Approximation algorithms for discrete stochastic optimization problems

David B. Shmoys
Cornell University

Stochastic Optimization

- Way of modeling uncertainty.
- Exact data is unavailable or expensive – data is uncertain, specified by a probability distribution.

Want to make the best decisions given this uncertainty in the data.

- Dates back to 1950's and the work of Dantzig.
- Applications in logistics, transportation models, financial instruments, network design, production planning, …

A priori optimization (no recourse)

Given: Probability distribution over inputs.

In advance: Compute master plan.

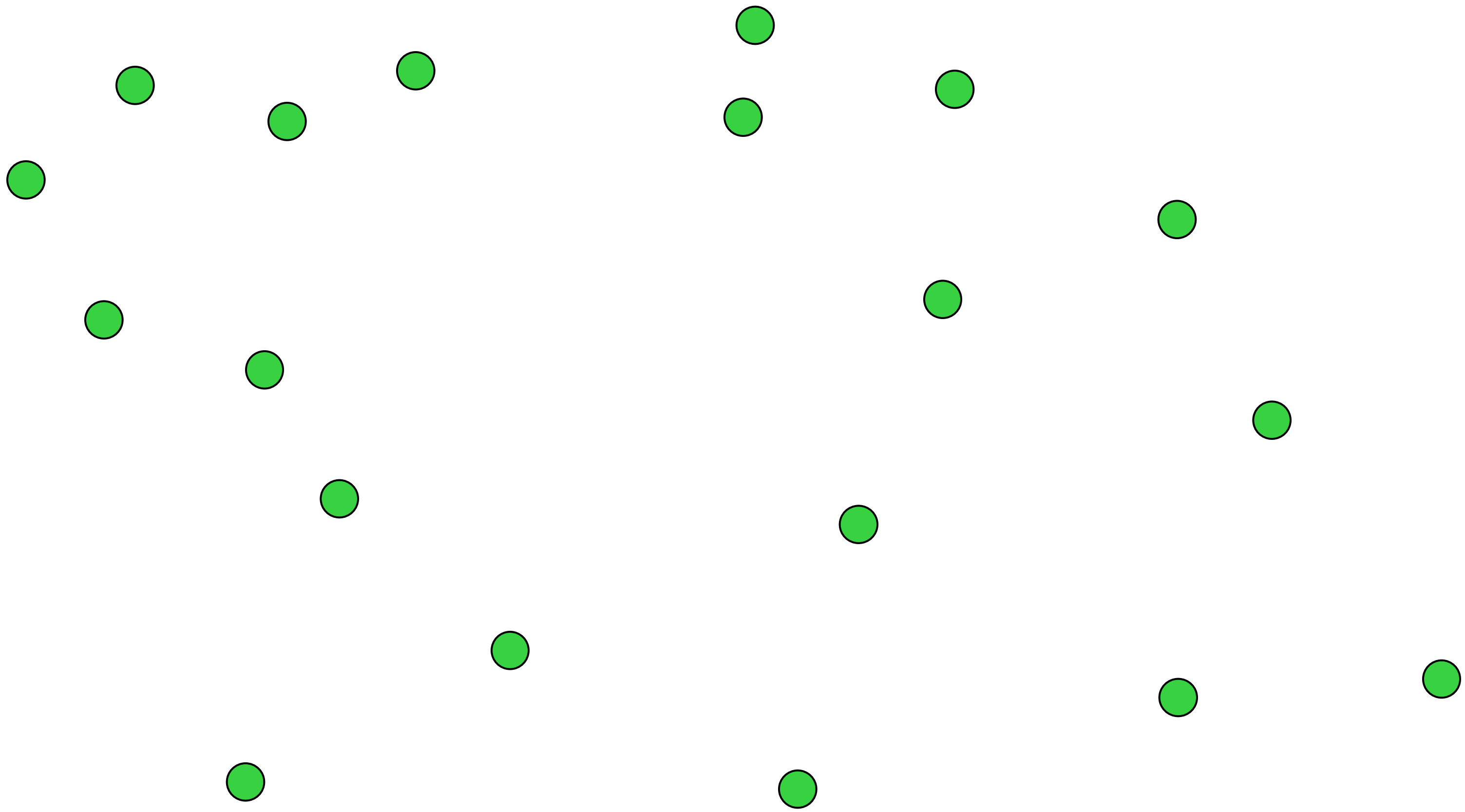
Observe the actual input scenario.

In real time: Adapt master plan to scenario.

Compute master plan to minimize

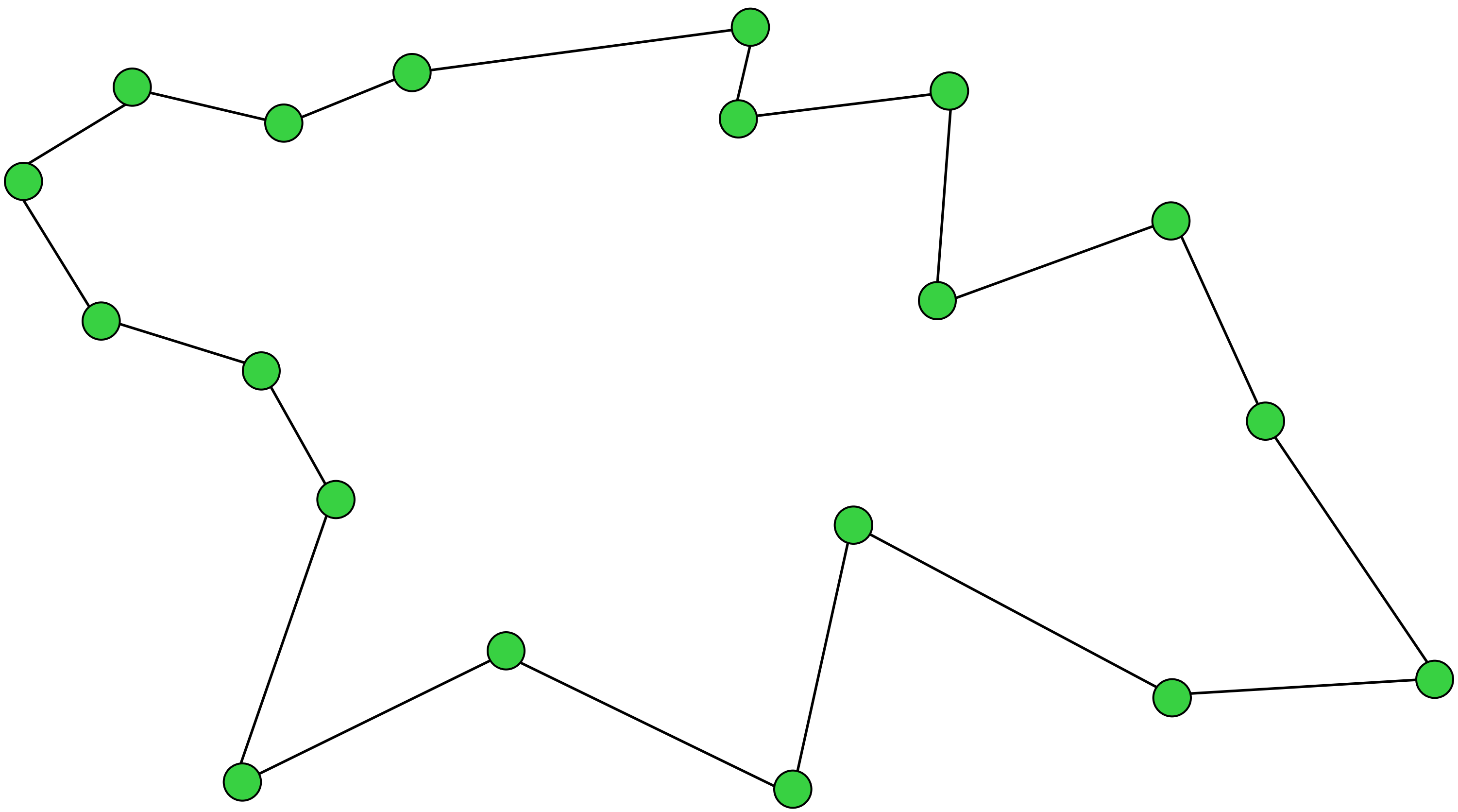
expected real time cost.

The Traveling Salesman Problem (TSP)



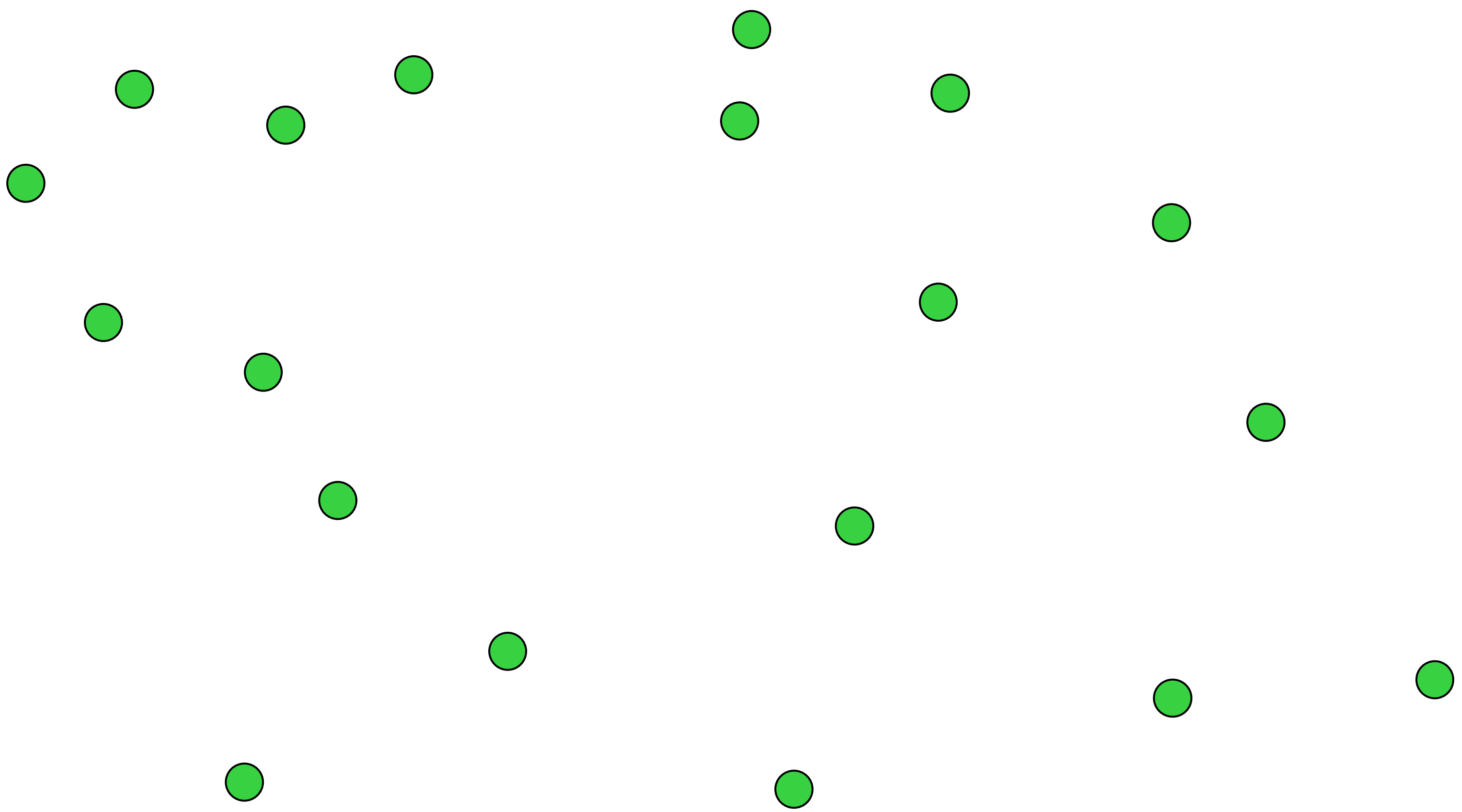
Given input points, compute tour τ to minimize total length $c(\tau)$

