Matching Based Augmentations for Approximating Connectivity Problems

R. Ravi
Carnegie Bosch Professor
Tepper School of Business
Carnegie Mellon University

Invited talk from LATIN 2006:
survey write-up in those proceedings

Outline

- Introduction and Early Examples
- Applications to Bicriteria Spanning Trees
- An application to online Steiner Trees
- Extensions

Context

- NP-hard connectivity problems
  - TSP, Steiner trees
  - Spanning trees with multiple objectives
  - Network design with price and distance
- Goal: Develop a technique for designing polynomial-time approximation algorithms
- Approximation ratio for minimization problem

$$\rho = \max_{\text{instances } I} \text{cost}(A(I)) \, / \, \text{cost}(OPT(I))$$

Matching Based Augmentation

- Iterative construction heuristic: Subgraph added at each iteration identified by examining the optimal solution (typically a matching variant)
- Each iteration's cost related to that of optimal. (Performance ratio is of the order of the number of iterations)

Warm-up: Christofides' Algorithm for Symmetric TSP

- Problem: Given a metric, find a tour of minimum total cost
- Algorithm [Christofides '76]:
  - Compute a MST T
  - Compute a minimum-cost matching M on the odd-degree nodes of T
  - $T \cup M$ is Eulerian and can be shortcut into a tour

Analysis of Christofides' Algorithm

- $c(\text{MST } T) \leq c(\text{OPT})$
  - Deleting an edge of OPT gives a spanning tree
- $c(M) \leq c(\text{OPT})/2$
  - Shortcutting the tour over the odd nodes, its cost does not increase (metric)
  - The shortcut tour can be decomposed into two matchings; pick the cheaper one
- $c(\text{output tour}) \leq 3c(\text{OPT})/2$

Features of Christofides'

- Two iterations
- Subgraph added at each iteration motivated by OPT, and the current state of the solution
- Each iteration charged to OPT separately

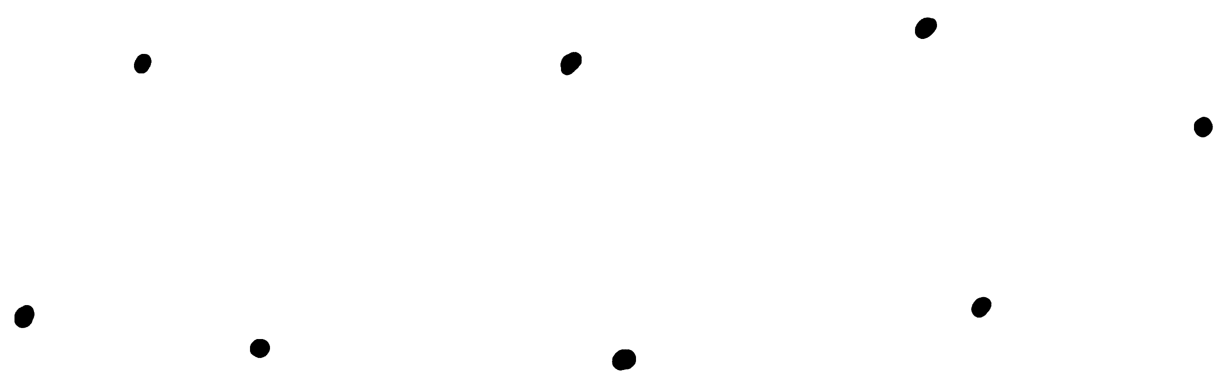
Warmup: FGM Algorithm for Asymmetric TSP

- Problem: Given an complete bidirected graph with asymmetric costs ($c_{ij} \neq c_{ji}$) but "metric" ($c_{ij} \leq c_{ik} + c_{kj}$), find a directed Hamiltonian tour of minimum cost
- Not clear how to approximate in two iterations

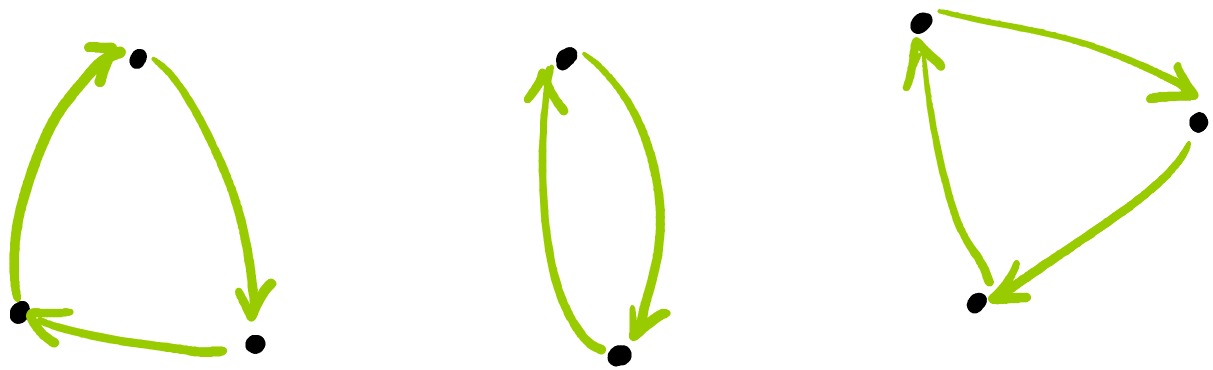
FGM Algorithm

- [Frieze, Galbiati, Maffioli '82] Algorithm:
  - Initialize all nodes as representatives
  - Iterate until only one rep remains
    - Find a minimum cost cycle cover on the representatives
    - Retain only one rep in each cycle in the cover
  - Resulting subgraph is connected and a union of cycles, hence Eulerian
  - Shortcut an Eulerian walk to obtain a tour

