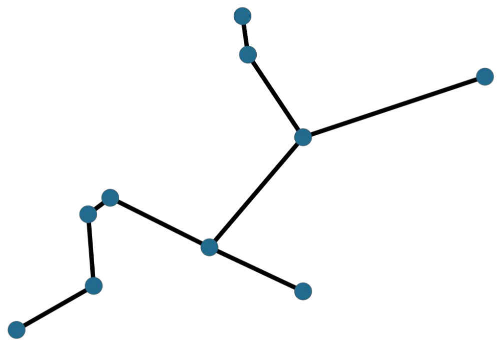
Spanning Tree, Matching, and TSP for Moving Points: Complexity and Regret

Nathan Wachholz, Subhash Suri

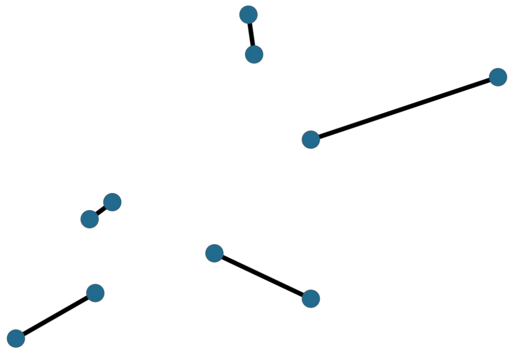
University of California, Santa Barbara

Motivation

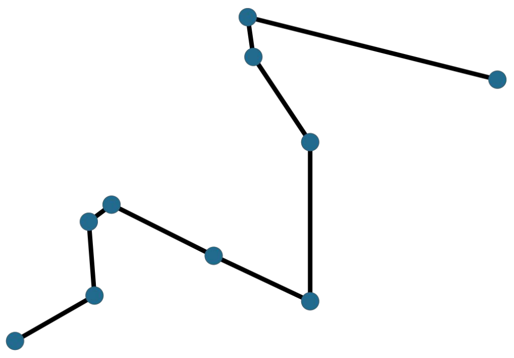
We often want to connect objects in space



Minimum Spanning Tree



Minimum Matching

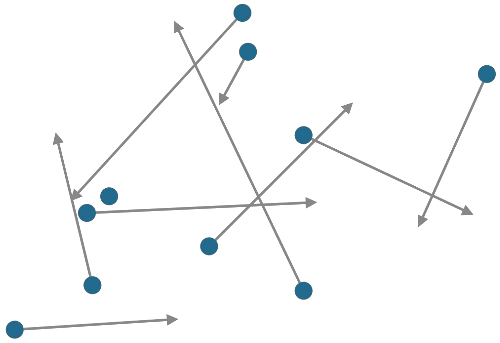


Traveling Salesman Path

Motivation

What if our objects are moving?

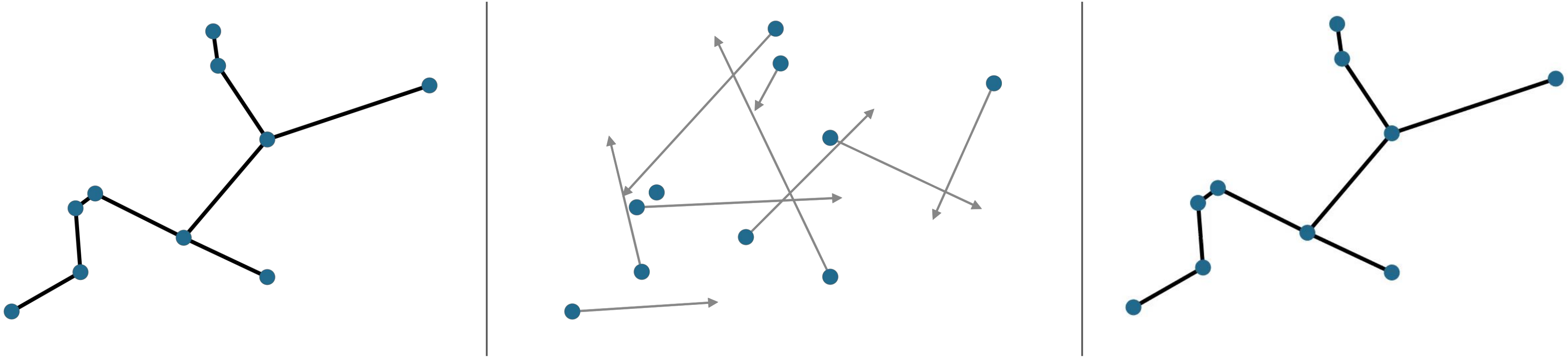
- Motion known in advance
- Straight line
- Constant speed
- $t \in [0, 1]$