The Traveling Salesman Problem (TSP)



Given input points, compute tour τ to minimize total length $c(\tau)$

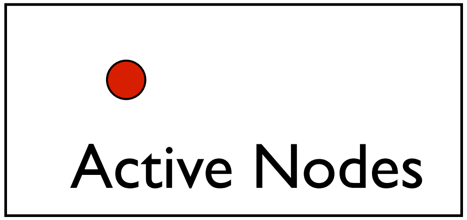
The *A Priori* TSP

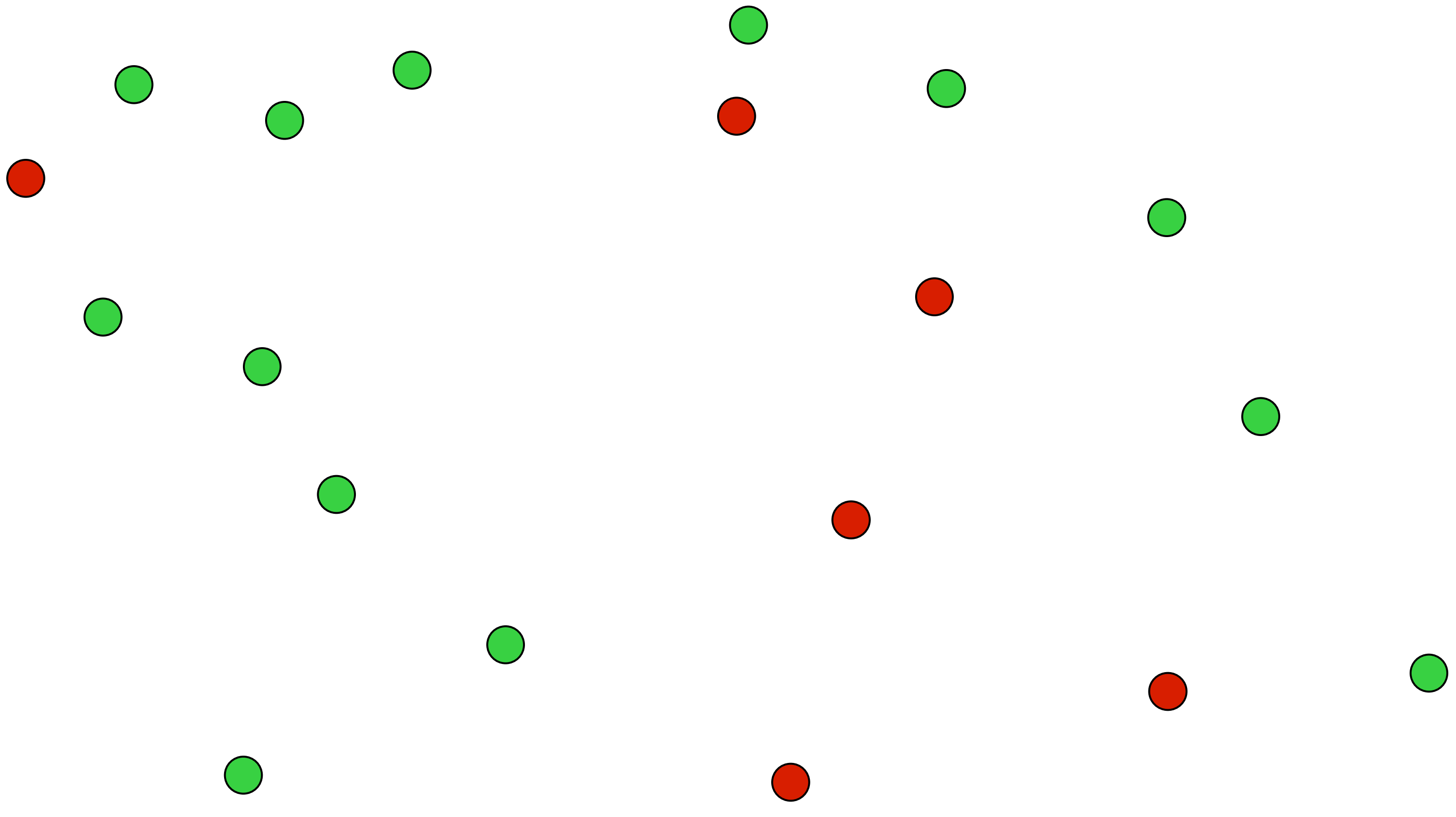


Given input points N and a distribution Π of active sets A 2 2^N

Need to specify the probability that a given set A is active

The *A Priori* TSP

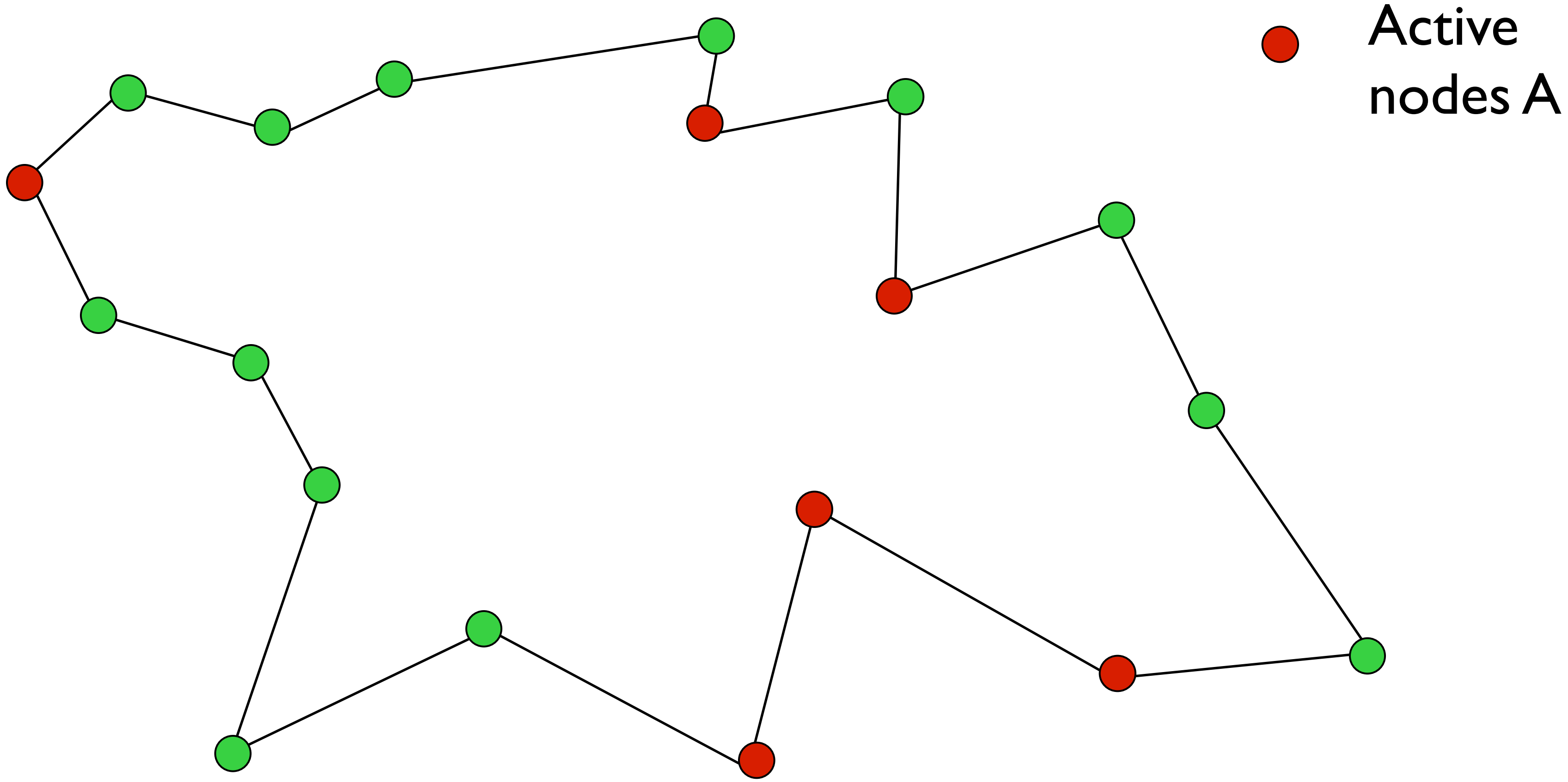




Given input points N and a distribution Π of active sets A 2 2^N

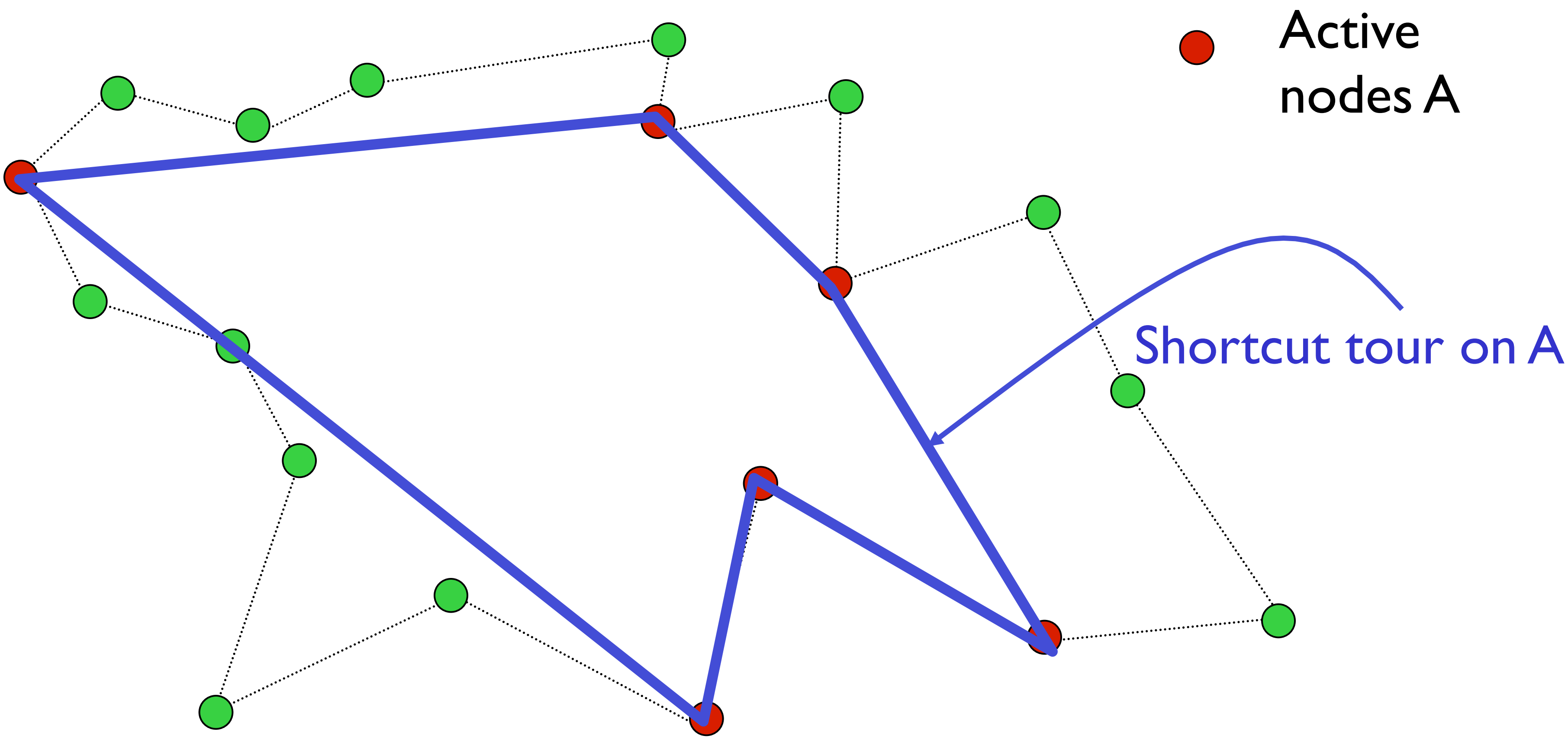
Need to specify the probability that a given set A is active