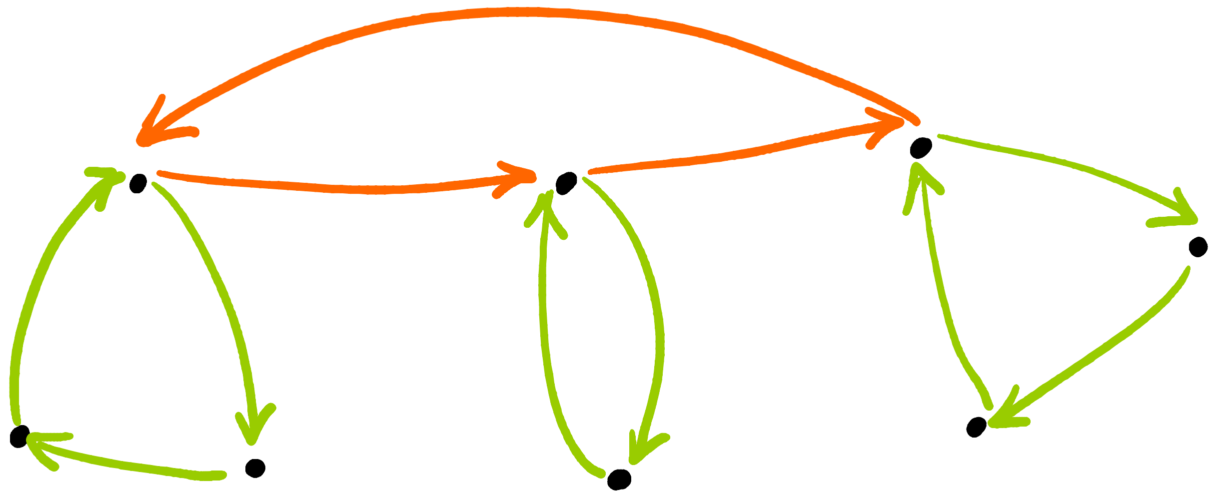
FGM Algorithm Example



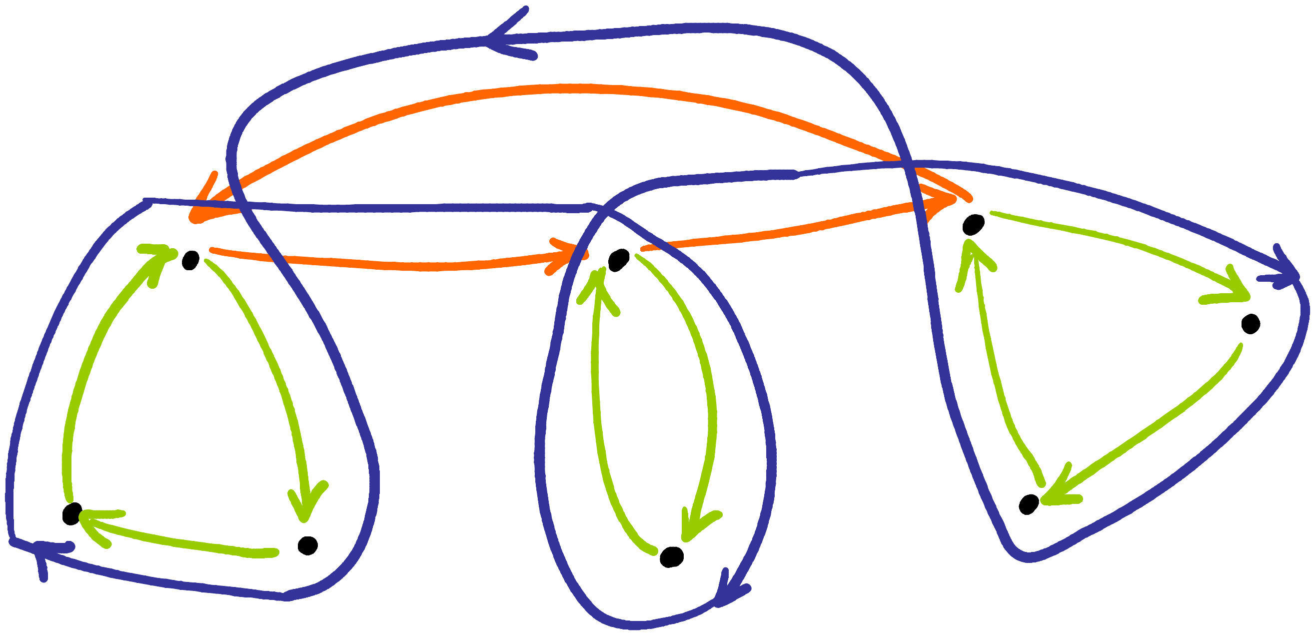
FGM Algorithm Example



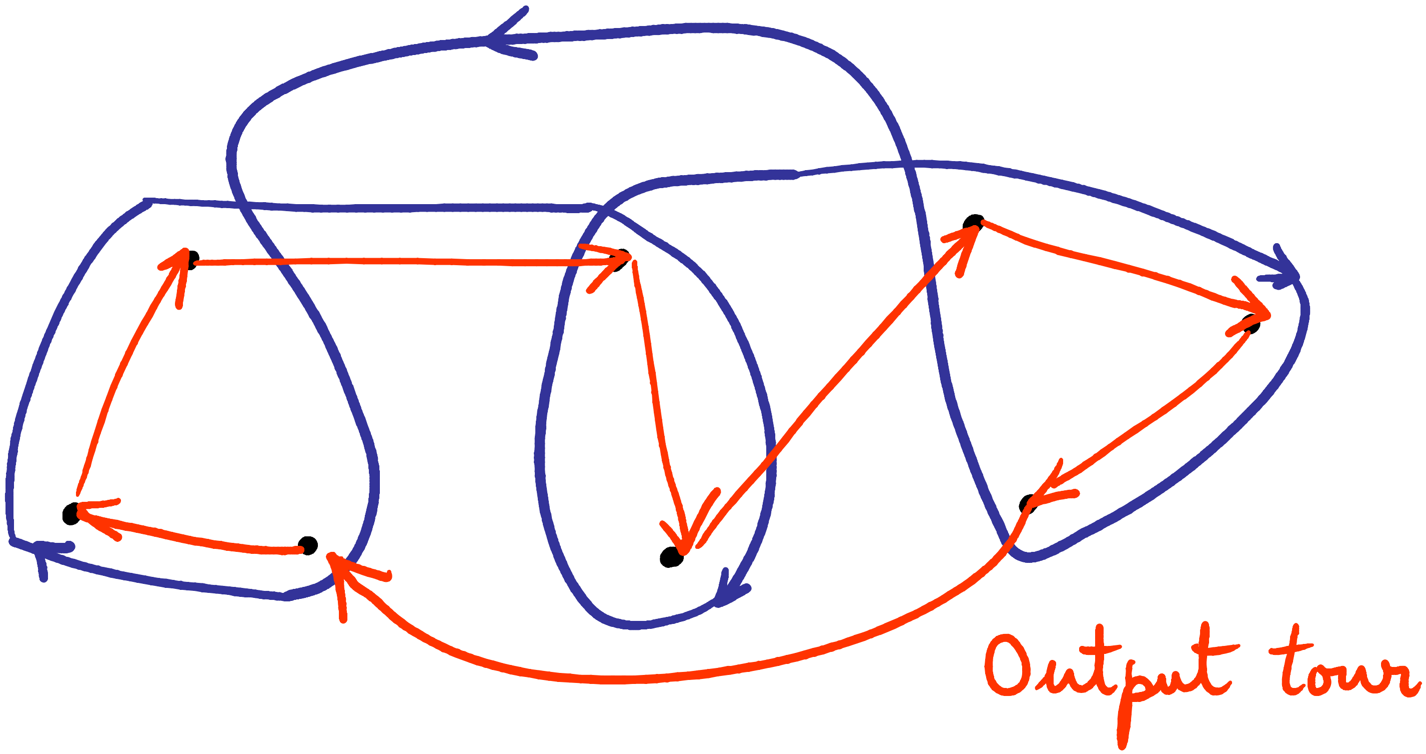
FGM Algorithm Example



FGM Algorithm Example



FGM Algorithm Example



FGM Algorithm Analysis

- Cost of cycle cover in each iteration is at most that of OPT
  - Shortcutting the optimal tour over all non-reps gives a single cycle covering all reps
- Number of iteration is $\log_2 |V|$
  - Every iteration reduces number of reps by a factor of 2 by merging each rep with at least one other rep
- Approximation ratio $\leq$ Number of iterations

Outline

- Introduction and Early Examples
- Applications to Bicriteria Spanning Trees
  - Degree-bounded min-cost trees
  - Tree Pairing Lemma
  - Diameter-bounded min-cost trees
- An application to online Steiner Trees
- Extensions

Bicriteria Spanning Trees

- Spanning tree problem on undirected graphs with two minimization objectives
  - Total Cost of the edges under costs *c*
  - Maximum degree of any node
  - Diameter of the tree under lengths *l*
- Common formulation of many networking problems
  - Small congestion/delay broadcast trees
  - Subroutine for minimum broadcast schedules

Budgeted Formulations

Min Obj1(T) s.t. Obj2(T) $\leq$ B

- Degree-bounded minimum-cost trees

  Min c(T) s.t. max-degree of any node $\leq$ D

- Diameter-bounded minimum-cost trees

  Min c(T) s.t. diameter under $l \leq L$

- Diameter-bounded min-degree trees

  Min dia_l(T) s.t. max-degree of any node $\leq$ D

All these problems are NP-hard

Bicriteria Approximations

- For “Min Obj1(T) s.t. Obj2(T) $\leq$ B”, an (α, β)- approximation returns a tree T’ with
  - Obj1(T’) $\leq \alpha$ Obj1(T*)
  - Obj2(T’) $\leq \beta$ B

  where T* is the optimal solution to the budgeted problem

Matching Based Augmentation

- Adapt iterative idea for bicriteria spanning tree problems
  - Iteratively add subgraph to construct final solution
  - Bound the value of *each* of the two objectives per iteration w.r.t. the optimal by carefully choosing subproblem

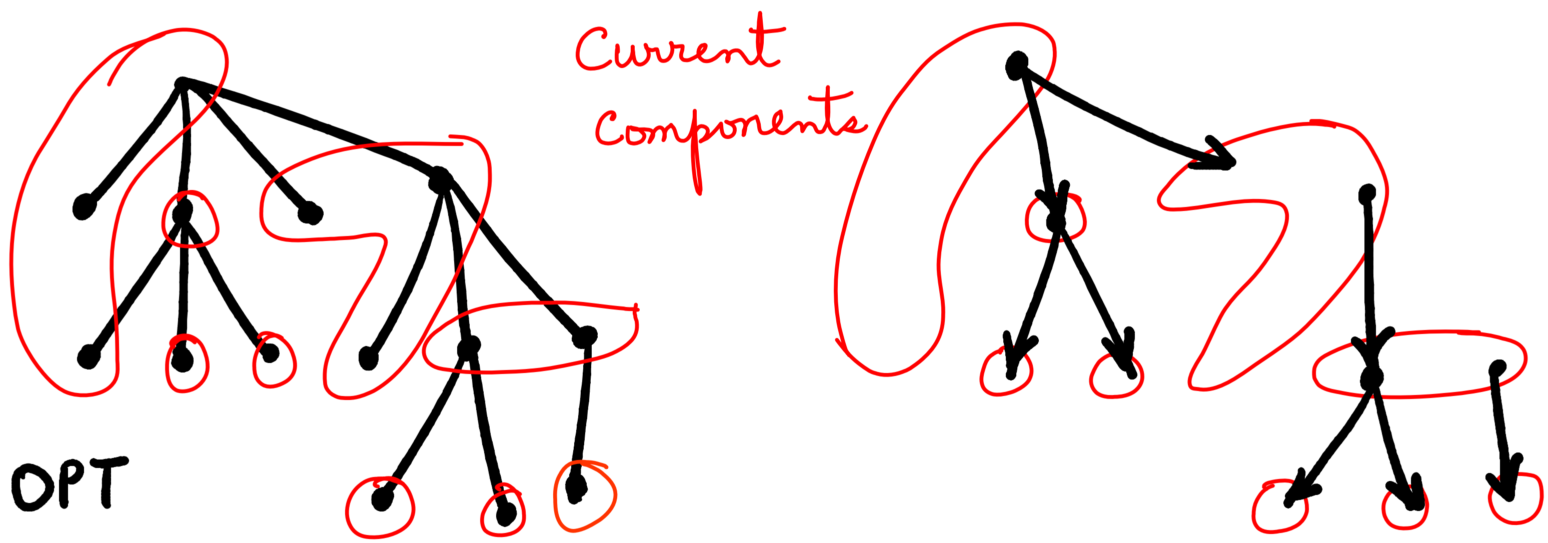
Degree-bounded minimum cost spanning tree