Motivation

When the points move, so does the optimal structure

Updating may be impossible, or computationally expensive



Minimum Spanning Tree

Kinetic MST

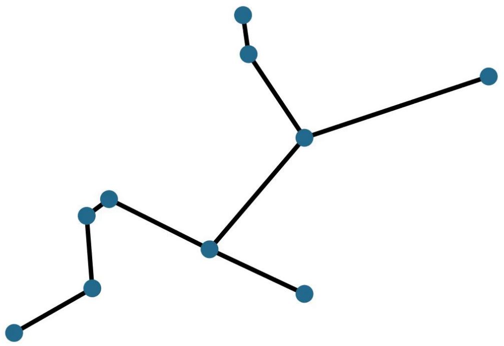
Motivation

What if we precompute a single, topologically fixed, spanning tree?

We care about the maximum length of a structure throughout the motion

Want to find the structure whose *maximum length* is *as small as possible*

Fixing the initial MST:



Motivation

This work builds upon Akitaya et. al.'s "The Minimum Moving Spanning Tree Problem" who

- Showed finding an MMST is NP-Hard in 2D
- Gave a $(2+\varepsilon)$-approximation in $O(n \log n)$ time

Our Work:

- Can motion restriction make MMST tractable?
- How bad is an MMST compared to a Kinetic MST?
- What about other structures?
- Other metrics?

Restrict points to 1D

All points are on the line

They move to the left or right

NP-Hard to find an MMST in 1D

Restrict points to 1D, unit speed

All points are on the line

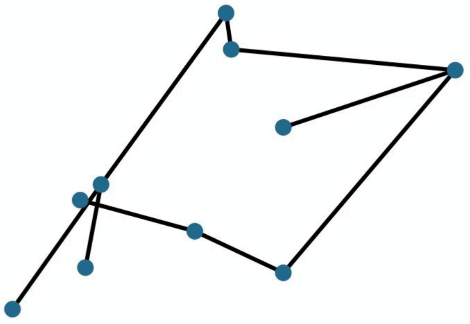
They move *1 unit* to the left or *1 unit* to the right

NP-Hard to find an MMST in 1D under unit speed

Consider the *regret* of a structure

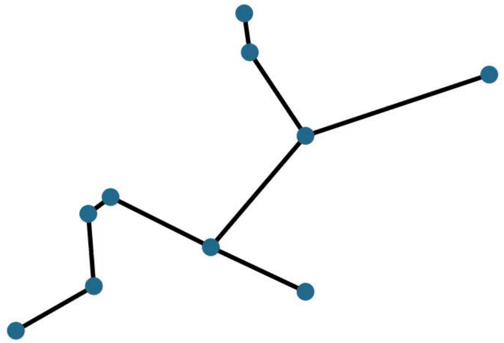
How bad is a single structure compared to an updating one?