The *A Priori* TSP



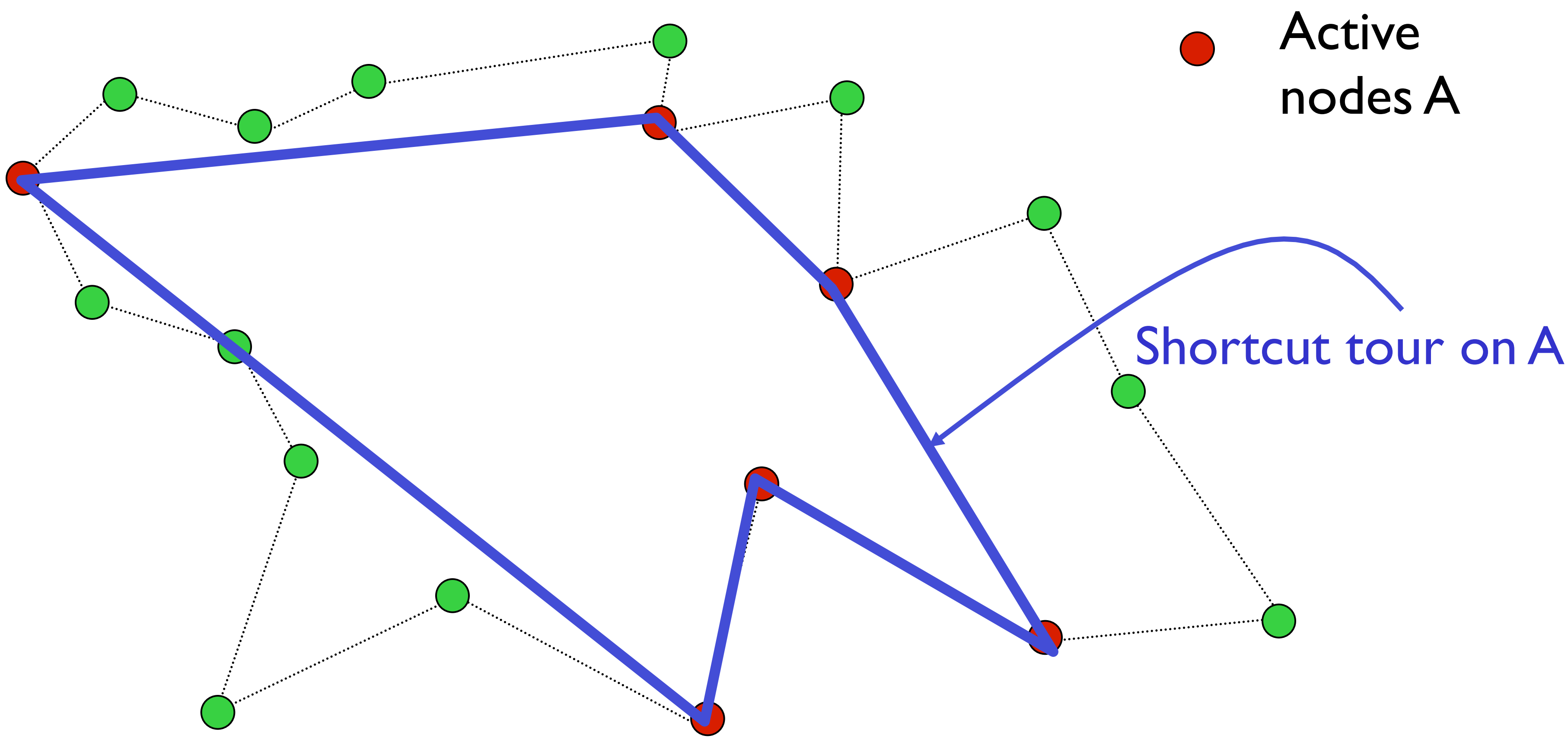
Given input points N and a distribution Π of active sets A 2 2^N, compute master tour τ to minimize expected length of the tour τ shortcut to serve only A

The *A Priori* TSP



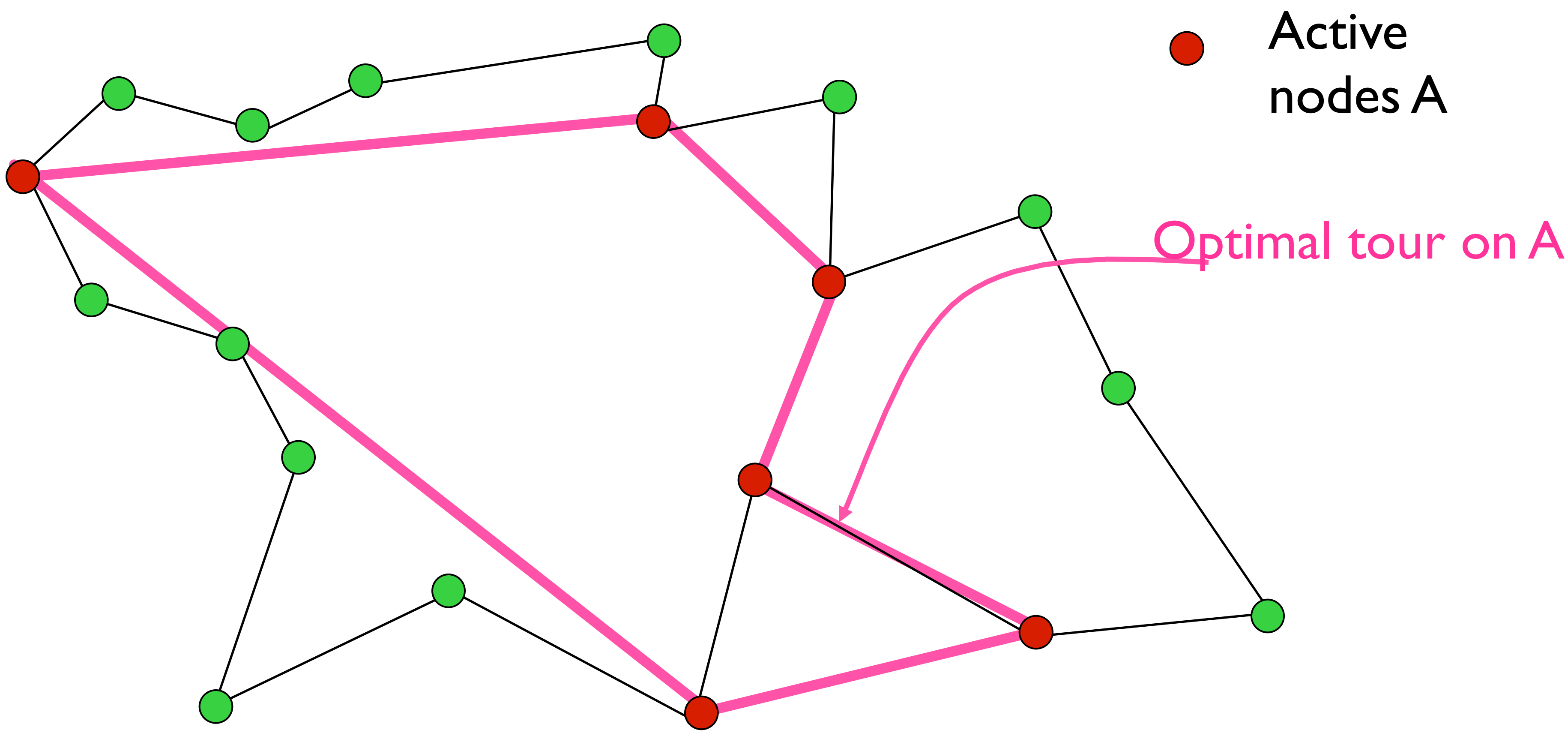
Given input points N and a distribution Π of active sets A 2 2^N, compute master tour τ to minimize expected length of the tour τ shortcut to serve only A

The *A Priori* TSP



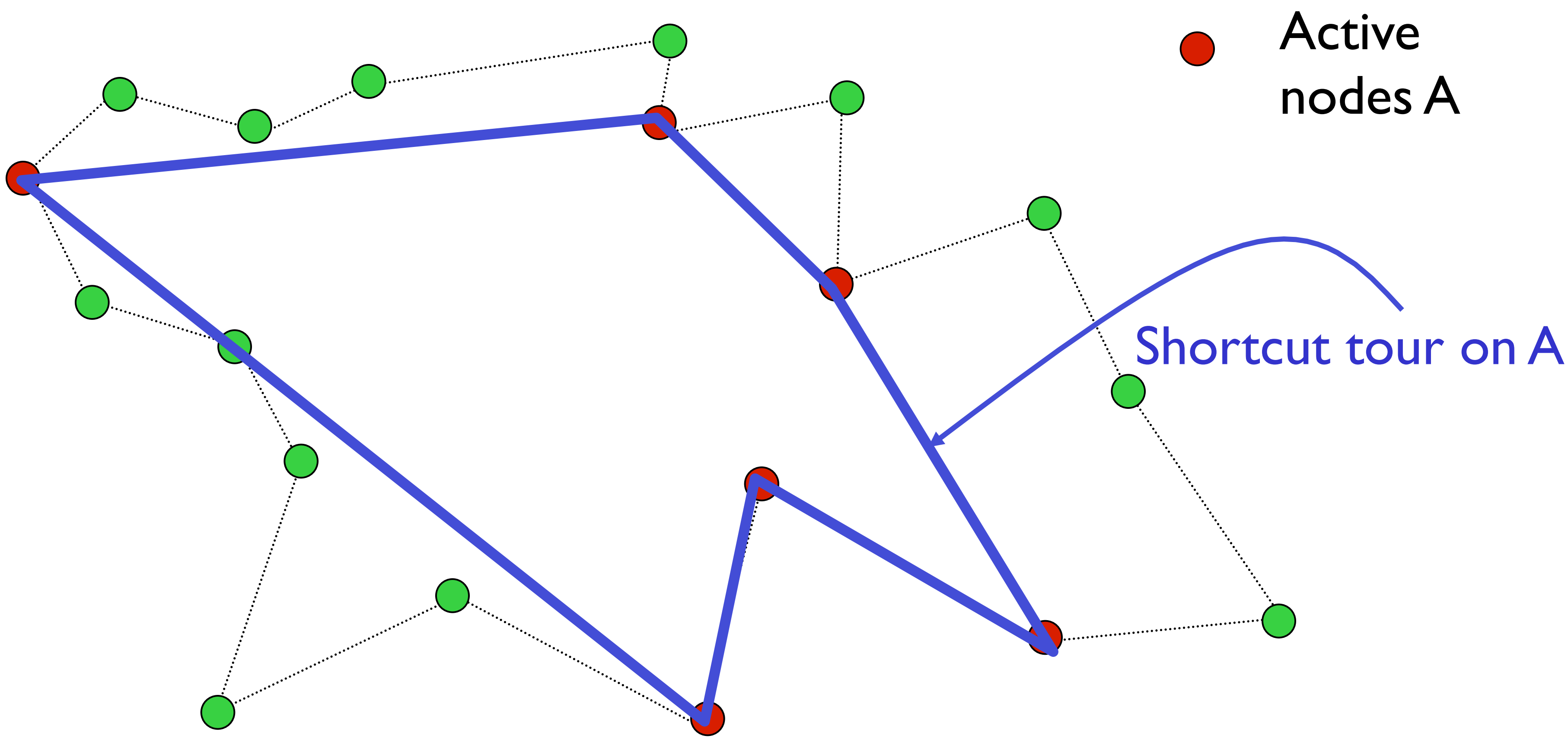
Given input points N and a distribution Π of active sets A 2 2^N, compute tour τ to minimize expected length $E_A\,[c(\,\tau_A\,)]$, where τ_A is the tour τ shortcut to serve only A

