Parametric and Kinetic Minimum Spanning Trees

Pankaj K. Agarwal

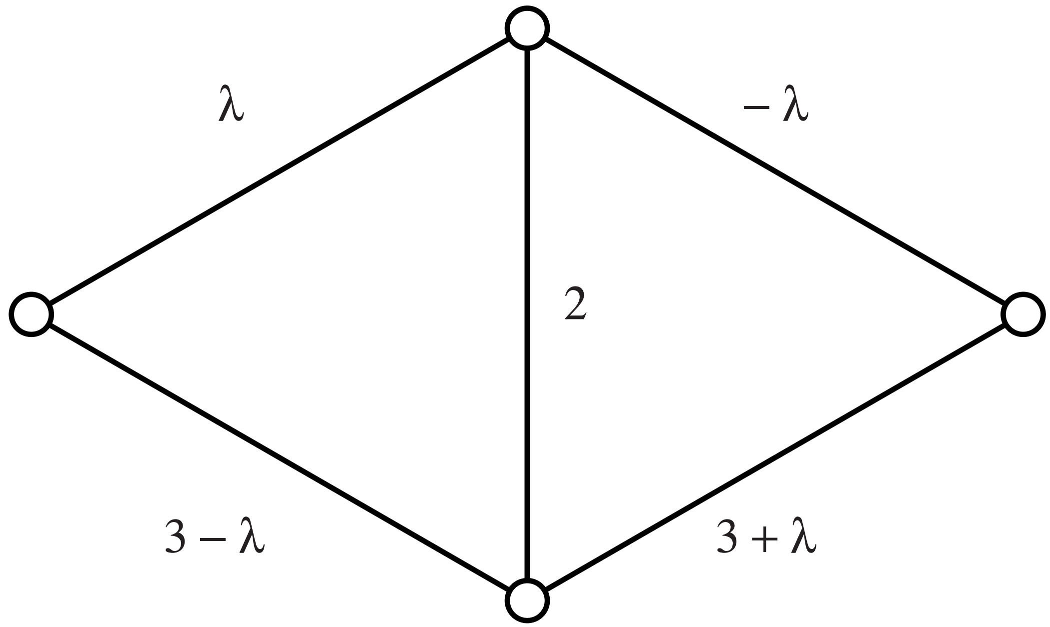
David Eppstein

Leonidas J. Guibas

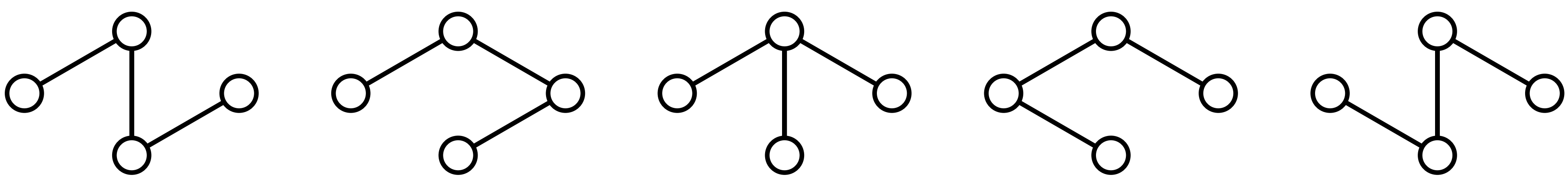
Monika R. Henzinger

Parametric Minimum Spanning Tree:

Given graph, edges labeled by linear functions



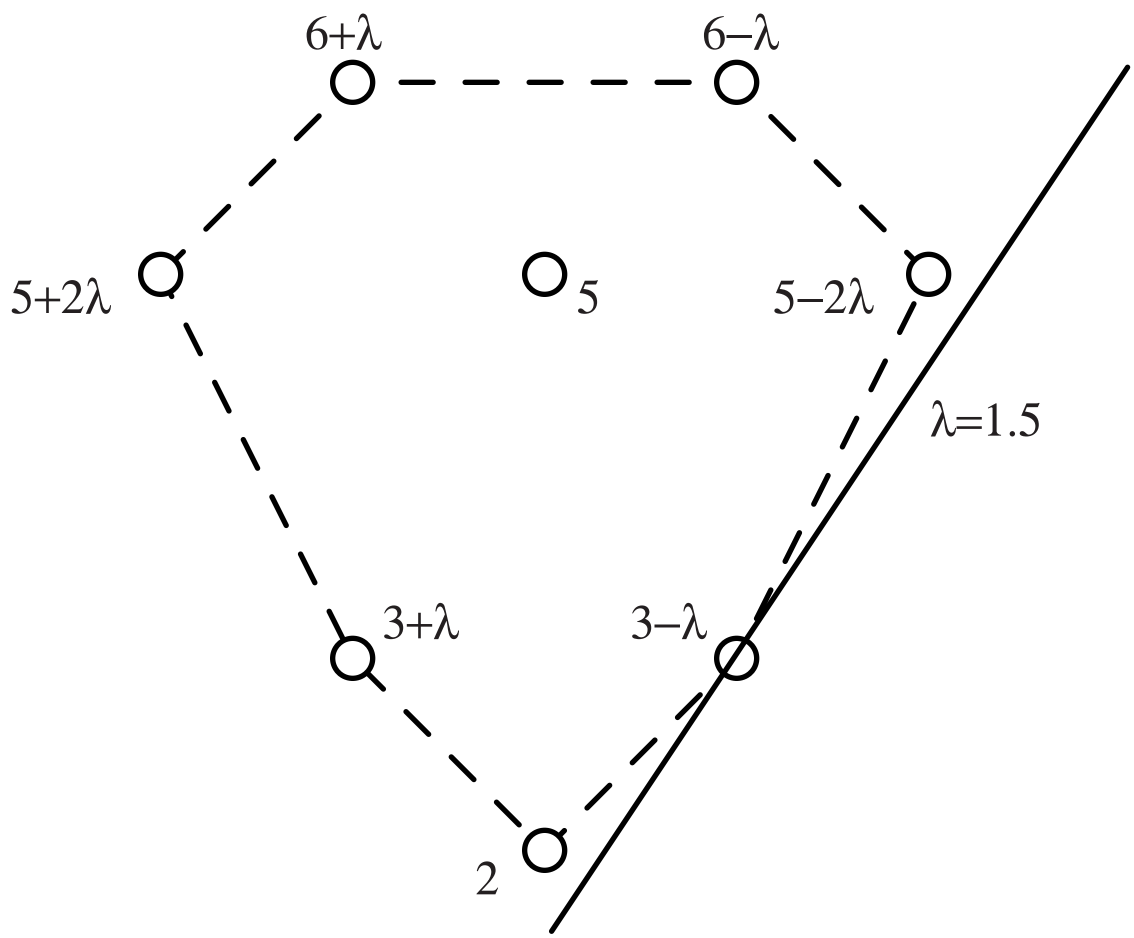
Find MST for each possible value of λ



$\lambda < -2$	$-2 < \lambda < -1$	$-1 < \lambda < 1$	$1 < \lambda < 2$	$2 < \lambda$
$5 + 2\lambda$	$3 + \lambda$	2	$3 - \lambda$	$5 - 2\lambda$

Geometric Interpretation:

Point (-B,A) for tree w/weight A + Bλ



Then MST(λ) = tangent to line w/slope λ

so parametric MST = lower convex hull

Applications

For any quasiconcave function $f(A, B)$
optimum tree must be a convex hull vertex

Tree w/ minimum cost-reliability ratio
($A = \text{cost}$, $B = -\log$ probability all edges exist):

$$f(A, B) = A \exp(B)$$

Tree w/ minimum variance in total weight
(if edge weights independent random variables):

$$f(A, B) = A - B^2$$

Tree with high probability of low total weight
(if edge weights independent Gaussian variables):

$$f(A, B) = A + \sqrt{B}$$

So each of these optima can be found
from parametric MST solution

Previous Results on Parametric MST

Number of breakpoints:

- $O(mn^{1/3})$ [Dey 1997]

- $\Omega(m\alpha(n))$ [Eppstein 1995]

Time to compute all trees:

- $O(mn \log n)$
  [Fernández-Baca, Slutzki, Eppstein 1996]

Dynamic Minimum Spanning Tree

An alternate form of time-varying data:
Weighted graph subject to discrete updates
(like parametric w/piecewise constant functions)

Many algorithms known

[Sleator, Tarjan 1983]
[Frederickson 1985]
[Eppstein 1991]
[Eppstein, Galil, Italiano, Nissenzweig 1992]
[Eppstein, Galil, Italiano, Spencer 1993]
[Henzinger, King 1997]
[Holm, de Lichtenberg, Thorup 1998]

Current best time: $O(\log^4 n)$ per update
better for restricted updates or planar graphs

Idea: apply dynamic graph algorithm techniques
to parametric MST problem

How to combine parametric and dynamic? Kinetic Algorithms!