Min c(T) s.t. max-degree of any node $\leq$ D

- Iteratively add subgraphs such that
  - Degree of any node $\leq$ O(D) and total cost of the subgraph $\leq$ O(c*)
  - Number of iterations can be well bounded
- To infer the subgraph problem, consider any partial solution and ask how the optimal solution can be used to make progress in connectivity

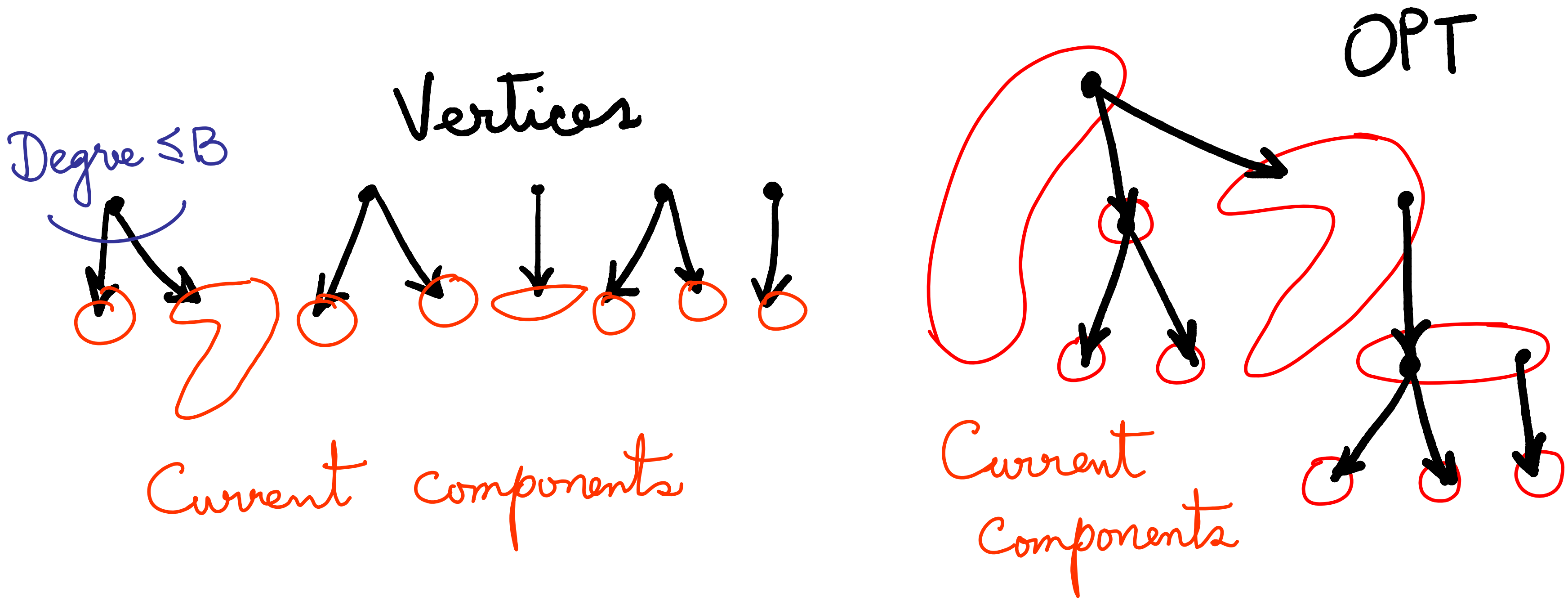
A subproblem based on OPT

- Consider OPT and the current solution



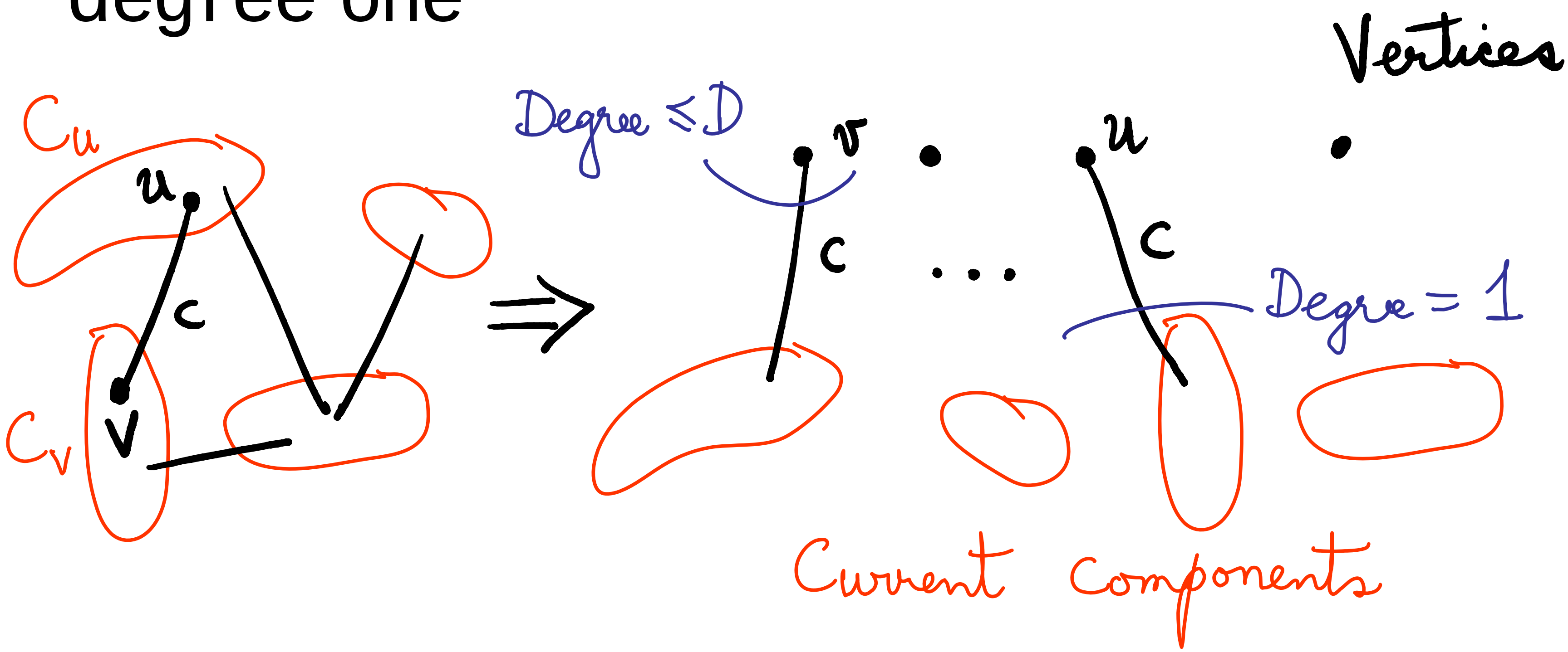
A subproblem based on OPT

- Consider OPT and the current solution to identify the connecting subproblem (Cf. Furer-Raghavachari '90)



A Minimum Cost D-Matching Problem

- Real nodes have degree bound D
- Current connected components need degree one



Algorithm

- Start with empty subgraph.
- Iterate until connected
  - Set up a bipartite min-cost D-matching problem on current components, solve and add to the solution
- Choose any spanning tree of the final subgraph

Analysis

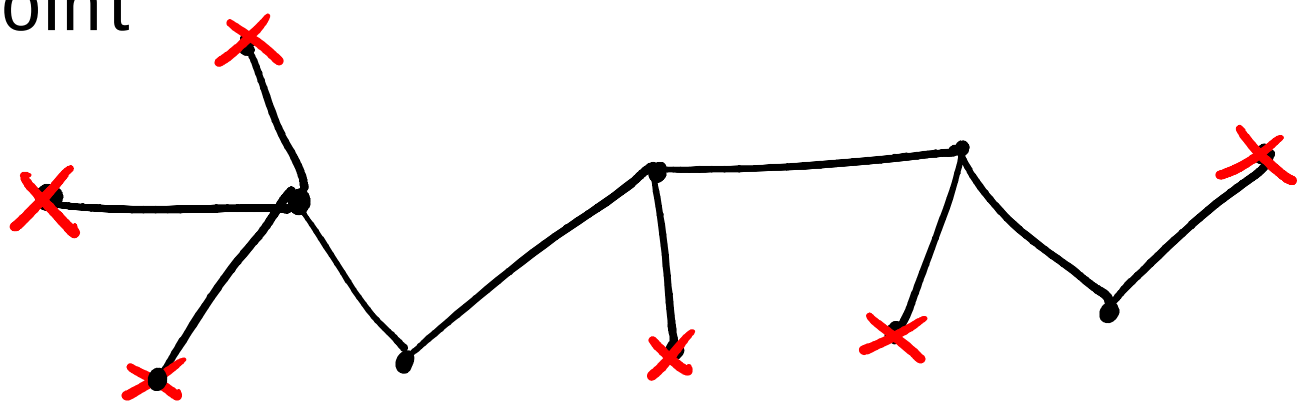
- Number of iterations is $\log_2 |V|$
  - Every connected component is joined with at least one other in every iteration
- Degree added to any node in an iteration is at most D + 1.
  - D from the real node side and 1 from the component side of the matching
- Cost of the matching added in an iteration is at most c*
- Final solution is an (O(log n), O(log n))-approximation