The *A Priori* TSP



Given input points N and a distribution Π of active sets A 2 2^N, compute tour τ to minimize expected length E_A [c(τ_A)], where τ_A is the tour τ shortcut to serve only A

The *A Priori* TSP



Given input points N and a distribution Π of active sets A 2 2^N, compute tour τ to minimize expected length $E_A [c(\tau_A)]$, where τ_A is the tour τ shortcut to serve only A

The *A Priori* TSP

Given input points N and a distribution Π of active sets A 2 2^N, compute tour ¿ to minimize expected length E_A [c(τ_A)], where τ_A is the tour τ shortcut to serve only A $\Rightarrow$ τ * (optimal solution)

Goal: Find tour τ such that E_A [c(τ_A)] $\leq$ ®E_A [c(τ *$_A$)] $\Rightarrow$ ®OPT

(This is an ®-approximation algorithm for the *a priori* TSP.)

How is the probability distribution on active set specified?

- A short (polynomial) list of possibile scenarios;
- Independent probabilities that each point is active;
- A black box that can be sampled.

The *A Priori* TSP

Given input points N and a distribution Π of active sets A 2 2^N, compute tour ¿ to minimize expected length E_A [c(τ_A)], where τ_A is the tour τ shortcut to serve only A $\Rightarrow$ τ * (optimal solution)

Goal: Find tour τ such that E_A [c(τ_A)] ≤ ®E_A [c(τ *$_A$)] $\Rightarrow$ ®OPT

(This is an ®-approximation algorithm for the *a priori* TSP.)

How is the probability distribution on active set specified?

- A short (polynomial) list of possibile scenarios;
- Independent probabilities that each point is active;
- A black box that can be sampled.

Some relevant history for *a priori* TSP

- Jaillet (1985, 1988), Bertsimas (1988), Jaillet, Bertsimas, & Odoni (1990) introduce problem – analyze with probabilistic assumptions on distances
- Schalekamp & S (2007) randomized O(log n)-approximation
- Maybecast problem Karger & Minkoff (2000)
- Rent-or-buy problem Gupta, Kumar, Pál, Roughgarden (2007)
- Stochastic Steiner Tree variants Gupta, Pál, Ravi, Sinha (2004)
  Gupta, Ravi, Sinha ('04), Hayraptian, Swamy, Tardos ('05)
  Garg, Gupta, Leonardi, Sankowski (2008)
- Universal TSP Bartholdi & Plazman (1989), Jia, Lin, Noubir, Rajaraman & Sundaram, (2005), Hajiaghayi, Kleinberg & Leighton (2006), Gupta, Hajiaghayi, Räcke (2006)

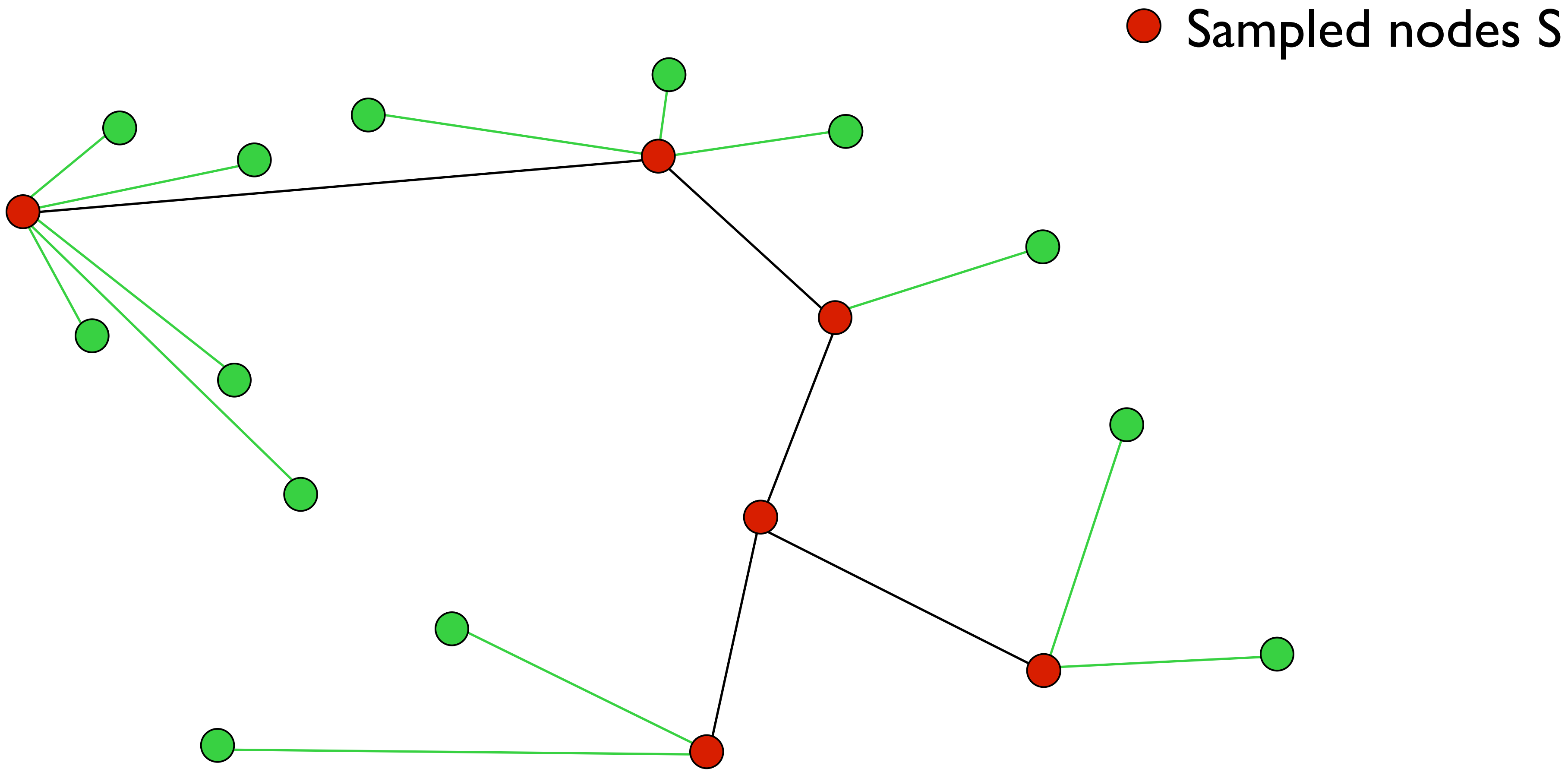
The One Random Sample Algorithm

1. Draw sample $S \subseteq N$ according to Π (i.e., pick each point j independently with probability p_j)
2. Build minimum spanning tree on S
3. For each $j \notin S$, connect j to its nearest neighbor in S
4. Build "double tree" tour of this tree $\Rightarrow \tau$

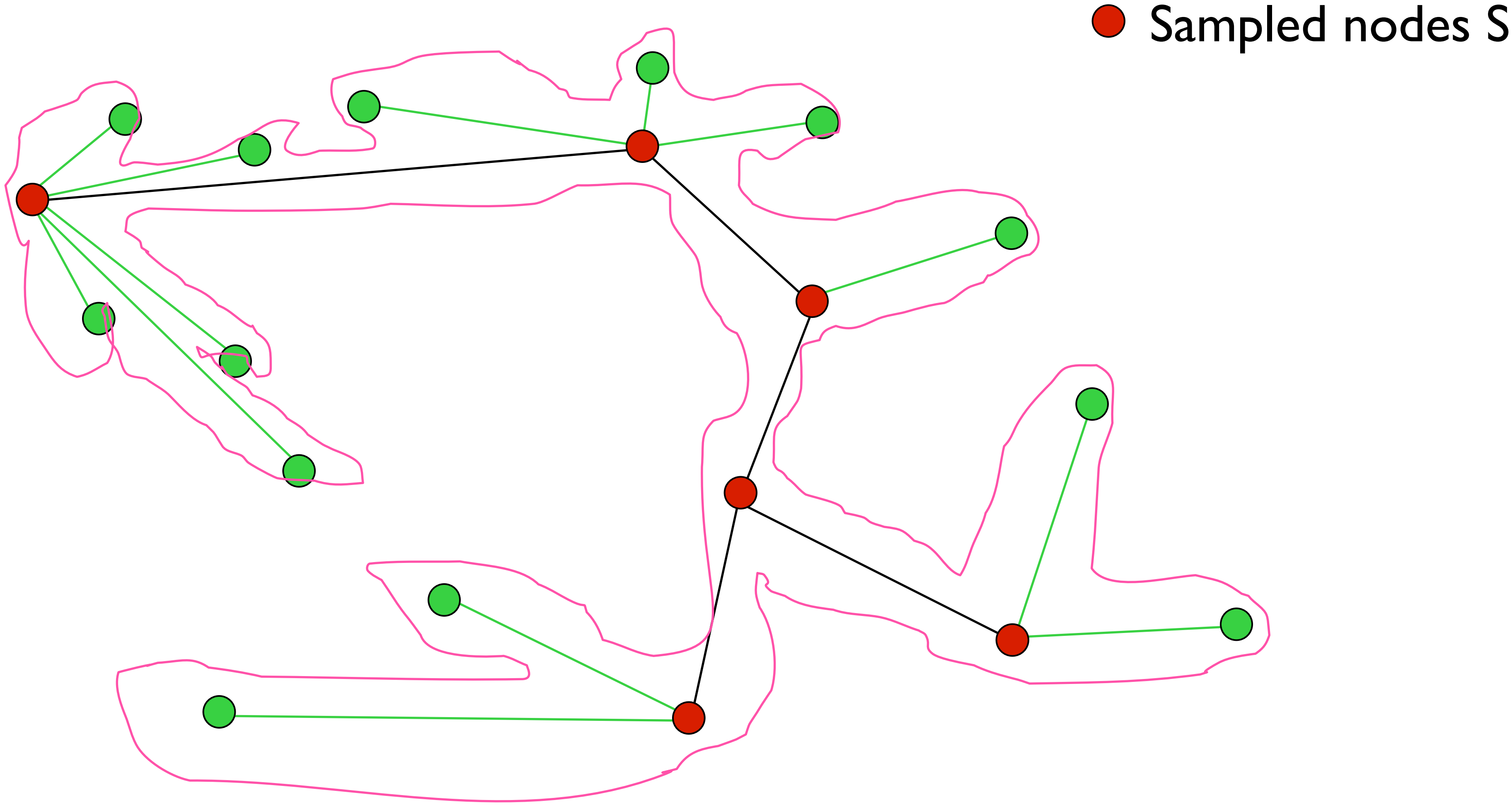
Simplifying Assumption: $\exists$ node r with $p_r = 1$ (wlog)

Theorem The one random sample algorithm is a 4-approximation algorithm for the *a priori* TSP.

