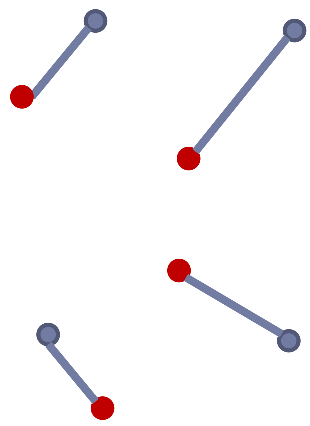
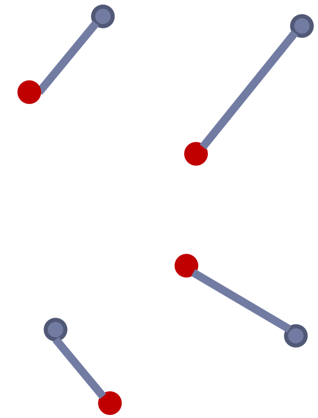
Optimal planar transport in near-linear time



Alex Andoni
(Columbia U)

Joint with: Aleksandar Nikolov (Toronto U),
Krzysztof Onak (IBM TJ Watson),
Grigory Yaroslavtsev (Indiana U, Bloomington)

Planar transport



- Problem:
  - Given two sets A, B of points in $\mathbb{R}^2$,
  - Find min-cost matching
  - weighted sets of size n
  - a.k.a., Earth-Mover Distance, Wasserstein metric…
- Classically: LP with n^2 variables
  - Best time: $\tilde{O}(n^2/\epsilon^4)$ for $1+\epsilon$ approx [Altschuler-Weed-Rigolet'17]
  - But can hope for $\ll n^2$ runtime!
    - input is of size $O(n)$

$$\min_{\pi \in \mathbb{R}_+^{n^2}} \sum_{ij} ||p_i - q_j|| \cdot \pi_{ij}$$

$$\text{s.t. } \pi \mathbf{1} = \frac{1}{n}\mathbf{1}$$

$$\pi^t \mathbf{1} = \frac{1}{n}\mathbf{1}$$

Main result:

For fixed ϵ, can solve in $n^{1+o(1)}$ time

Brief history of planar transportation

- Long history of work [Vaidya'89, Agarwal-Efrat-Sharir'00, Varadarajan-Agarwal'99, Agarwal-Varadarajan'04, Indyk'07, Sharathkumar-Agarwal'12]
  - Transportation: $O(1)$ approximation in $\tilde{O}(n)$ time
  - EMD (unweighted): $1+\epsilon$ approximation in $\tilde{O}(n)$ time

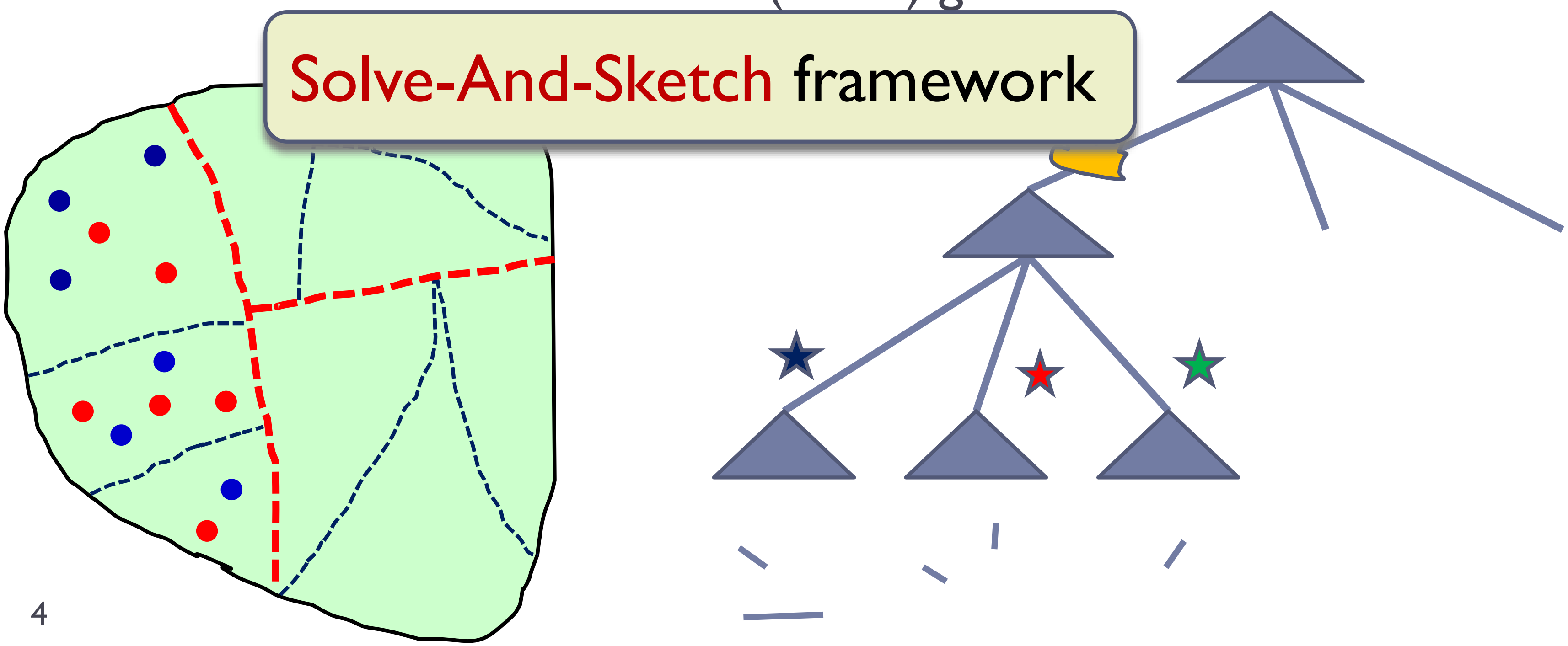
- High-dimensional case: points in $\mathbb{R}^d$
  - $\exp(d)$ slow-down…