- w(T) = the max cost of static tree T
- w(K) = the max cost of a Kinetic MST
- Regret of T is r(T) = w(T) / w(K)



MMST

vs



Kinetic MST

Consider the *regret* of a structure

There exists a set of *n* points in 1D with regret $\Omega(\sqrt{n})$

Regret is unbounded in 1D

But under unit speed motion, constant regret bound

Regret is strictly 2-bounded in 1D under unit speed

Other structures: Minimum Matching, Traveling Salesman

Also NP-Hard in 1D under unit speed

Also have unbounded regret in 1D at arbitrary speeds

TSP regret is 2-bounded for unit speed on a line

Matching regret is not 2-bounded (we don’t know what it is; 2.2 is our worst so far)

Other metrics

Original paper gave an alternative metric that was tractable:

- Minimum bottleneck moving spanning tree (MBMST)
- Naively $O(n^2)$, recent paper brought this to $O(n^{4/3} \log^3 n)$

We introduced another intuitive metric:

- Minimum average moving spanning tree (MAMST)
- Also naively $O(n^2)$

1D MMST Hardness Reduction

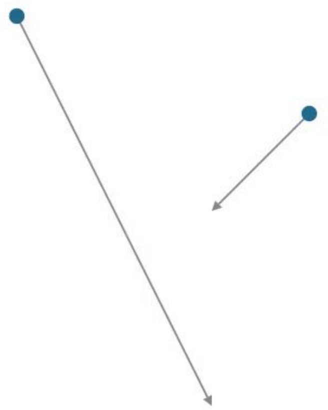
Let’s take a peek at the details of this reduction

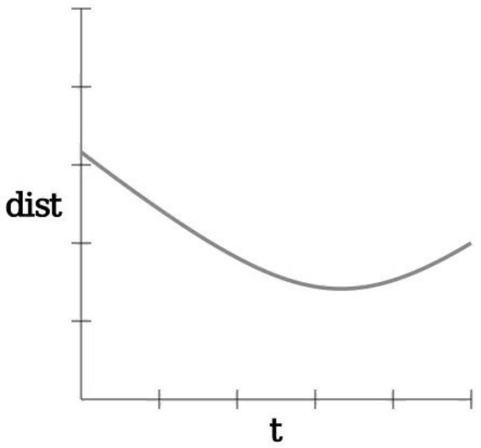
MMST is Hard in 1D - Preliminaries

Akitaya et. al. showed the distance between moving points is convex

The weight of a tree is a convex function

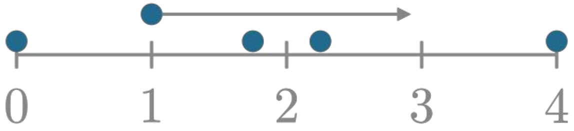
Therefore the maximum weight of a tree occurs at *t=0* or *t=1*





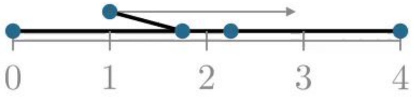
MMST is Hard in 1D - Single Gadget

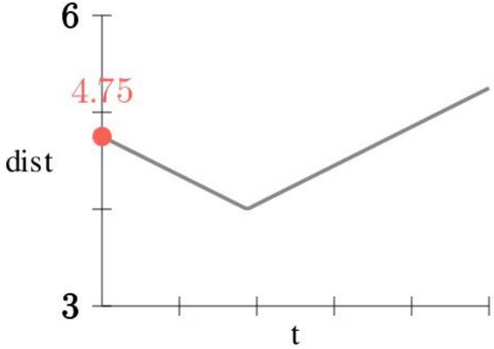
Consider an MMST on the following 5 moving points in 1D



MMST is Hard in 1D - Single Gadget

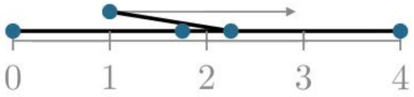
Then there are two locally minimal options:

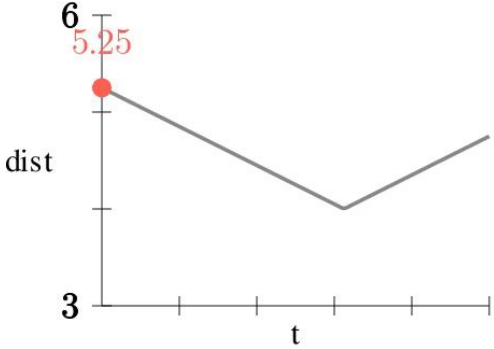




Initial: $w_T(0) = 4.75$

Final: $w_T(1) = 5.25$





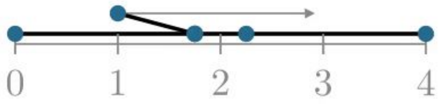
Initial: $w_T(0) = 5.25$

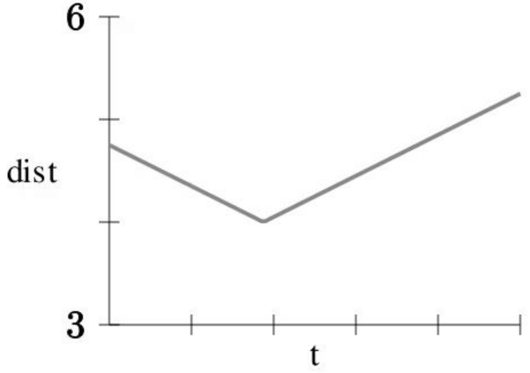
Final: $w_T(1) = 4.75$

MMST is Hard in 1D - Single Gadget

We can modify the points to adjust the costs

The difference between $w_T(0)$ and $w_T(1)$ can vary from 0 to 0.5

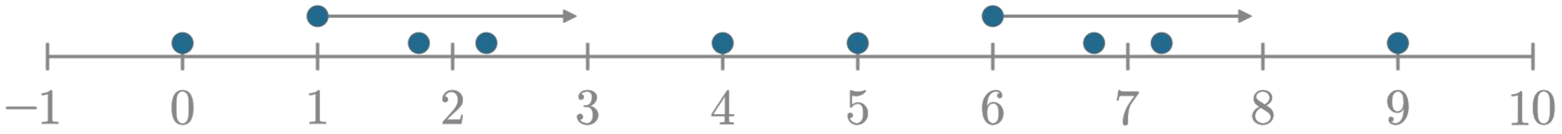




MMST is Hard in 1D - Multiple Gadgets

What if we have multiple gadgets?

Each gadget has 2 locally optimal configurations

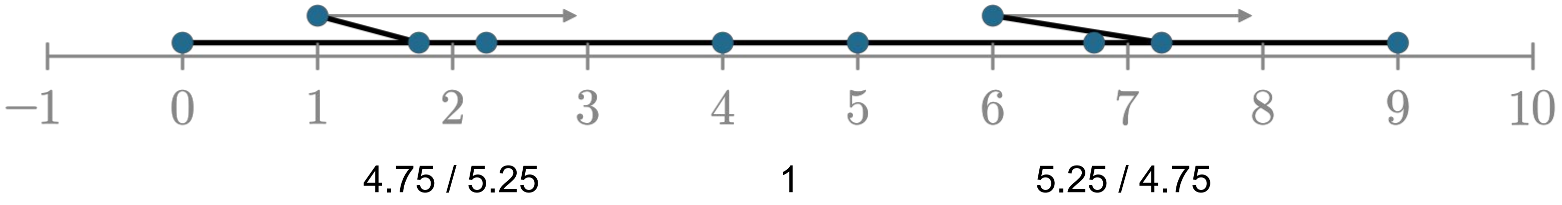


MMST is Hard in 1D - Multiple Gadgets

An MMST is a combination of the subtrees on each gadget

Maximum cost of the tree is determined by the configuration of the gadgets

In this example, $w_T(0) = w_T(1) = w(T) = 10 + 1 = 11$



MMST is Hard in 1D - Partition Problem

We will reduce from the Partition Problem, which is NP-Hard:

Given a list $(a_0, a_1, \ldots, a_{n-1})$ of n integers,

Partition them into two lists that have the same sum

Example:

(1, 2, 5, 6, 10) → (1, 5, 6) and (2, 10)

We can also normalize:

(0.1, 0.2, 0.5, 0.6, 1) → (0.1, 0.5, 0.6) and (0.2, 1)

MMST is Hard in 1D - Putting it together

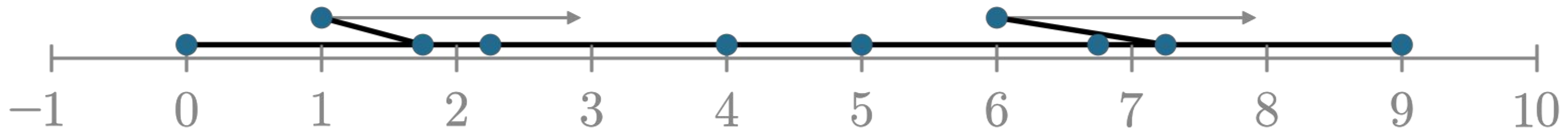
Now create a gadget S_i for each value a_i (normalized)

Consider an MMST on the gadgets

There is a subtree T_i on each gadget, in one of two configurations:

- Left: $w_{Ti}(0) = 5 - a_i/4$ and $w_{Ti}(1) = 5 + a_i/4$
- Right: $w_{Ti}(0) = 5 + a_i/4$ and $w_{Ti}(1) = 5 - a_i/4$

Let $L = \{ i \mid T_i \text{ is Left} \}$ and $R = \{ i \mid T_i \text{ is Right} \}$



MMST is Hard in 1D - Putting it together

Let $L = \{ i \mid T_i \text{ is Left} \}$ and $R = \{ i \mid T_i \text{ is Right} \}$

- Left: $w_{Ti}(0) = 5 - a_i/4$ and $w_{Ti}(1) = 5 + a_i/4$
- Right: $w_{Ti}(0) = 5 + a_i/4$ and $w_{Ti}(1) = 5 - a_i/4$

The cost of all subtrees is:

$$w(0) = 5n - \sum_{i \in L} \frac{a_i}{4} + \sum_{i \in R} \frac{a_i}{4} \qquad w(1) = 5n + \sum_{i \in L} \frac{a_i}{4} - \sum_{i \in R} \frac{a_i}{4}$$

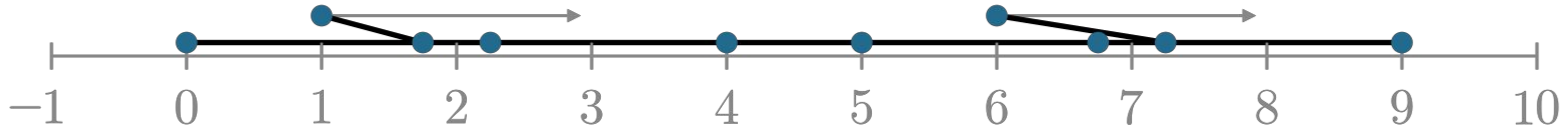
$$w(\text{subtrees}) = \max\{w(0), w(1)\} = 5n + \left| \sum_{i \in L} \frac{a_i}{4} - \sum_{i \in R} \frac{a_i}{4} \right|$$

MMST is Hard in 1D - Putting it together

So we find an MMST, and ask if its weight is 6n - 1

If it is, then we know there is a solution to the partition problem

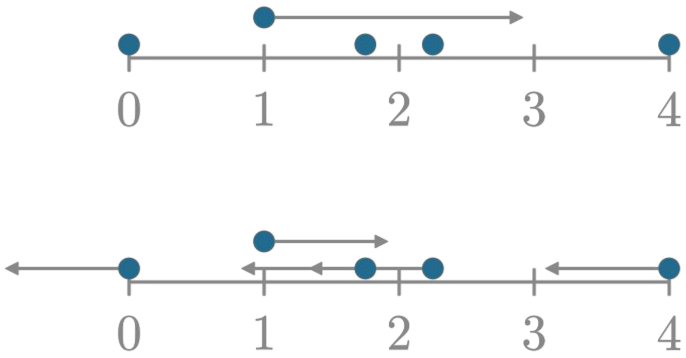
Can easily get the solution by looking at the MMST



MMST is Hard in 1D

MMST is Hard in 1D!