Outline

- Introduction and Early Examples
- Applications to Bicriteria Spanning Trees
  - Degree-bounded min-cost trees
  - Tree Pairing Lemma
  - Diameter-bounded min-cost trees
- An application to online Steiner Trees
- Extensions

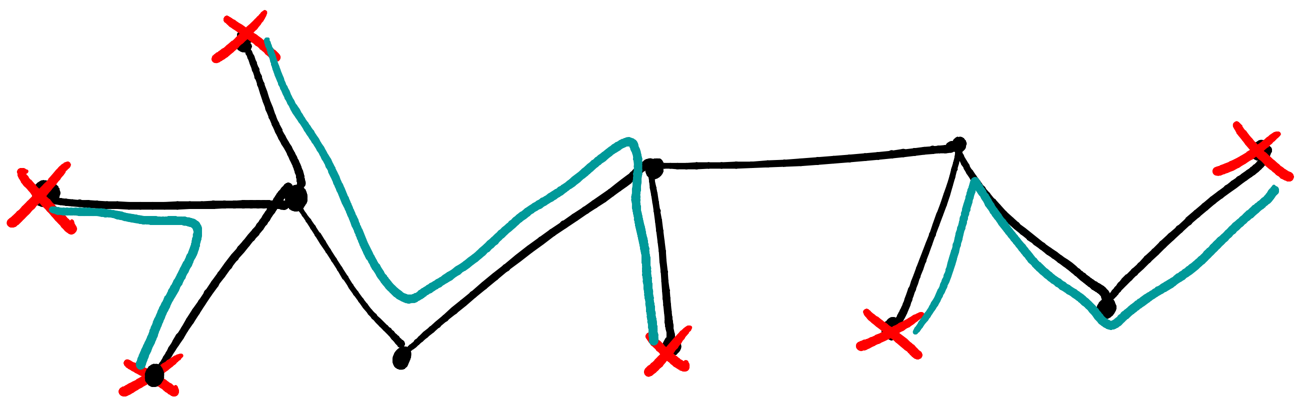
A Tree-Pairing Lemma

- Given an even number of nodes of a tree, there is a pairing of these nodes such that the tree-paths between the pairs are edge-disjoint



A Tree-Pairing Lemma

- Given an even number of nodes of a tree, there is a pairing of these nodes such that the tree-paths between the pairs are edge-disjoint



  - Pairing minimizing the total path length has this property

Proposed Application

- Identify representatives in each connected component of the current solution
- Use lemma to pair them up in a hypothetical optimal solution
- Infer the resulting matching problem that needs to be solved to augment the current solution (to halve the number of components)

Outline

- Introduction and Early Examples
- Applications to Bicriteria Spanning Trees
  - Degree-bounded min-cost trees
  - Tree Pairing Lemma
  - Diameter-bounded min-cost trees
- An application to online Steiner Trees
- Extensions

Diameter-bounded minimum cost spanning tree

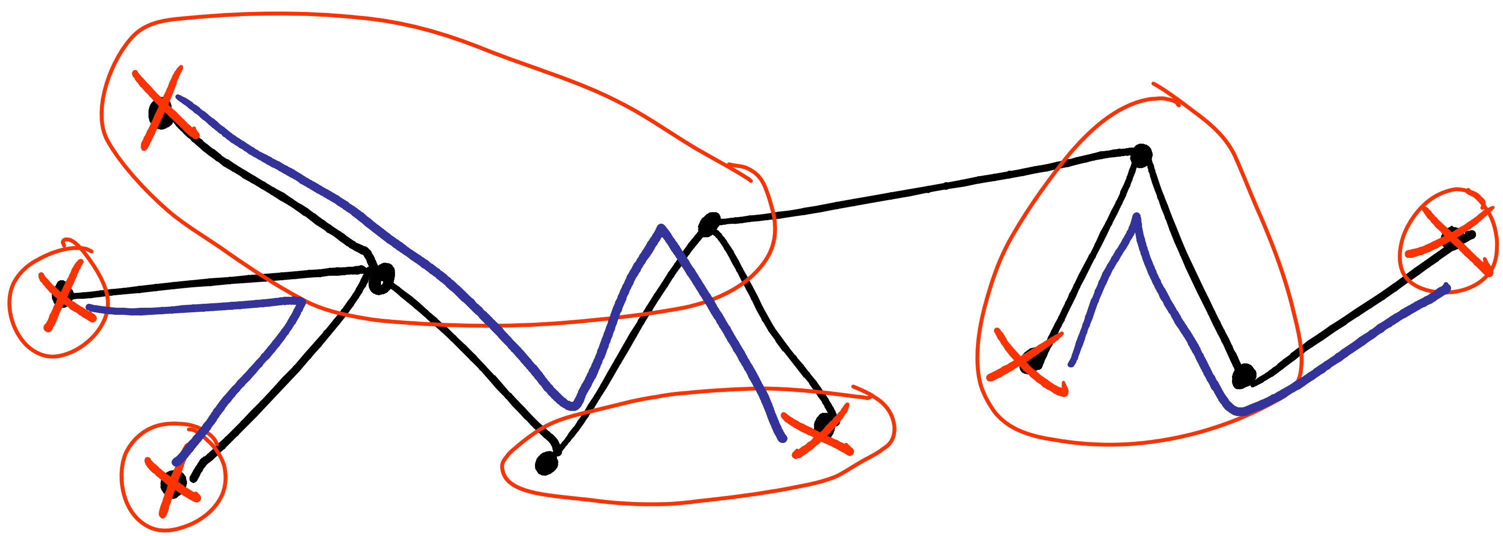
Given costs *c* and lengths *l* on the edges,

Min c(T) s.t. $\text{diameter}_l(T) \leq L$

- Iteratively add subgraphs such that
  - Diameter of added subgraph $\leq O(D)$ and total cost of the subgraph $\leq O(c^*)$
  - Number of iterations can be well bounded
- To infer the subgraph problem, consider any partial solution and ask how the optimal solution can be used to make progress in connectivity