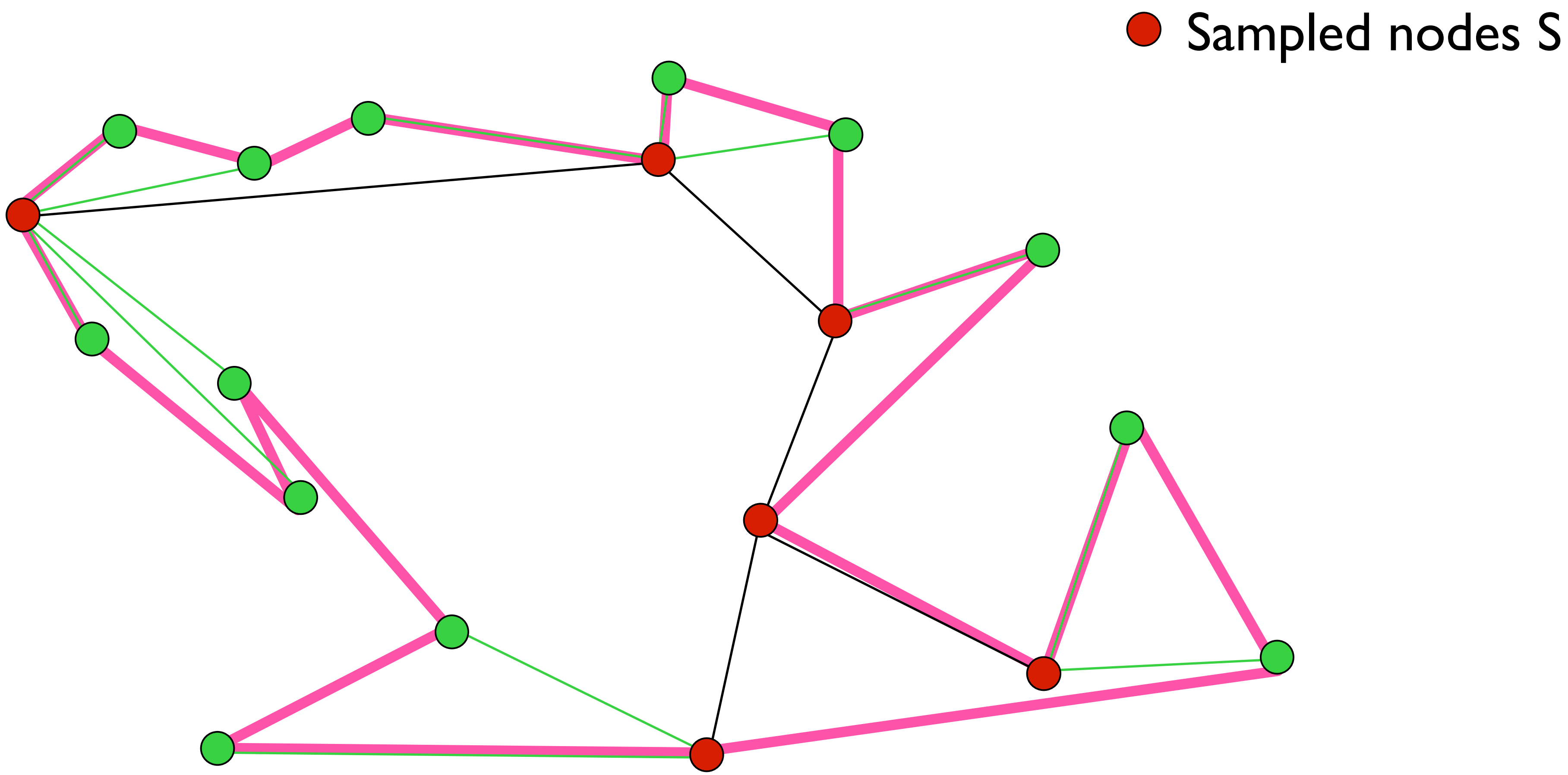
Running the Algorithm



Running the Algorithm



Running the Algorithm



Analyzing the Algorithm

Let $D_j(S)$ be the distance from j to its nearest neighbor in S-{j}

Let MST(S) be the length of the minimum spanning tree on S

Goal: Analyze $E_S [E_A [c(\tau_A)]]$

Fact 1. $E_S [D_j(S)] = E_S [D_j | j \notin S] = E_S [D_j \mid j \in S] = E_A [D_j(A) | j \in A]$

Why? Choice of S-{j} is independent of whether j 2 S, and
S and A are independent draws from same distribution

Fact 2. MST(A) ≤ c(¿ $^*_A$) for each A µ N

Why? Tour ¿* shortcut to A still contains spanning tree

Fact 3. $\sum_{j \neq r} 1(j \in A)\, D_j(A) \le c(\tau^*_A)$ for all A

Why? Any tour on A "leaves" each node i by some edge

Let $D_j(S)$ be the distance from j to its nearest neighbor in S-{j}

Let MST(S) be the length of the minimum spanning tree on S

Goal: analyze $E_S [E_A [c(\tau_A)]]$

Fact 1. $E_S [D_j(S)] = E_S [D_j | j \notin S] = E_S [D_j \mid j \in S] = E_A [D_j(A) | j \in A]$

Fact 2. $MST(A) \leq c(\tau^*_A)$ for all A

Fact 3. $\sum_{j \neq r} 1(j \in A)\, D_j(A) \leq c(\tau^*_A)$ for all A

Key Idea: always pay for backbone built on S (for any active A)

$$E_S[E_A[c(\tau_A)]] \leq E_S[2MST(S)]+E_S [E_A [\sum_{j \neq r} 1(j \in A)\, 1(j \notin S)\, 2D_j(S)]]$$

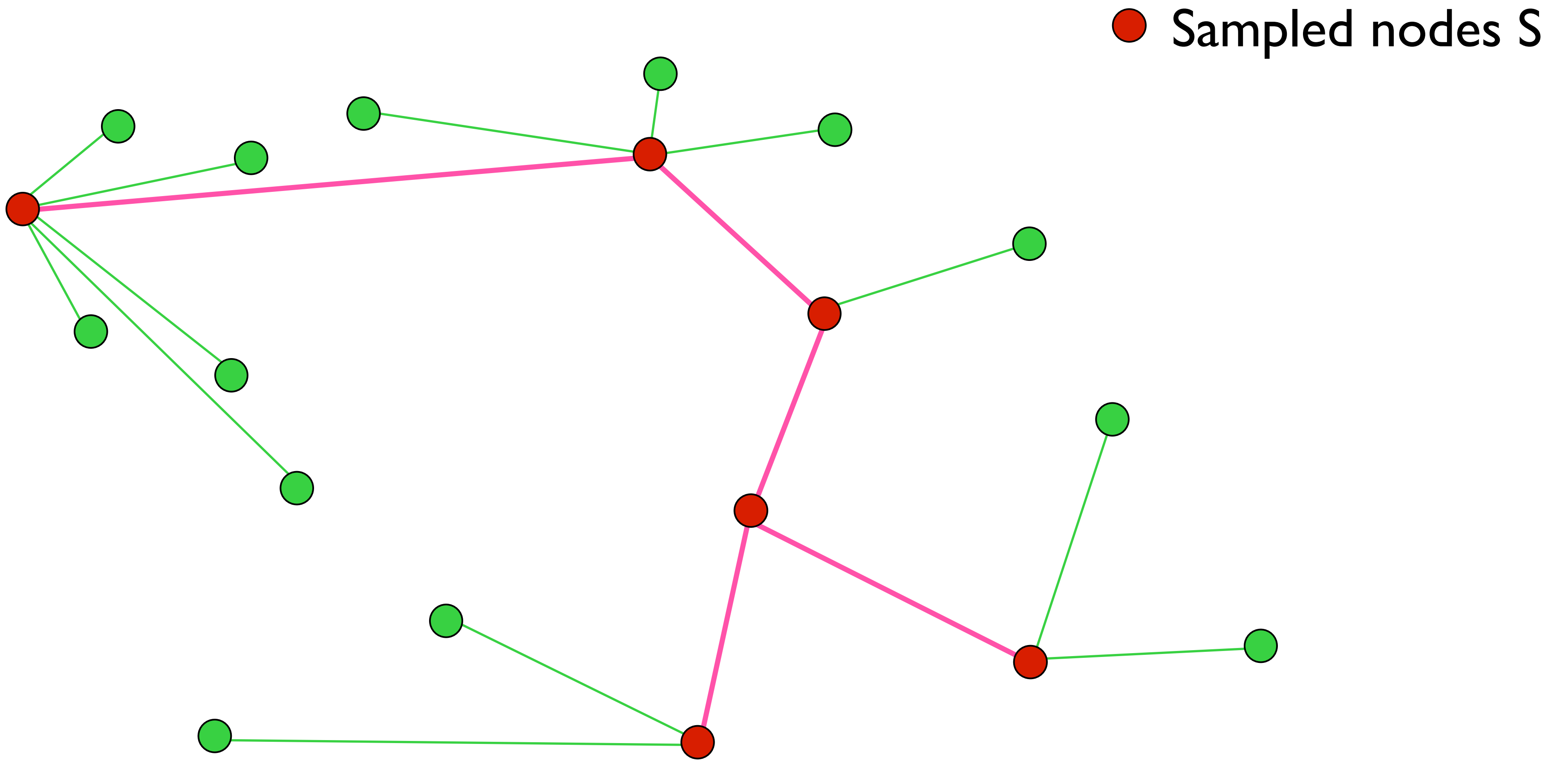
$$= E_S [2MST(S)] + \sum_{j \neq r} E_{S,A} [1(j \in A) 1(j \notin S)\, 2D_j (S)]$$