Even if all points move at the same speed!



Regret is Unbounded for General 1D Motion

We can construct a set of points to achieve any desired regret ratio

Say our target regret is $b = 2$. Define $k = 2b = 4$.

Let $p_{i,j}$ denote a 1D moving point that starts at i and ends at j.

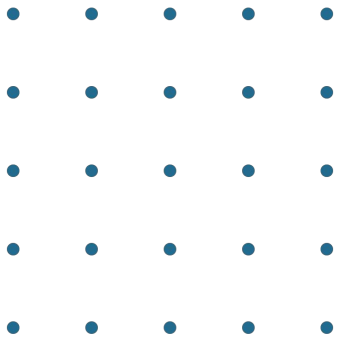
Let S be the set of all $p_{i,j}$ for $0 \leq i, j \leq k$.

Here's what they look like for $b = 2$:

Regret is Unbounded for General 1D Motion

Let's imagine each 1D moving point as a static point in 2D

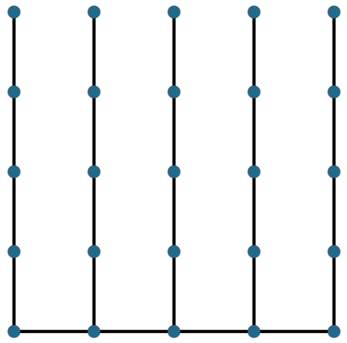
Put each $p_{i,j}$ at $P_{i,j} = (i, j)$. Now we have a grid:



Project onto the *x*-axis to get initial positions, or the *y*-axis for final positions

Regret is Unbounded for General 1D Motion

Now imagine a tree on the 2D points:



Initial: $w_T(0) = 4$

Final: $w_T(1) = 20$

Map the tree back to the 1D points and:

- initial cost is the sum of its horizontal components
- final cost is the sum of its vertical components

Regret is Unbounded for General 1D Motion

The best we could do is w(T) = 12.

A kinetic MST will always have cost w(K) = 4:

So in this example, regret is always at least 3 > b = 2.

Regret is Unbounded for General 1D Motion

We can generalize for any b > 1. Define k = 2b.

Any fixed tree has a maximum cost of at least $((k+1)^2-1)/2 > k^2/2$

A kinetic MST has fixed cost k

So regret is at least $k^2/2k = k/2 = b$

The construction with n points gives a $\Omega(\sqrt{n})$ regret ratio for general 1D motion

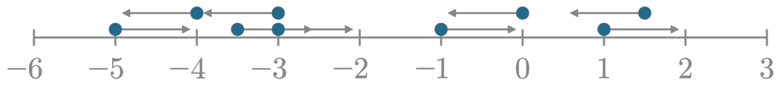
Regret is Bounded for *Unit Speed* in 1D

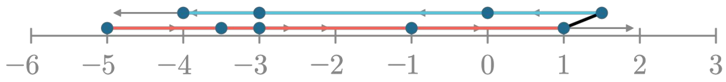
Regret of an MMST on 1D, unit speed moving points is no more than 2

Basic idea:

- Connect Leftwards-moving points
- Connect Rightwards-moving points
- Add an edge connecting them
- Each is always less than the dynamic cost (3-bound)

Being more careful gives a 2-bound





Open Problems

Regret bound of unit speed minimum matching

Regret bound in higher dimensions (unit speed or otherwise)

MAMST faster than $O(n^2)$

Partially-kinetic structures (allow n updates to the structure)

Thank you!

Any questions?