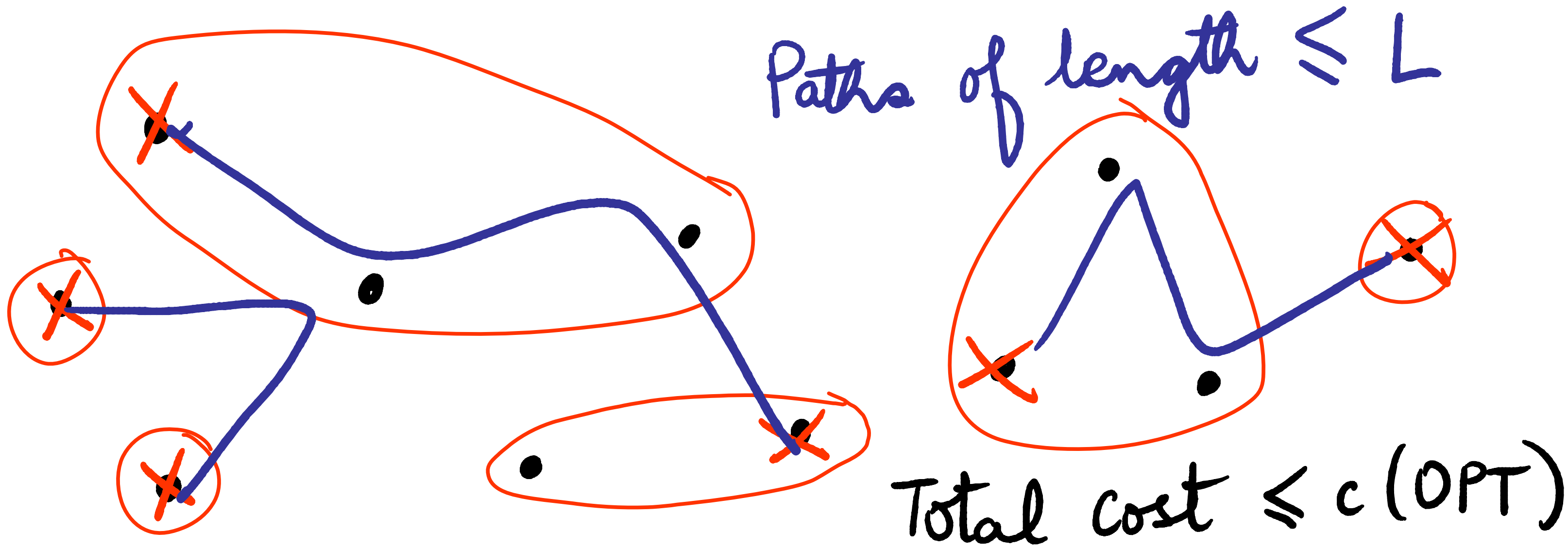
Apply tree pairing lemma to OPT

- Consider OPT and one rep from each component in the current solution



Apply tree pairing lemma to OPT

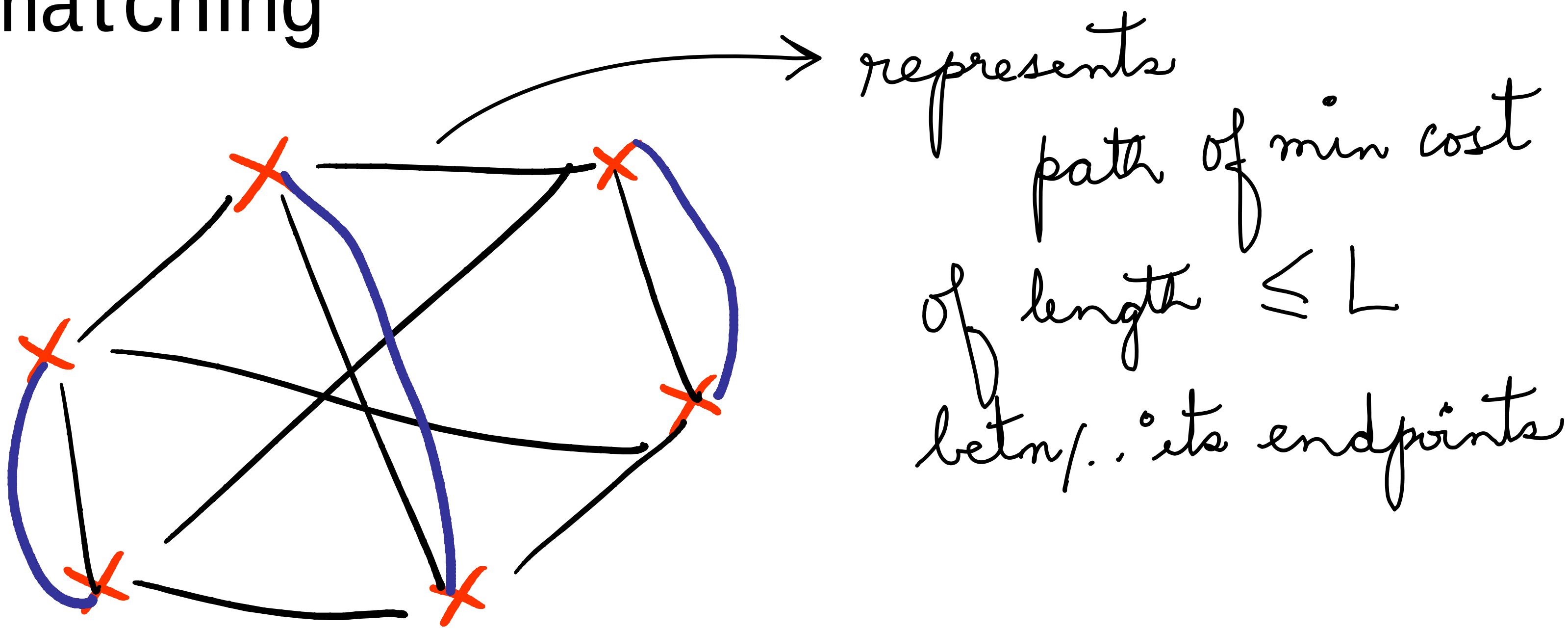
- Consider OPT and one rep from each component in the current solution



to infer the augmenting subproblem

Minimum Cost Length-bounded Matching Subproblem

- Match the reps using paths of length at most L, minimizing the total cost of matching



Detail: Length-bounded Matchings

- To enforce every matched pair has a length-L bounded path, solve the length-L bounded min cost path problem for every pair of reps
  - For a given pair, find a minimum-cost of a length-L bounded path between the pair
  - Also NP-hard but can get a PTAS, i.e., a length-L path of cost at most $(1 + \varepsilon)$ times the minimum for any fixed $\varepsilon > 0$
- Compute such costs for every pair of reps and solve min-cost perfect matching problem in this complete graph on reps

Algorithm

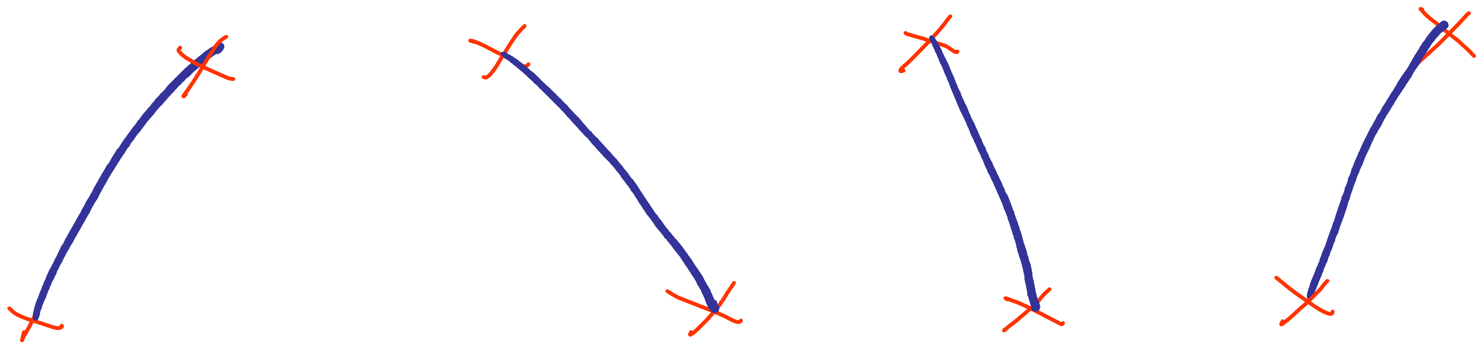
- Start with empty subgraph, all nodes are reps
- Iterate until connected
  - Set up a length-L bounded min-cost matching problem on current reps, solve and add to the solution
- Choose any spanning tree of the final subgraph?