$$= E_S [2MST(S)] + 2\sum_{j \neq r} p_j (1-p_j)\, E_S[D_j(S)]$$

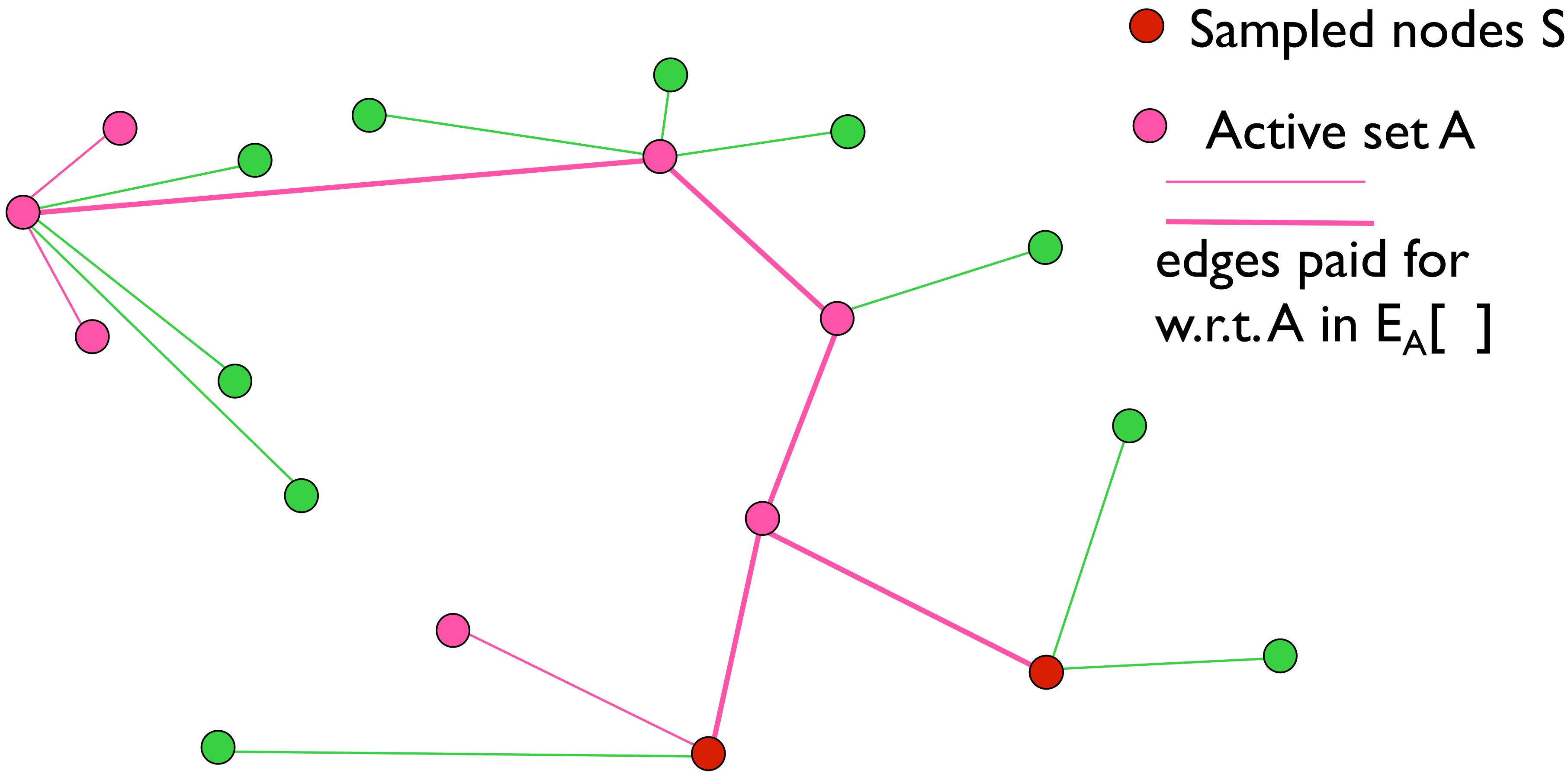
$$\leq 2(E_S [MST(S)] + \sum_{j \neq r} p_j\, E_S[D_j(S)])$$

$$\leq 2 (OPT+OPT)$$

Analyzing the Algorithm



Analyzing the Algorithm



Always pay for all of backbone and just those attached leaves you need

Cost of shortcut tour for A is at most twice the cost of these edges

Let $D_j(S)$ be the distance from j to its nearest neighbor in S-{j}

Let MST(S) be the length of the minimum spanning tree on S

Goal: analyze $E_S [E_A [c(\tau_A)]]$

Fact 1. $E_S [D_j(S)] = E_S [D_j | j \notin S] = E_S [D_j | j \in S] = E_A [D_j(A) | j \in A]$

Fact 2. $MST(A) \leq c(\tau^*_A)$ for all A

Fact 3. $\sum_{j \neq r} 1(j \in A) D_j(A) \leq c(\tau^*_A)$ for all A

Key Idea: always pay for backbone built on S (for any active A)

$$E_S[E_A[c(\tau_A)]] \leq E_S[2MST(S)]+E_S [E_A [\sum_{j \neq r} 1(j \in A)\ 1(j \notin S)\ 2D_j(S)]]$$

$$= E_S [2MST(S)] + \sum_{j \neq r} E_{S,A} [1(j \in A) 1(j \notin S)\ 2D_j (S)]$$

$$= E_S [2MST(S)] + 2\sum_{j \neq r} p_j (1-p_j) E_S[D_j(S)]$$

$$\leq 2(E_S [MST(S)] + \sum_{j \neq r} p_j E_S[D_j(S)])$$

$$\leq 2 (OPT+OPT)$$

Let $D_j(S)$ be the distance from j to its nearest neighbor in S-{j}

Let MST(S) be the length of the minimum spanning tree on S

Goal: analyze $E_S\,[\,E_A\,[c(\tau_A)]]$

Fact 1. $E_S\,[D_j(S)] = E_S\,[D_j |\, j \notin S] = E_S\,[D_j \mid j \in S] = E_A\,[D_j(A) |\, j \in A]$

Fact 2. $MST(A) \le c(\tau^*_A)$) $E_A[MST(A)] \cdot OPT$

Fact 3. $\sum_{j\neq r} 1(j \in A)\, D_j(A) \le c(\tau^*_A)$) $\sum_{j\neq r} p_j\, E_A[D_j(A)] \cdot OPT$

Key Idea: always pay for backbone built on S (for any active A)

$E_S[E_A[\,c(\tau_A)]] \le E_S[\,2MST(S)\,] + E_S\,[\,E_A\,[\,\sum_{j\neq r} 1(j \in A)\ 1(j \notin S)\ 2D_j(S)]]$

$= E_S\,[\,2MST(S)\,] + \sum_{j\neq r} E_{S,A}\,[1(j \in A) 1(j \notin S)\ 2D_j\,(S)]$

$= E_S\,[2MST(S)] + 2\sum_{j\neq r} p_j\,(1\text{-}p_j)\, E_S[D_j(S)]$

$\le 2(\,E_S\,[MST(S)] + \sum_{j\neq r} p_j\, E_S[D_j(S)])$

$\le 2\,(OPT+OPT)$

The One Random Sample Algorithm

1. Draw sample $S \subseteq N$ according to Π (i.e., pick each point j independently with probability p_j)
2. Build minimum spanning tree on S
3. For each $j \notin S$, connect j to its nearest neighbor in S
4. Build "double tree" tour of this tree $\Rightarrow \tau$

Simplifying Assumption: $\exists$ node r with $p_r = 1$ (wlog)

Theorem (S & Talwar) The one random sample algorithm is a 4-approximation algorithm for the *a priori* TSP.

Two Footnotes

Can be derandomized -

Analyzing the Algorithm

Let $D_j(S)$ be the distance from j to its nearest neighbor in S-{j}

Let MST(S) be the length of the minimum spanning tree on S

Goal: analyze $E_S [E_A [c(\tau_A)]]$

Note: $E_S [D_j(S)] = E_S [D_j \mid j \notin S] = E_S [D_j \mid j \in S] = E_A [D_j (A) \mid j \in A]$

$MST(A) \leq c(\tau^*_A)$ for all A

$\sum_{j \neq r} 1(j \in A) D_j(A) \leq c(\tau^*_A)$ for all A

$E_S[E_A[c(\tau_A)]] \leq E_S[2MST(S)]+E_S [E_A [\sum_{j \neq r} 1(j \in A) 1(j \notin S) 2D_j(S)]]$

$= E_S [2MST(S)] + \sum_{j\neq r} E_{S,A} [1(j \in A)1(j \notin S) 2D_j (S)]$

$= E_S [2MST(S)] + 2\sum_{j \neq r} p_j (1-p_j) E_S[D_j(S)]$

$\leq 2(E_S [MST(S)] + \sum_{j \neq r} p_j E_S[D_j(S)]) \leq 2 (OPT+OPT)$

Let $D_j(S)$ be the distance from j to its nearest neighbor in S-{j}

Let MST(S) be the length of the minimum spanning tree on S

Goal: analyze $E_S [E_A [c(\tau_A)]]$

Fact 1. $E_S [D_j(S)] = E_S [D_j | j \notin S] = E_S [D_j \mid j \in S] = E_A [D_j(A) \mid j \in A]$

Fact 2. $MST(A) \leq c(\tau^*_A)$ for all A

Fact 3. $\sum_{j \neq r} 1(j \in A) D_j(A) \leq c(\tau^*_A)$ for all A

Key Idea: always pay for backbone built on S (for any active A)

$E_S[E_A[c(\tau_A)]] \leq E_S[2MST(S)]+E_S [E_A [\sum_{j \neq r} 1(j \in A)\, 1(j \notin S)\, 2D_j(S)]]$

$= E_S [2MST(S)] + \sum_{j\neq r} E_{S,A} [1(j \in A)1(j \notin S)\, 2D_j (S)]$

$= E_S [2MST(S)] + 2\sum_{j \neq r} p_j (1-p_j) E_S[D_j(S)]$

$\leq 2(E_S [MST(S)] + \sum_{j \neq r} p_j E_S[D_j(S)])$

≤ 2 (OPT+OPT)

Two Footnotes

Can be derandomized – Williamson & van Zuylen (2007) show how to deterministically achieve twice guarantee for rent-or-buy/connected facility location problem by the method of conditional probabilities (by an LP estimate)

Assumption that $p_r = 1$ is not needed;

Need only that $D_j(S)$ is well defined.

Modify Π to condition on that each set has cardinality ≥ 2

Can sample according to this new distribution also, and this just rescales things (any tour has cost 0 restricted to 0 or 1 points) but must be careful about dependence

Theorem (S & Talwar) There is a deterministic 8-approximation algorithm for the a priori TSP in the independent activation model

What about the black box model?

Recent work of Gorodezky, R. Kleinberg, S, & Spencer shows that for a (slightly) restricted class of algorithms can embed a universal computation in an a priori one, and thereby show a non-constant lower bound on performance guarantees possible with a polynomial number of samples

Two-Stage Recourse Model

Given : Probability distribution over inputs.

Stage I : Make some advance decisions – plan ahead or hedge against uncertainty.

Observe the actual input scenario.

Stage II : Take recourse. Can augment earlier solution paying a recourse cost.

Choose stage I decisions to minimize

(stage I cost) + (expected stage II recourse cost).

