Approximate k-MSTs and k-Steiner Trees

via

the Primal-Dual Method and Lagrangean Relaxation

Tim Roughgarden

Cornell University

and IBM Almaden

joint work with

Fabián A. Chudak (Tellabs)

David P. Williamson (IBM Almaden)

The k-MST Problem

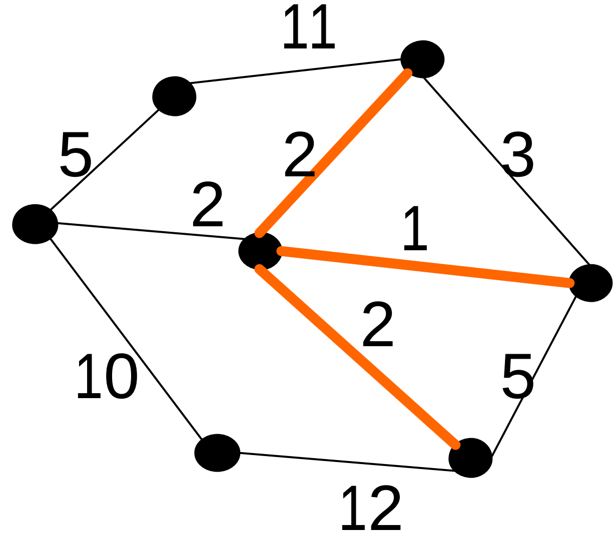
Given :

- An undirected graph $G = (V,E)$
- costs $c_e \geq 0$ on edges
- a parameter k

Goal :

- min-cost tree spanning $\geq k$ vertices

Example $(k = 4)$:



Some History

Fact: k-MST is NP-hard

Approximation Algorithms:

- $O(k^{1/2})$ [Ravi et al. 94]
- $O(\log^2 k)$ [Awerbuch et al. 95]
- $O(\log k)$ [Rajagopalan/Vazirani 95]
- 17 [Blum/Ravi/Vempala 95]
- 5 [Garg 96]
- 3 [Garg 96]
- 2 + ? [Arora/Karakostos 00]
- 2 [Garg 00]

Motivating Question

Observation: all constant-factor approx algs for k-MST rely on a primal-dual alg for prize-collecting Steiner tree

Prize-collecting Steiner tree:

- given: graph $G=(V,E)$, costs $c_e \geq 0$ on E, penalties ≥ 0 on V
- goal: tree minimizing cost of its edges + penalties of unspanned vertices

Question: why?

The Connection

Punchline: a PCST problem arises as a Lagrangean relaxation of the k-MST problem

Roughly:

- complicating constraint = tree spans ≥ k vertices

- lift to objective function
  - use parameter ? ≥ 0 to penalize trees spanning < k vertices

    Þ get a PCST problem with all vertex penalties equal to ?

The Agenda

Our goal: with this insight, revisit existing approx algs for k-MST and derive

- a simpler algorithm description
- a simpler proof of approx ratio

Þ we will focus on Garg's 5-approximation algorithm

Our Inspiration

Jain/Vazirani '99 gave:

- a primal-dual 3-approximation algorithm for uncapacitated facility location
- a reduction from k-median to facility location, via Lagrangean relaxation
  - nasty constraint = open = k medians
  - facility cost = penalty parameter ?

Þ yields a 6-approximation algorithm for k-median

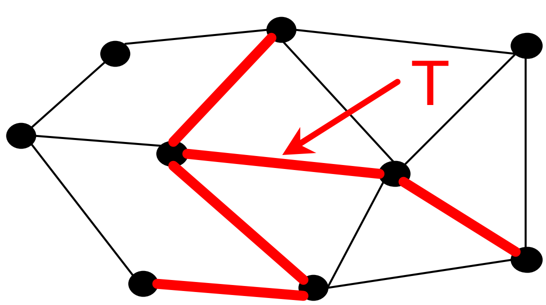
Sketch of k-MST Formulation

minimize:

cost of edges in tree T = c(T)

subject to:

vertices spanned by T are connected



also:

T fails to span = n-k vertices

Lagrangean Relaxation of k-MST

Choose: value for Lagrangean penalty parameter ? ≥ 0

Lagrangean relaxation k-MST(?):

minimize:

\# of unspanned vertices

$$c(T) + ? \, [|V \setminus T| - (n-k)]$$

subject to:

vertices spanned by T are connected

Fact: ? ≥ 0 Þ lower bound for k-MST

Prize-Collecting Steiner Tree

Suppose: penalty on every vertex is set to ? $\geq$ 0.

minimize:

c(T) + ? |V\T|

# of unspanned vertices

subject to:

vertices spanned by T are connected

same as before

Þ only difference from k-MST(?) is missing -?(n-k) in obj fn

A Primal-Dual Algorithm for PCST

Good news: we know how to approximate PCST well.

- Theorem [Goemans, Williamson 95]: In poly-time, can construct a tree T + feasible (fractional) dual s.t.

$$\underbrace{\text{primal cost}}_{c(T)\,+\,\text{penalty of } V\setminus T} = 2 \times \underbrace{\text{dual cost}}_{\text{from dual of PCST LP relaxation}}$$

- Stronger [GW 95]:

$c(T) + 2 \times$ penalty of $V\setminus T = 2 \times$ dual cost

From PCST to k-MST

$64K Question: What does the PCST result imply for k-MST?

- interpret k-MST(?) as a PCST instance with all penalties = ? ≥ 0

Problems:

- different primal objective fns
  - k-MST: c(T)
  - PCST: c(T) + ? |V\T|
- different duals
  - recall: k-MST(?), PCST differ only by a ?(n-k) term in obj fns

  Þ duals identical except for this ?(n-k) term (in obj fns)

An Inherited Guarantee

- running GW algorithm on k-MST(?) yields tree T, dual s.t.:

c(T) + 2? |V\T| = 2 × PCST dual cost

- if |V\T| = n-k (T spans k vertices)

  Þ subtracting 2?(n-k) on each side:

c(T) = 2 × [PCST dual cost - ?(n-k)]

= 2 × [k-MST(?) dual cost]

= 2 × OPT

Þ done if we can find magic value for ?

The Catch

Problem: what if no value of ? yields a tree T spanning exactly k vertices?

Solution (a la [Jain/Vazirani 99])

- ? = 0 Þ T will be empty
- ? suff. large Þ T spans all vertices
- via bisection search, can find:
  - $?_1 < ?_2$ and $?_1 \approx ?_2$
  - $(?_1, T_1, y_1)$, T_1 spans $k_1 < k$ vertices
  - $(?_2, T_2, y_2)$, T_2 spans $k_2 > k$ vertices
- will combine T_1, T_2 and y_1, y_2 to get feasible + near-optimal solution

Combining Two Guarantees

So far: we have guarantees for $i=1,2$:

$$c(T_i) + 2?_i(n-k_i) = 2[\text{PCST dual cost of } y_i]$$

Idea: take a convex combination of the two so previous calculation works.

Þ choose μ_1, μ_2 s.t.

$$\mu_1 (n-k_1) + \mu_2 (n-k_2) = n-k$$

Þ assume $? = ?_1 = ?_2$, get:

$$\mu_1 c(T_1) + \mu_2 c(T_2) + 2?(n-k)$$

$$= 2[\text{PCST dual cost of } \underbrace{\mu_1 y_1 + \mu_2 y_2}_{\text{feasible dual}}]$$