Additional Complication

- Diameter is not additive like degree

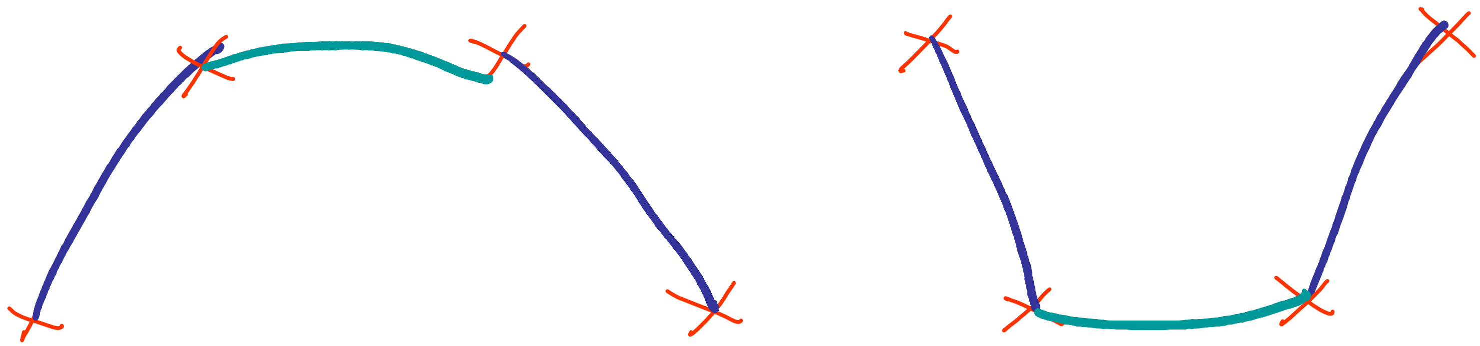
Additional Complication

- Diameter is not additive like degree



Additional Complication

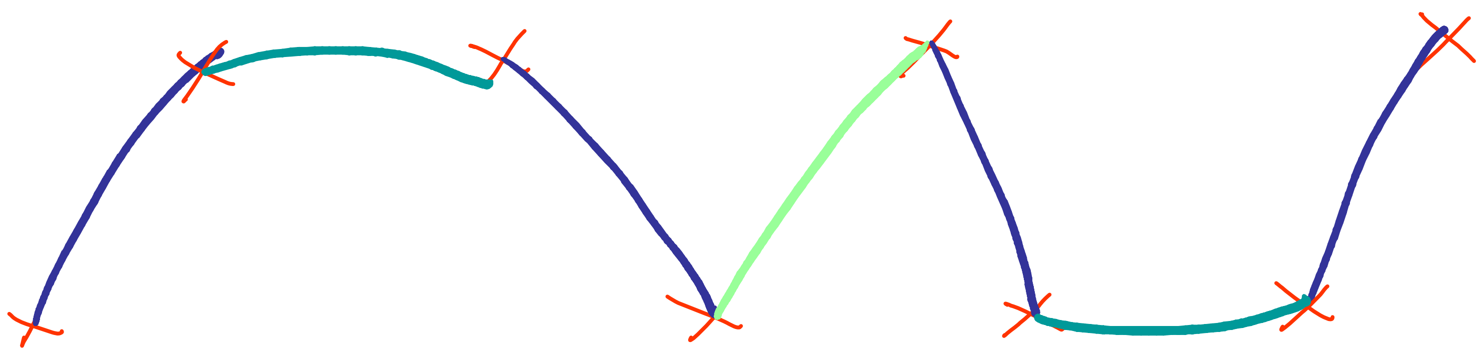
- Diameter is not additive like degree



... and can grow despite each subgraph being bounded in diameter

Additional Complication

- Diameter is not additive like degree



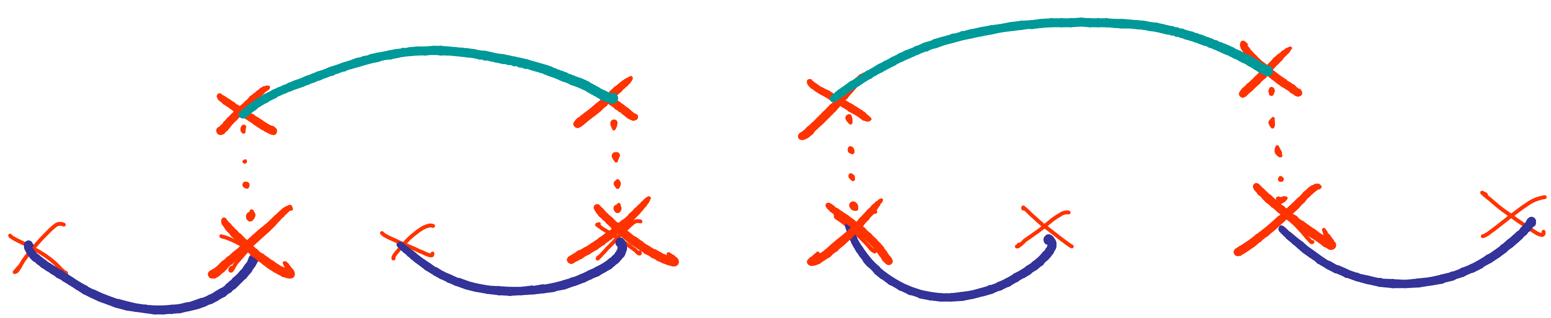
… and can grow despite each subgraph being bounded in diameter

Simple fix

- Promote one rep from each pair to bound diameter by number of iterations

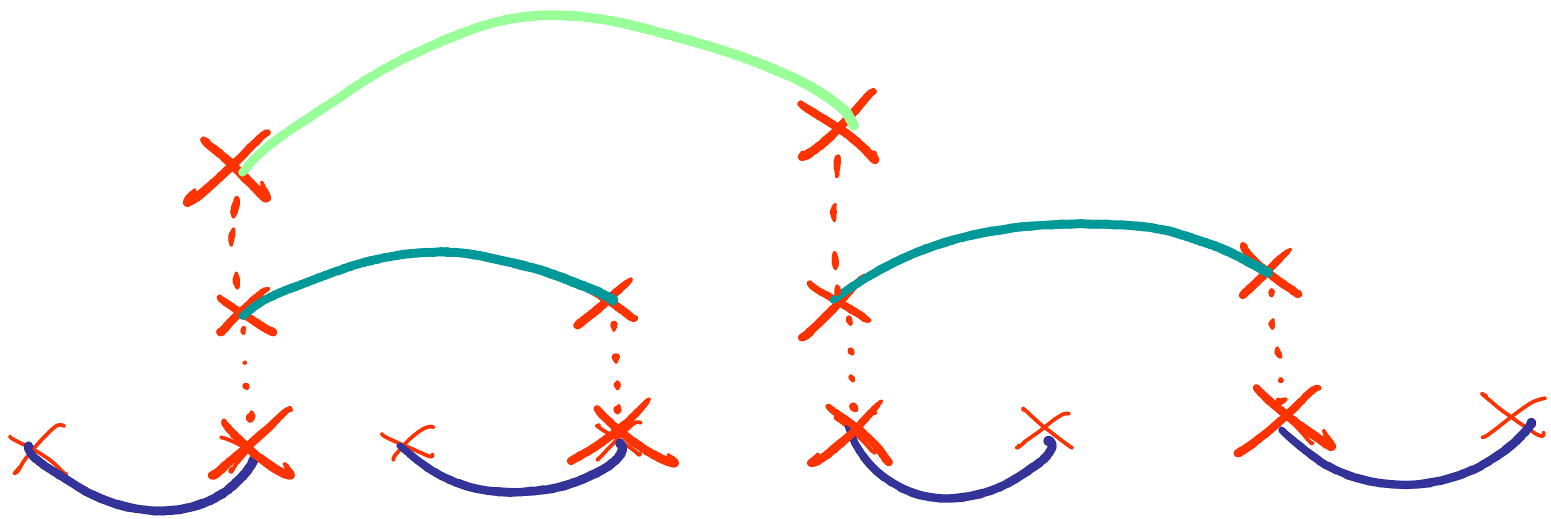
Simple fix

- Promote one rep from each pair to bound diameter by number of iterations



Simple fix

- Promote one rep from each pair to bound diameter by number of iterations



Algorithm

- Start with empty subgraph, all nodes are reps
- Iterate until connected
  - Set up a length-L bounded min-cost matching problem on current reps, solve and add to the solution
  - Retain only one rep from each matched pair
- Choose a minimum-diameter tree under *l* of the final subgraph

Analysis

- Number of iterations is $\log_2 |V|$
  - Every rep is joined with at least one other in every iteration
- Any initial rep in an iteration-i component, has a path of length at most iL to the current rep in the current solution.
  - Induction
- Cost of the matching added in an iteration is at most $(1 + \varepsilon)$ c*
  - Tree pairing lemma
- Final solution is an (O(log n), O(log n))-approximation

Diameter-bounded min-degree trees

Min $dia_l(T)$ s.t. max-degree of any node $\leq$ D

- Arises as a subroutine in finding a minimum broadcast schedule under the telephone model
- MBA-technique using the tree pairing lemma leads to a minimum node-congestion bounded-length path matching problem solved using randomized rounding of an LP
- Rep promoting fix is useful to ensure bounded diameter growth

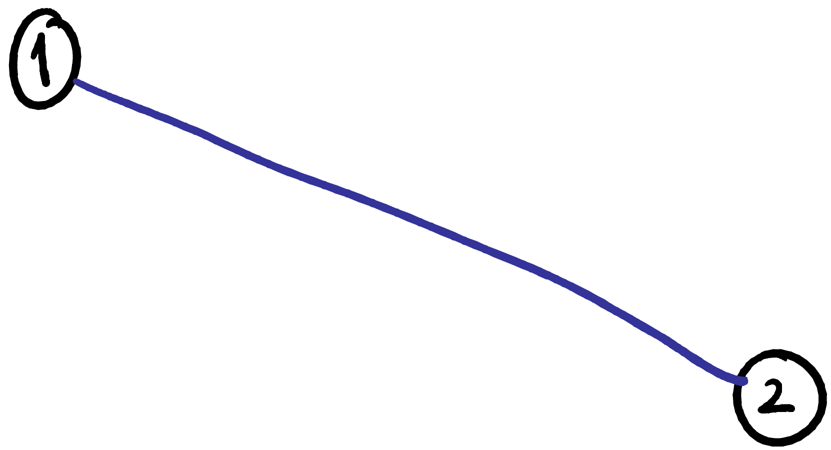
Outline

- Introduction and Early Examples
- Applications to Bicriteria Spanning Trees
- An application to online Steiner Trees
- Extensions

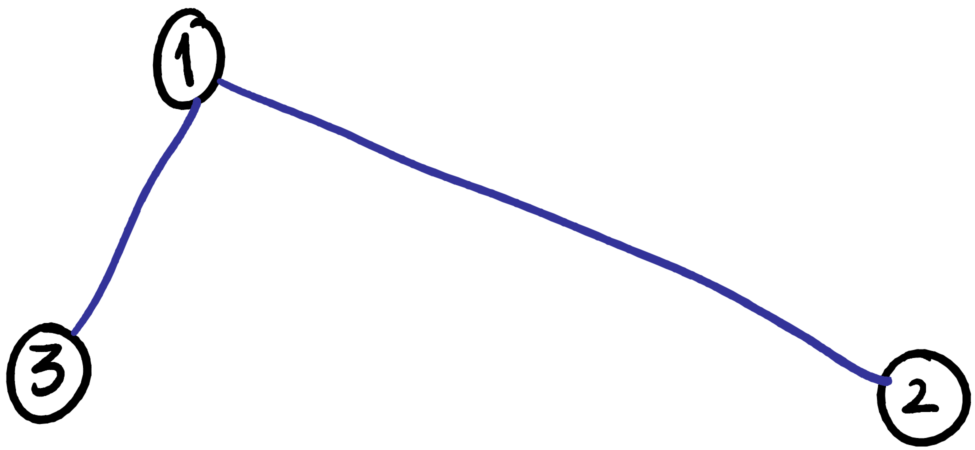
Application to Online Steiner Trees

- Given a set of terminals that arrive online, connect terminals as they arrive
- Minimize the maximum ratio of the cost of current Steiner tree to that of the minimum for this set of terminals (competitive ratio)
- Greedy Algorithm:
  - Connect next terminal via shortest path to the current tree
- Can use tree pairing lemma to show log k competitive ratio