2-Stage Steiner Tree Problem

Given a set of points N (with root) in a metric space, integer inflation factor λ, and distribution over 2^N

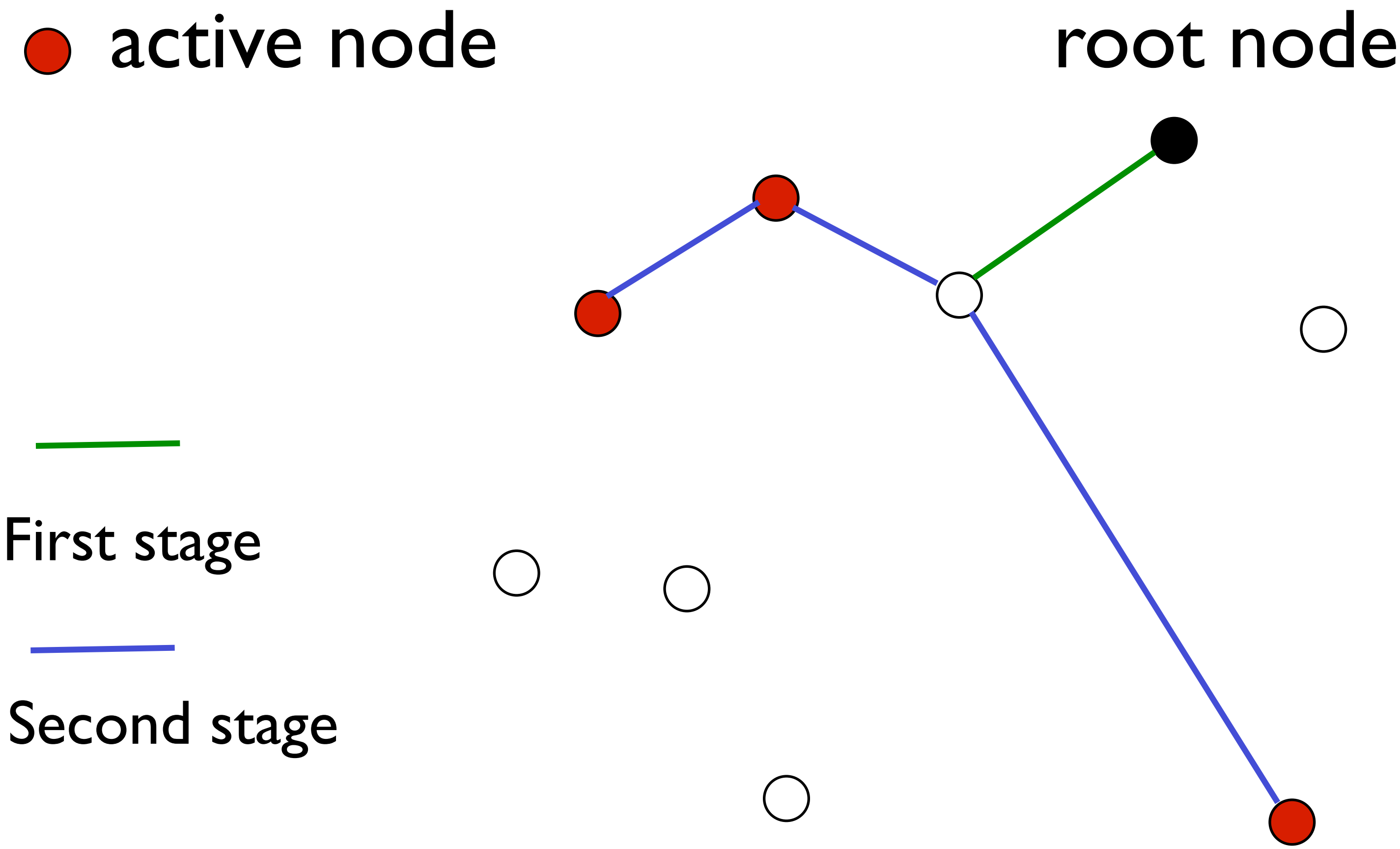
Stage I: install edges A_I – cost of e is c_e

Set of active terminals $T \subseteq N$ is selected (including root)

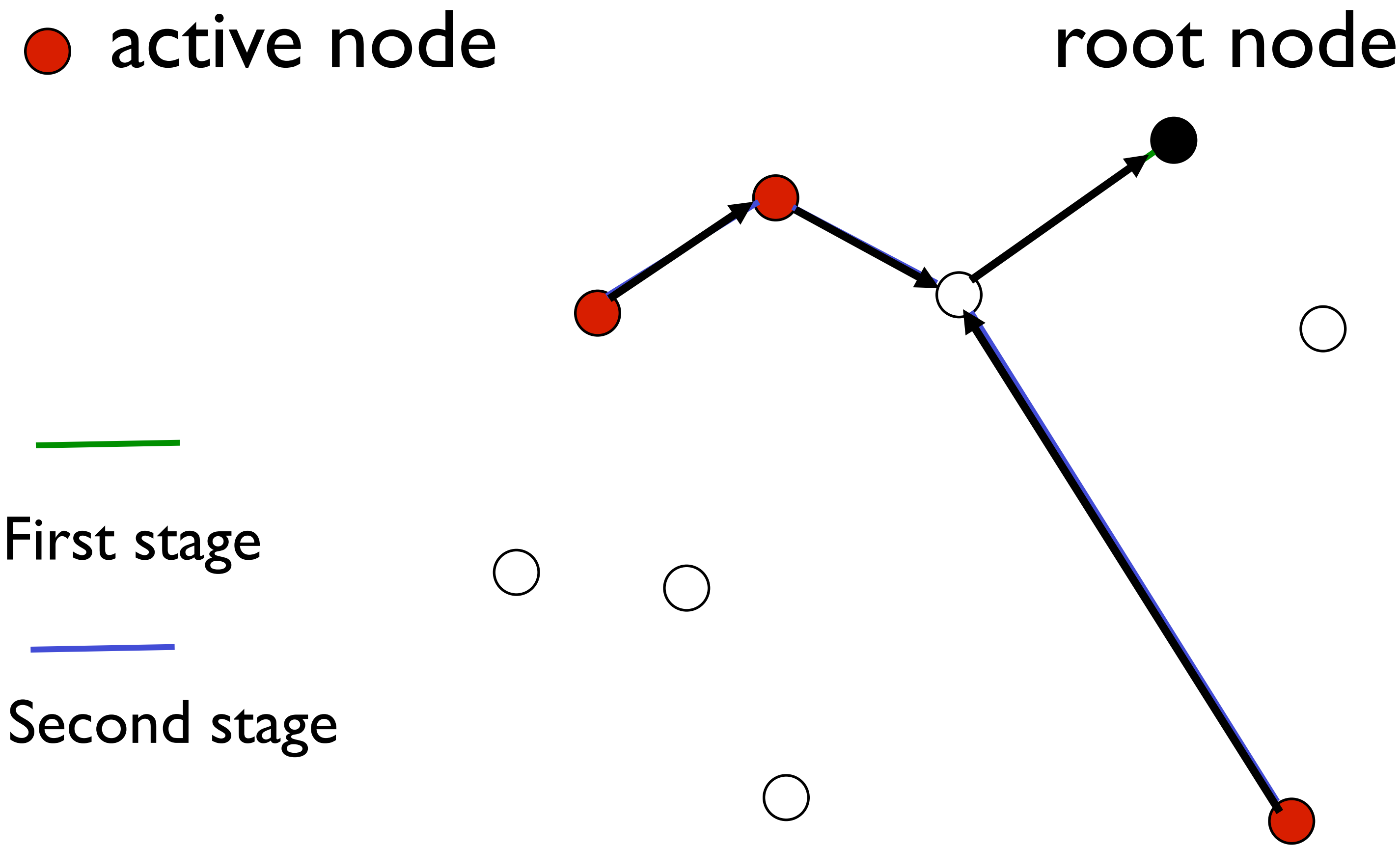
Stage II: install edges A_{II} s.t. $A_I \cup A_{II}$ is Steiner tree on T - cost of edge e is λc_e

Goal: Minimize
(cost of edges installed in stage I) +
$\lambda\, \mathbf{E}_{T \subseteq N}$ [cost of edges installed for scenario T].