Interpret λ as time parameter
start with parametric problem, small λ
increase λ and perform updates
maintaining correct MST at each point in process

Idea: model short-term predictability
and long-term unpredictability
of real-world applications

Two kinds of updates possible:

structural:
edge insertions and deletions

functional:
relabel existing edge w/new function

Other Kinetic Algorithms

[Basch, Guibas, Hershberger 1997]
[Basch, Guibas, Zhang 1997]
[Guibas 1998]
[Agarwal, Erickson, Guibas 1998]
[Basch, Erickson, Guibas, Hershberger, Zhang 1999]

Basic data structures
(Priority queue)

Computational geometry
(Convex hull, closest pair, binary space partition, polygon intersection)

Typical time bounds are polylog $\times$ worst case number of changes to solution

New Results

General graphs:

- $O(m^{2/3} \log^{4/3} m)$ per output change
- $O(n^{2/3} \log^{4/3} n)$ times worst case # changes

Minor-closed graph families
(including planar graphs):

- $O(n^{1/2} \log^{3/2} n)$ per output change

Minor-closed families with only functional updates:

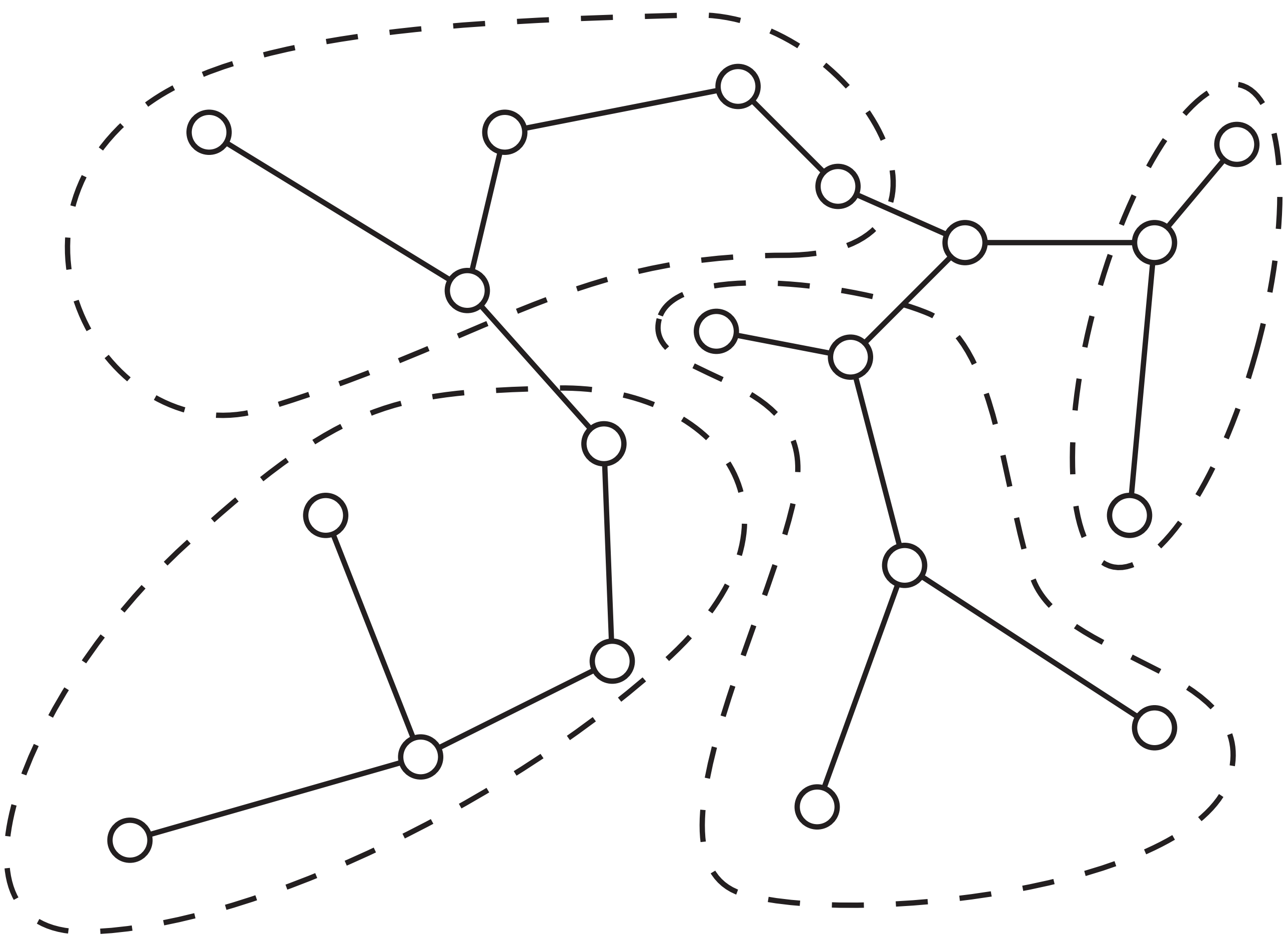
- $O(n^{3/2})$ preprocessing (nonplanar graphs only)
- $O(n^{1/4} \log^{3/2} n)$ times worst case # changes

Some randomized improvements to polylogs

Idea I: Clustering

Expand vertices so graph has degree three, then...