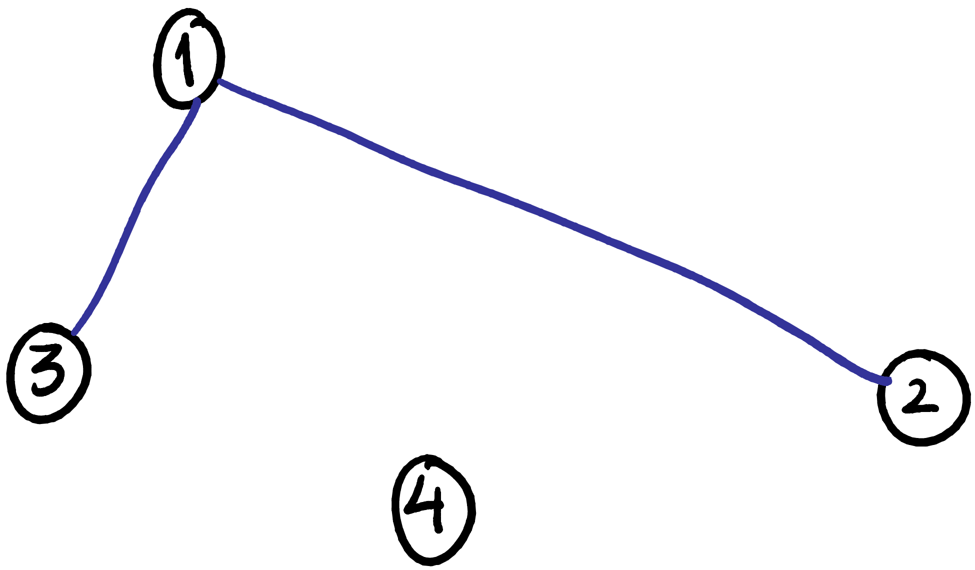
Greedy Example



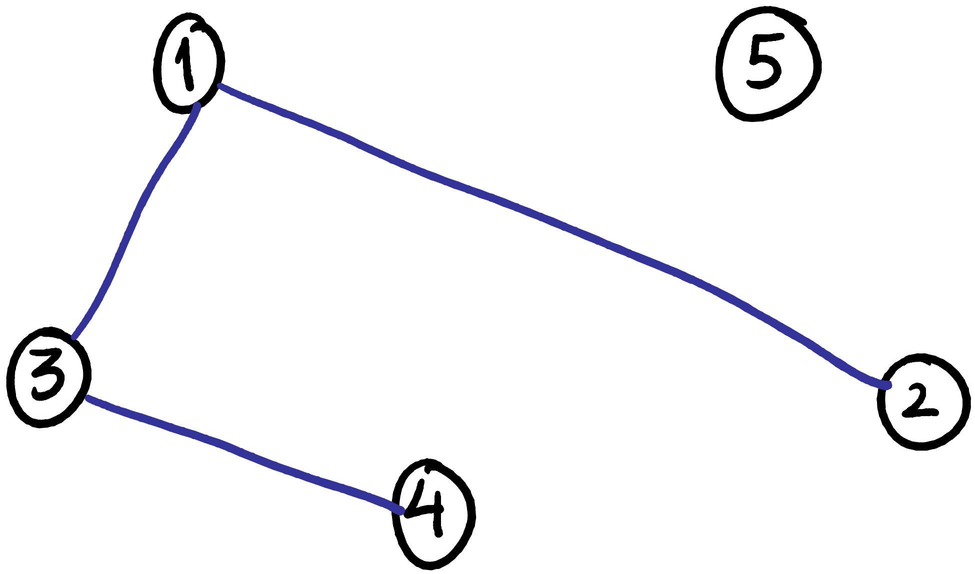
Greedy Example



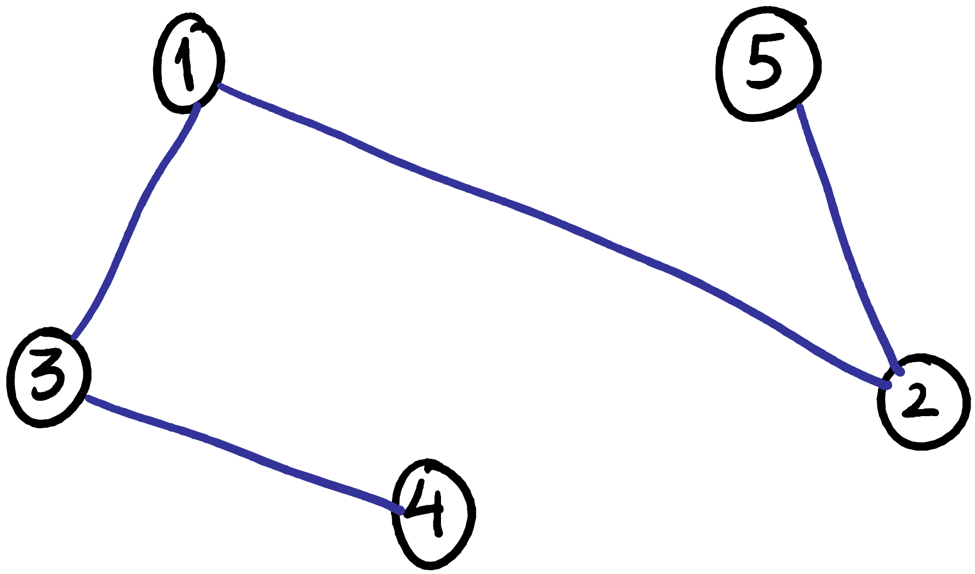
Greedy Example



Greedy Example



Greedy Example

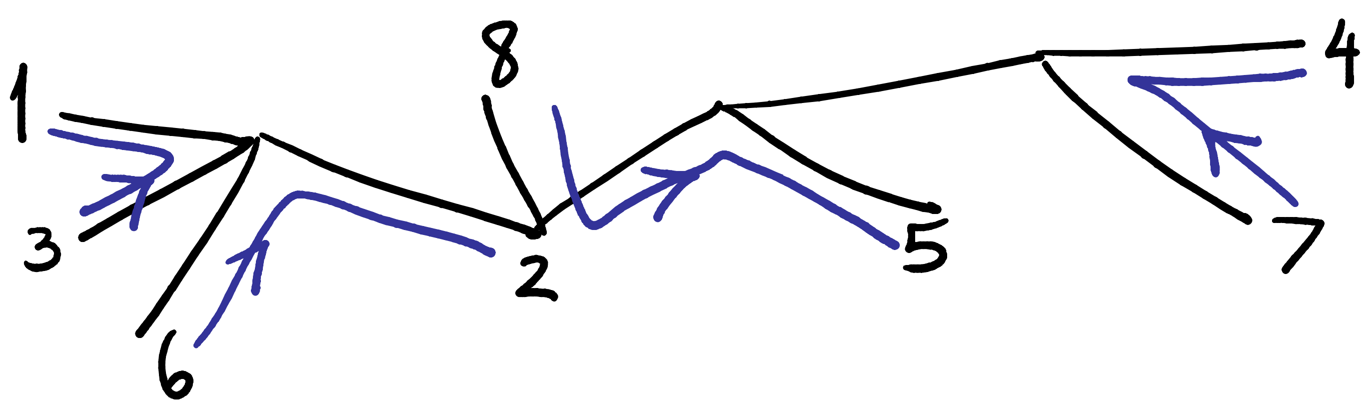


Analysis of Greedy

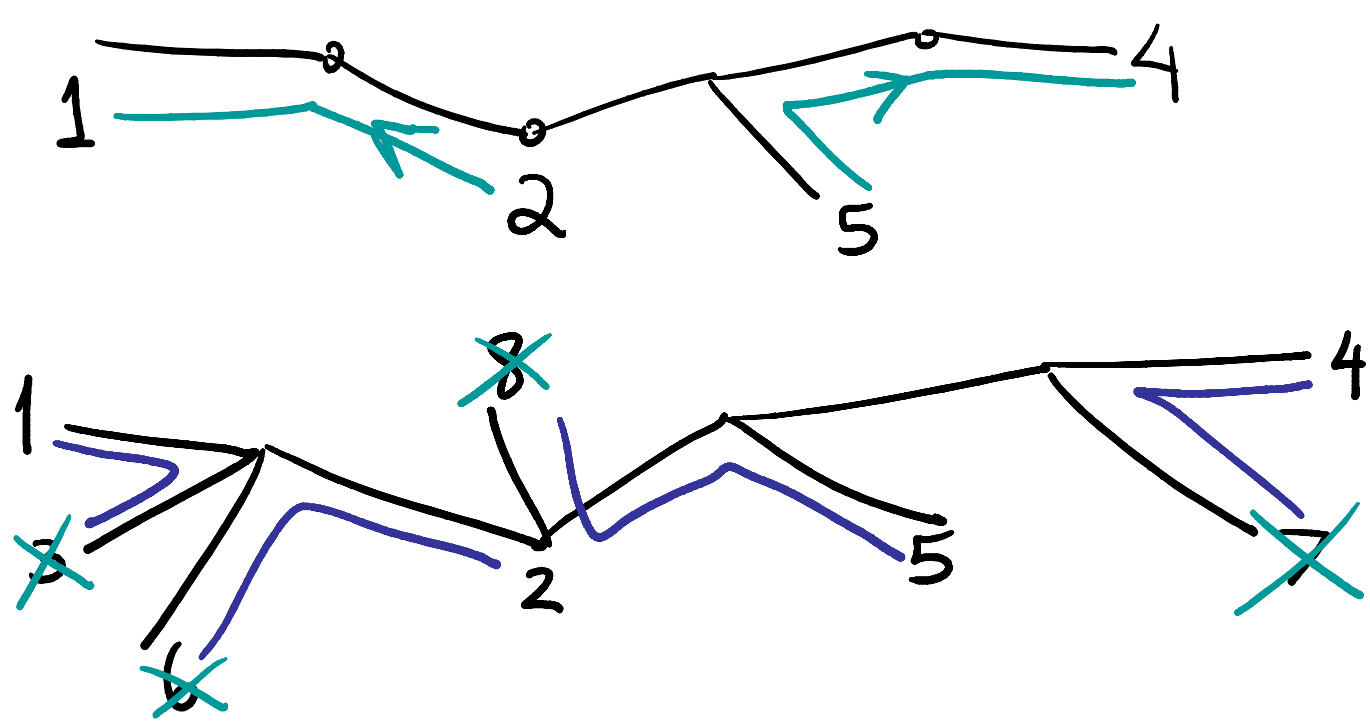
- Terminals 1,...,k added to form current greedy tree G.
- Cost of greedy tree $\leq \sum^{k}_{i=2}$ c(path from i to closest terminal among 1 though i-1)

  $\leq \sum^{k}_{i=2}$ c(path from i to any earlier terminal)
- Charging Idea: Use tree pairing lemma in the optimal tree to find a pairing of every terminal to one that arrived before it

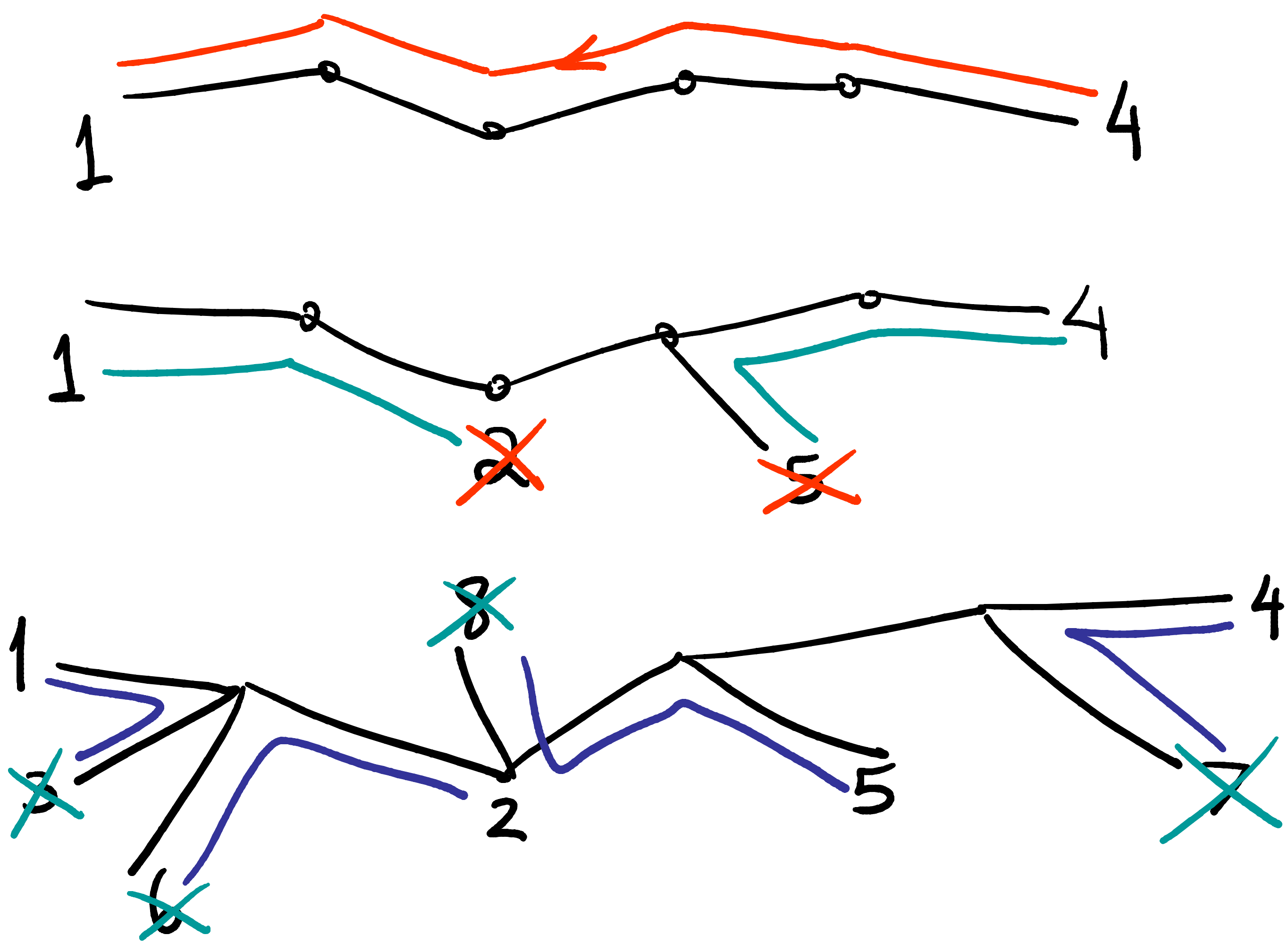
Tree pairing applied to OPT



Tree pairing applied to OPT



Tree pairing applied to OPT



Pairing every terminal to one before

- Start will all k terminals, and pair them up using paths in an optimal tree
- Assign path for the pair to the terminal that arrived later (to reach the earlier one) and delete assigned terminals
- Repeat on remaining terminals
- All terminals assigned in $\log_2 k$ iterations
- Assignments suffice to bound greedy tree

Outline

- Introduction and Early Examples
- Applications to Bicriteria Spanning Trees
- An application to online Steiner Trees
- Extensions

Extension: Cost-Distance Network Design

- Find a spanning tree T rooted at given *r* minimizing $c(T) + \sum_v \text{distance}_l(r,v)$
- Algorithm (assuming $|V| = 2^k$) [Meyerson, Munagala, Plotkin, 2000]
  - Initialize all nodes as reps
  - Iterate for i = 1 to k
    - Find a min-weight matching on the reps under the weights $w = c + 2^{i-1}\, l$ (Cf. weight of edges to root = c + l)
    - *Randomly* choose a rep from each pair to retain

Sketch of Analysis