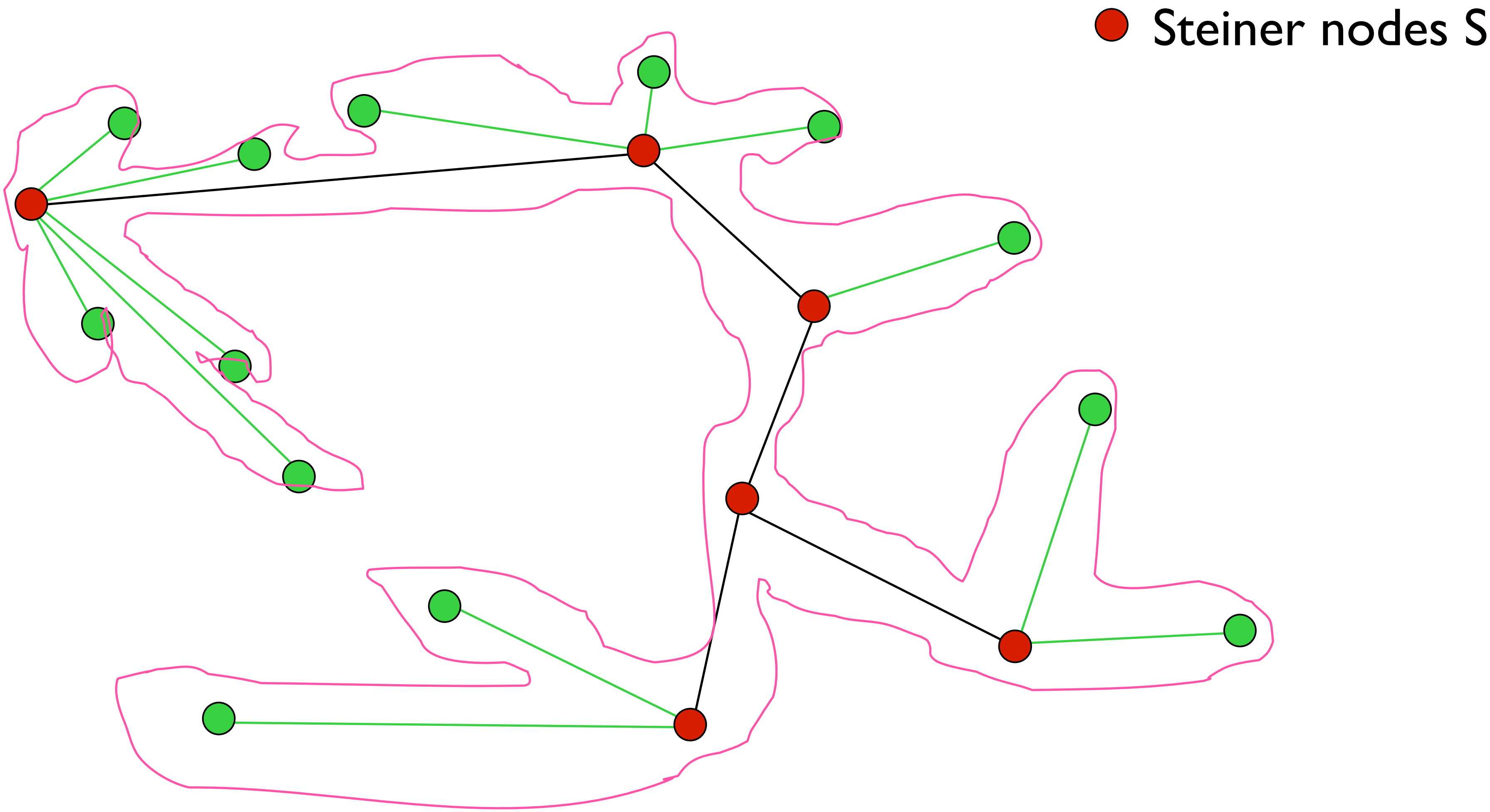
An Example



An Example

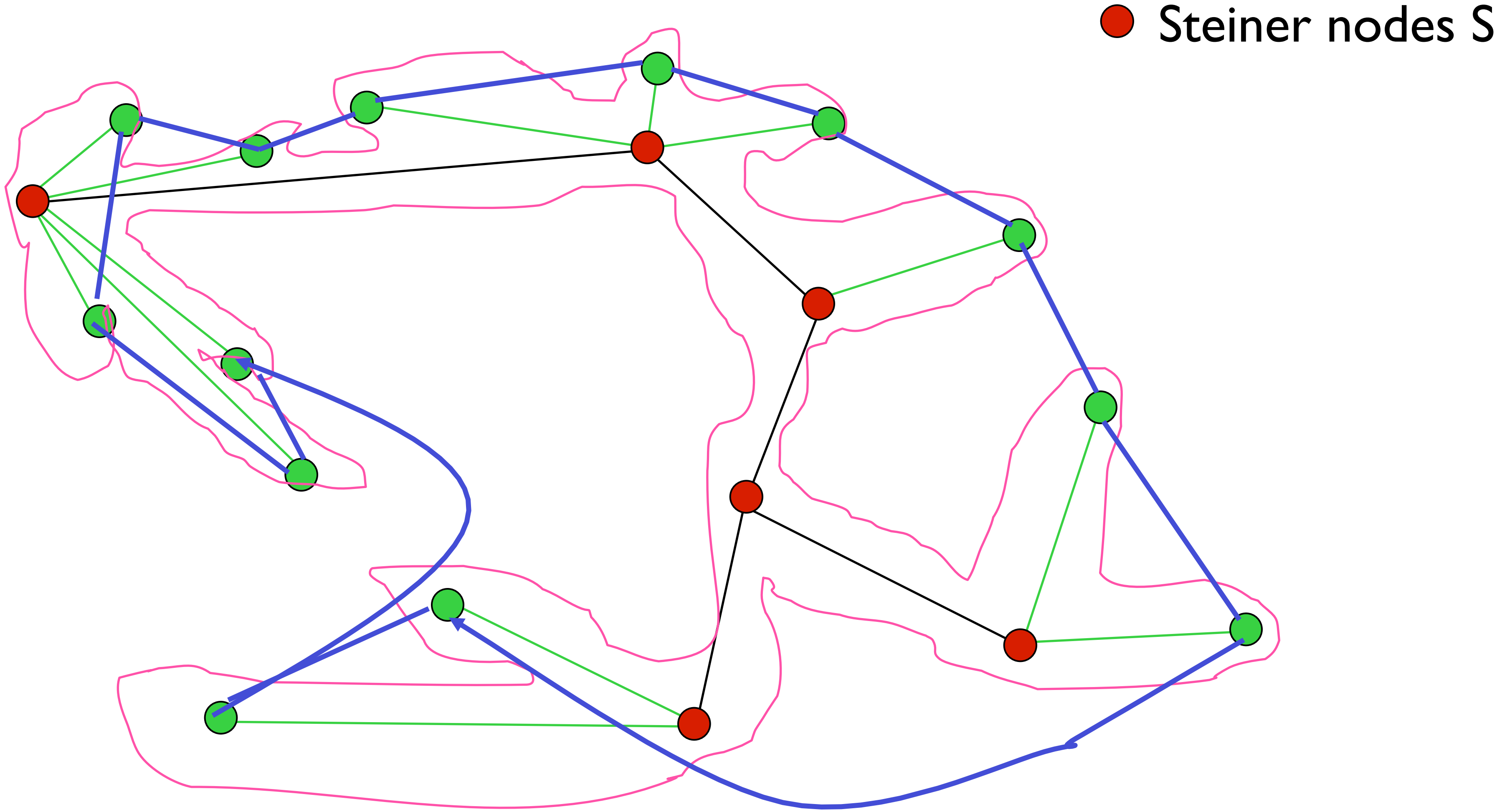


Deterministic Steiner Tree



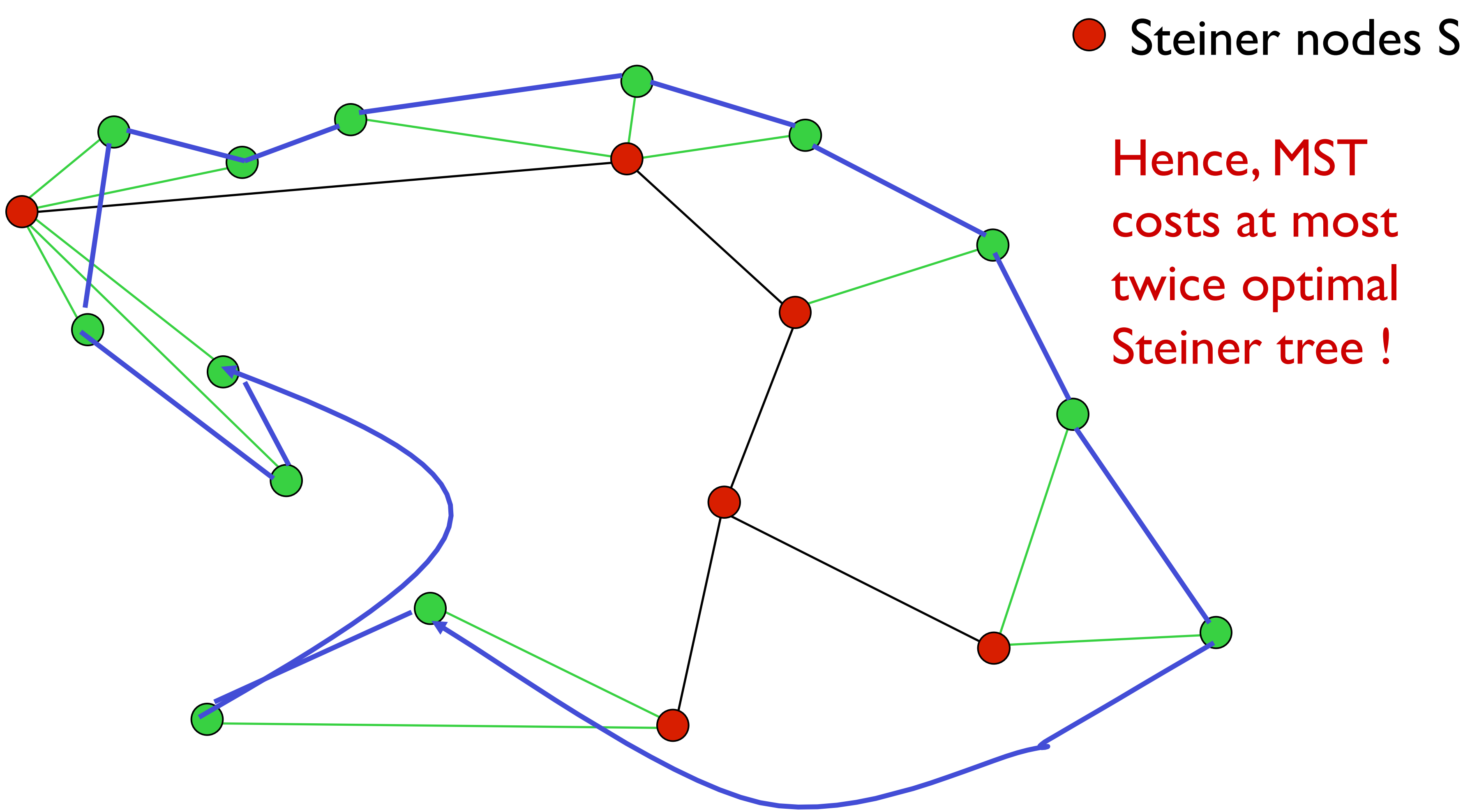
Walking "around" optimal Steiner tree gives connected graph, so can view as connected graph on just terminal nodes (by shortcuts)

Deterministic Steiner Tree



Walking "around" optimal Steiner tree gives connected graph, so can view as connected graph on just terminal nodes (by shortcuts)

Deterministic Steiner Tree



Hence, MST costs at most twice optimal Steiner tree !

Walking “around” optimal Steiner tree gives connected graph, so can view as connected graph on just terminal nodes (by shortcuts)

Boosted Sampling Algorithm
(Gupta, Pál, Ravi, Sinha)

- Draw λ independent samples $S_1, S_2, \ldots, S_\lambda \rightarrow S$
- First stage decision: compute minimum spanning tree (MST) for S (including root), and install those edges $\rightarrow$ Alg_I
- Observe scenario T (independently drawn from same dist)
- Compute (rooted) minimum spanning tree on $S \cup T$,
  (but make cost of edges Alg_I all 0)
  and let e[j] be edge from j to its parent
- Let $Alg_{II} \leftarrow \{ e[j] : j \in T \}$

Boosted Sampling Algorithm (Gupta, Pál, Ravi, Sinha)

- Draw λ independent samples $S_1, S_2, \ldots, S_\lambda \to S$
- First stage decision: compute minimum spanning tree (MST) for S (including root), and install those edges $\to \mathrm{Alg}_I$
- Observe scenario T (independently drawn from same dist)
- Compute (rooted) minimum spanning tree on $S \cup T$,
  (but make cost of edges Alg_I all 0)
  and let e[j] be edge from j to its parent
- Let $\mathrm{Alg}_{II} \leftarrow \{ e[j] : j \in T \}$

Theorem Boosted Sampling is a 4-Approximation Algorithm

First Stage Cost

Optimal cost $Z^* = c(Opt_I) + \lambda\, E_{T \subseteq N} [\, c(Opt_{II}(T))\,]$

We compute MST on $S \leftarrow S_1 \cup \ldots \cup S_\lambda$ for Stage I

(this is 2-approximation for S)

How expensive is it to connect S? Could use

$$Opt_I \cup Opt_{II}(S_1) \cup \ldots \cup Opt_{II}(S_\lambda)$$

Each S_i is identical random T so its expected cost is

$$c(Opt_I) + \lambda\, E_{T \subseteq N} [\, c(Opt_{II}(T))] \rightarrow Z^*$$

Since MST is 2-approximation algorithm $\Rightarrow$

expected Stage I cost is at most $2Z^*$

Cost sharing role of parental edge

- Build a MST on a set $S \cup T$ (plus root)
- Focus on parental edge $e[j]$ for each $j \in S \cup T$
- Total edge cost is $\sum_{j \in S \cup T} c_{e[j]}$
- But this is ≤ twice cost of optimal Steiner tree on $S \cup T$
- Attribute share $c_{e[j]}/2$ of optimal cost to j
- Total share cost is ≤ optimal Steiner tree cost

Second Stage Cost

- Algorithm computes Steiner tree for $S_I \cup \ldots \cup S_\lambda \cup T$
- Consider $T \leftarrow Opt_I \cup Opt_{II}(S_I) \cup \ldots \cup Opt_{II}(S_\lambda) \cup Opt_{II}(T)$
- Role of $\lambda+1$ sets, $S_I, \ldots, S_\lambda, T$ is symmetric
- $E[c(T)] \le c(Opt_I) + (\lambda+1)\, E[c(Opt_{II}(S_i))] \le (\lambda+1)/\lambda\ Z^*$
- Form $D_1, \ldots, D_\lambda$ by deleting nodes in multiple sets
- $\sum_{j \in T\text{-}S} c_{e[j]} + \sum_i \sum_{j \in D_i} c_{e[j]} \le 2c(T)$
- By symmetry, $E[\sum_{j \in T\text{-}S} c_{e[j]}] \le 2c(T)/(\lambda+1)$
- Hence, $E[\sum_{j \in T\text{-}S} c_{e[j]}] \le 2Z^* / \lambda \Rightarrow$ Stage II cost $\le 2Z^*$!

$\Rightarrow$ Boosted Sampling is 4-approximation algorithm

Discrete Stochastic Optimization and Approximation Algorithms

- Area of emerging importance
- Rich source of algorithmic questions
- Can one prove a strong result for approximate stochastic dynamic programming? [Levi Roundy & S] [Halman, Klabjan, Mostagir, Orlin & Simchi-Levi]
- When is sampling information good enough to derive near-optimal solutions?
- Reconsider some well-studied problems but now in "black box" model, not just specific distributions
- Expectation is not enough

Thank You.