Result: $\mu_1 c(T_1) + \mu_2 c(T_2) = 2 \times OPT$

An Easy Case

We have:

$$\mu_1 c(T_1) + \mu_2 c(T_2) = 2OPT$$

T_2 (the big tree) is a feasible solution

Þ if $\mu_2 \geq ½$, just return T_2 for a 4-approximation

Þ can assume $\mu_1 \geq ½$, $\mu_2 = ½$

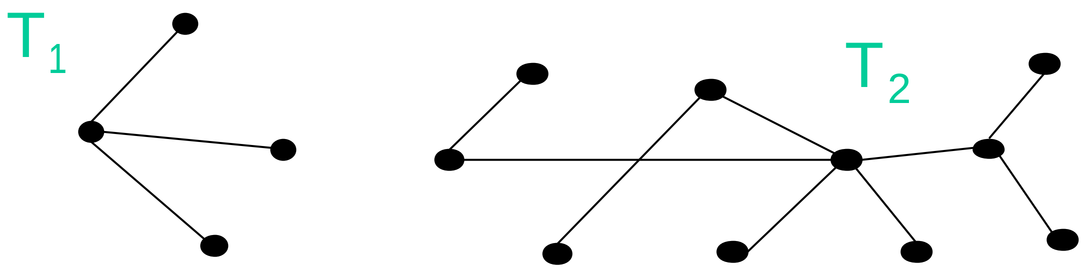
In harder case: supplement the small tree (T_1) with a few vertices from T_2

Supplemental Vertices

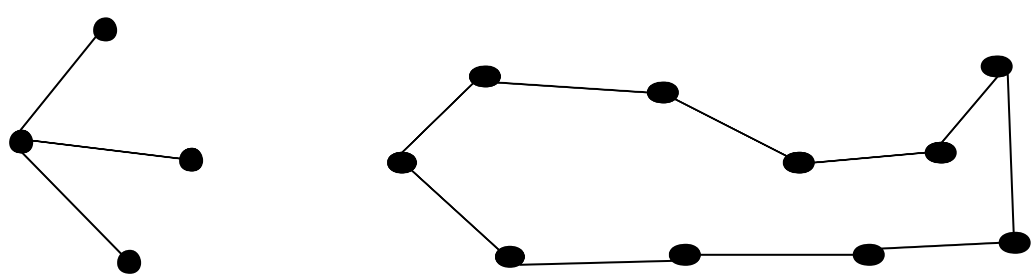
Algorithm [Garg 96]

$k = 7$

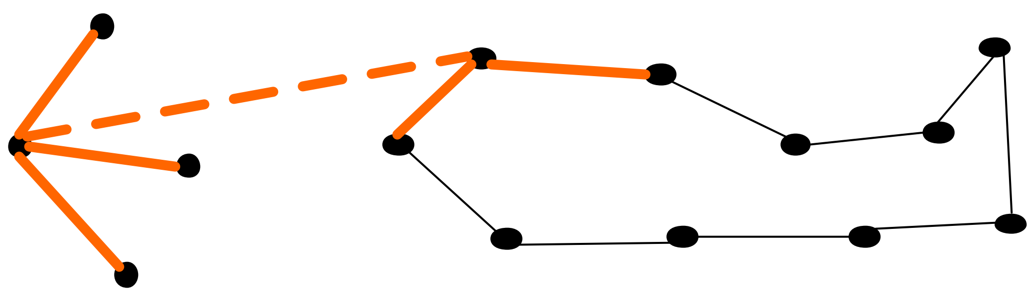
$k_1 = 4,\ k_2 = 10$



double edges of T_2 + shortcut to tour:



connect T_1 to cheapest path of tour:



k-Steiner Trees

The k-Steiner tree problem:

- given a graph with costs on edges and a distinguished set of terminal vertices
- find the min-cost tree spanning at least k terminals

Result:

- can extend previous algorithm + analysis, get a 5-approximation
  - use vertex penalties only for terminals

Open Questions

Directions for future work:

- further applications of Jain/Vazirani's techniques
  - Garg's 3-approximation algorithm
  - other NP-hard problems
- extend framework to handle many complicating constraints