Group MST into k clusters of $O(n/k)$ edges,
at most two edges crossing each cluster boundary
[Frederickson 1985]



Form bundles of non-tree edges, according to the clusters containing their endpoints

adjust clusters as MST changes

Classification of MST changes

MST always changes by swap:
insert non-tree edge, delete tree edge

Three types of swap:

- Intra-cluster swap: tree edge belongs to cluster containing both non-tree edge endpoints

- Dual-cluster swap: tree edge belongs to cluster containing both non-tree edge endpoints

- Inter-cluster swap: tree edge and non-tree edge are in disjoint clusters

Finding Intra-Cluster Swaps

Use Megiddo's parametric search
to find last value of λ
for which the cluster has same MST

Decision oracle is (static) MST algorithm

Time: $\tilde{O}(m/k)$ per changed cluster
Each update changes $O(1)$ clusters
so $\tilde{O}(m/k)$ total

Finding Inter-Cluster Swaps

Collapse each bundle or cluster to superedge

Weight of bundle superedge = min in bundle
Weight of cluster superedge = max in path

Handle weight queries using convex hull of coefficients of edge labels in bundle or cluster

Find swap by parametric search in collapsed graph

Time for parametric search: $\tilde{O}(\#\text{ bundles})$

Time to rebuild convex hulls:
$\tilde{O}(m/k)$ per changed cluster

Finding Dual-Cluster Swaps

“Ambivalent data structure” [Frederickson 1997]

For each non-tree edge endpoint, there are two tree paths inside the cluster to the two cluster exits.

Non-tree edge stores a candidate swap per exit
Found by traversing MST within cluster
querying dynamic convex hull of path edges

Each bundle stores a candidate swap per exit
the best among all swaps stored by its edges

Best dual-cluster swap found by checking which candidate is correct for each bundle, picking the best of the correct candidates

Time to update edge and bundle candidates:
$\tilde{O}(m/k)$ per changed cluster

Time to find best swap: $\tilde{O}(\#\text{ bundles})$

Analysis of Clustering

General graphs:

Total time $\tilde{O}(m/k + k^2)$
Optimal $k = \tilde{O}(m^{1/3})$

$\tilde{O}(m^{2/3})$ per MST change

Sparse (minor-closed) graph families:

Total time $\tilde{O}(n/k + k)$
Optimal $k = \tilde{O}(n^{1/2})$

$\tilde{O}(n^{1/2})$ per MST change

Idea II: Sparsification

[Eppstein, Galil, Italiano, Nissenzweig 1992]
[Fernández-Baca, Slutzki, Eppstein 1996]

Split edges of graph into two subsets

$$G = G_1 \cup G_2$$

Maintain MST of each subset
(two smaller kinetic problems)

Combine to get MST of overall graph
(one sparse structurally kinetic problem)

$$MST(G) = MST(T_1 \cup T_2)$$

Sparsification Analysis

Replaces factors of *m* by factors of *n*
in any general graph MST algorithm

But subproblems changes may not
propagate to global MST

so also replaces factors of actual MST changes
with worst-case # changes

Therefore: general graph kinetic MST
$\tilde{O}(n^{2/3})$ times worst-case # changes

Separator Based Sparsification

[Eppstein, Galil, Italiano, Spencer 1993]

Given *functionally* kinetic problem

Form separator decomposition of sparse graph

Solve MST problems on each side of separator
(two smaller functionally kinetic problems)

Use solutions to form *compact certificate*
(graph with $O(\sqrt{n})$ vertices
having same kinetic behavior as original subgraph)

Combine certificates
(one very small structurally kinetic problem)

Total time: $\tilde{O}(n^{1/4})$ per worst-case change

Conclusions and Open Problems

New kinetic MST algorithms

Some improvement to parametric MST
especially in the planar case (now $O(n^{19/12})$)
but for general graphs, still not $o(mn)$

Planar graph algorithm uses clustering
in sparsified subproblems —
can we instead use sparsification recursively?

Geometric kinetic MST?
Edge weights become quadratic instead of linear