- Inductively maintain that the problem on the retained reps (with demand of its component aggregated on it) has expected cost-distance at most optimal
  - Randomization is crucial here
- Expected value of cost-distance accrued in each iteration is at most optimal
- Performance ratio is bounded by number of iterations, i.e., logarithmic in number of nodes

Extension: Simultaneous Optimization for Concave Costs

- Given a root *r*, costs c on edges, and a nondecreasing concave function f, find tree T routing one unit of flow from every vertex to *r* minimizing $\sum_e c_e f(flow_e)$
- Goal: Find one tree that is simultaneously near-optimal for *all* concave f's
- [Goel-Estrin '03] Randomized tree construction guaranteeing

$$E[\max_f c(T)/c(T^*_f)] \leq \log n + 1$$

Hierarchical Matching Algorithm

- Algorithm (assuming $|V| = 2^k + 1$, and c is metric) [Goel Estrin '03]
  - Initialize all non-root nodes as reps
  - Iterate for i = 1 to k
    - Find a min-weight matching on the reps under the costs c
    - *Randomly* choose a rep from each pair to retain
  - Connect last rep to root

Sketch of Analysis

- Use randomization to argue that expected cost of aggregated instance is bounded by optimal
- Bound expected flow cost on every matching by the expected cost of aggregated instance
- Use demand basis functions (powers-of-two) and concavity of f to argue about all f's simultaneously

Summary

- MBA: An iterative construction heuristic based on matching-variant subproblem
- Performance ratio proportional to number of iterations

Open Problems

- Any relation to set-cover type greedy algorithms or analysis that also result in logarithmic guarantees?
- MBA solution for diameter-L bounded min cost generalized Steiner tree problem?