal-dual bound
  - For every terminal *t*, there is $v \in R^0$ within $4 \times c.r.$ of t



SST: main result

- 24-approximation for Stochastic Steiner Tree (Improvement to 16-approx possible)
- Method: Primal-dual overlaid on LP solution
- Extensions to more general network design with routing costs
- Per-scenario risk-bounds incorporated and rounded



Main Techniques in other results

- Stochastic Facility Location – Rounding natural LP formulation using filter-and-round (Lin-Vitter, Shmoys-Tardos-Aardal) carefully [Details in IPCO '04]
- Stochastic Minimum Spanning Tree – Both scenario and black-box models - Randomized rounding of natural LP formulation gives nearly best possible O(log [No. of vertices] + log [max cost/min cost of an edge across scenarios]) approximation result [Details in IPCO '05]
- Multi-stage general covering problems – Boosted sampling with rejection based on ratio of scenario's inflation to maximum possible works [manuscript]

Summary

- Natural boosted sampling algorithm works for a broad class of stochastic problems in black-box model
- Boosted sampling with rejection extends to multi-stage covering problems in the black-box model
- Existing techniques can be cleverly adapted for the scenario model (E.g., LP-rounding for Facility location, primal-dual for Vertex Covers, combination of both for Steiner trees)
- Randomized rounding of LP formulations works for black-box formulation of spanning trees