  - $O(\log n \cdot \log d)$ approximation in $\tilde{O}(n)$ time [Charikar'02, Indyk-Thaper'04, Grauman-Darell'05, A-Indyk-Krauthgamer'07]
  - Via embedding into ℓ_1
  - Also useful for efficient Nearest Neighbor Search

Approach: Sketching + Divide-And-Conquer

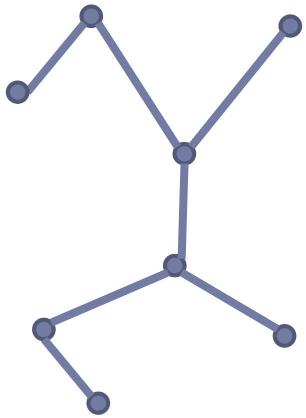
- Partition the space hierarchically in a "nice way"
- In each part
  - Compute a "local solution" for the local view
  - Sketch the solution using small space
  - Combine local sketches into (more) global solution

Solve-And-Sketch framework



Detour: Minimum Spanning Tree

- Find MST of an implicit graph on n points in $\mathbb{R}^d$



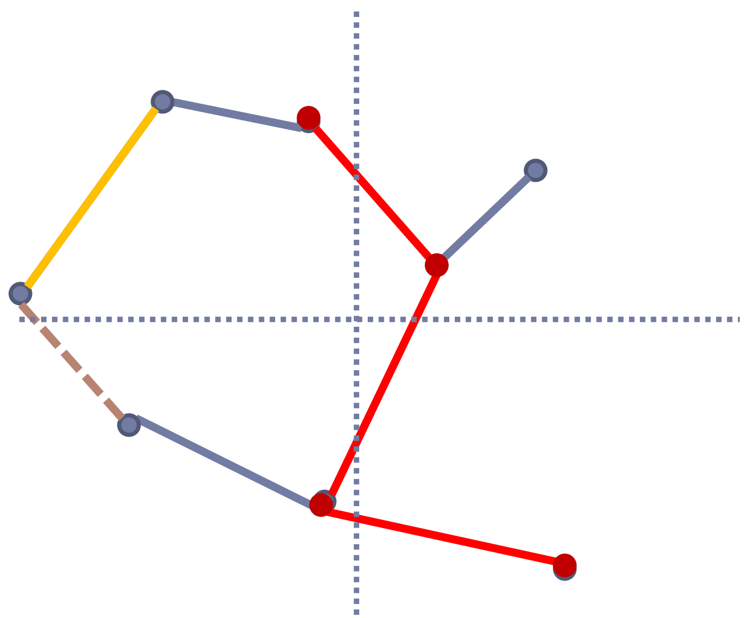
MST via Solve-and-Sketch: try 1

- Partition the space hierarchically in a "nice way"
- In each part
  - Compute a "local solution" for the local view
  - Sketch the solution using small space
  - Combine local sketches into (more) global solution

quad trees!

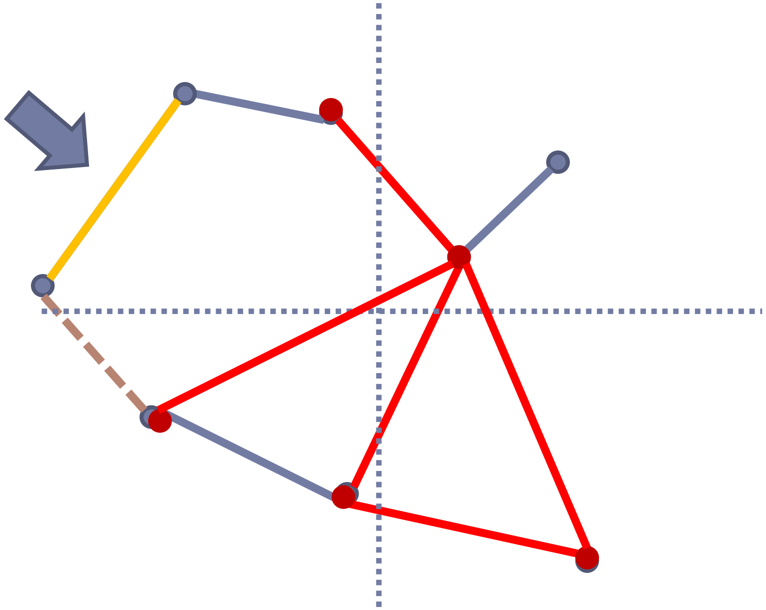
local MST

a representative point
(to connect to rest)



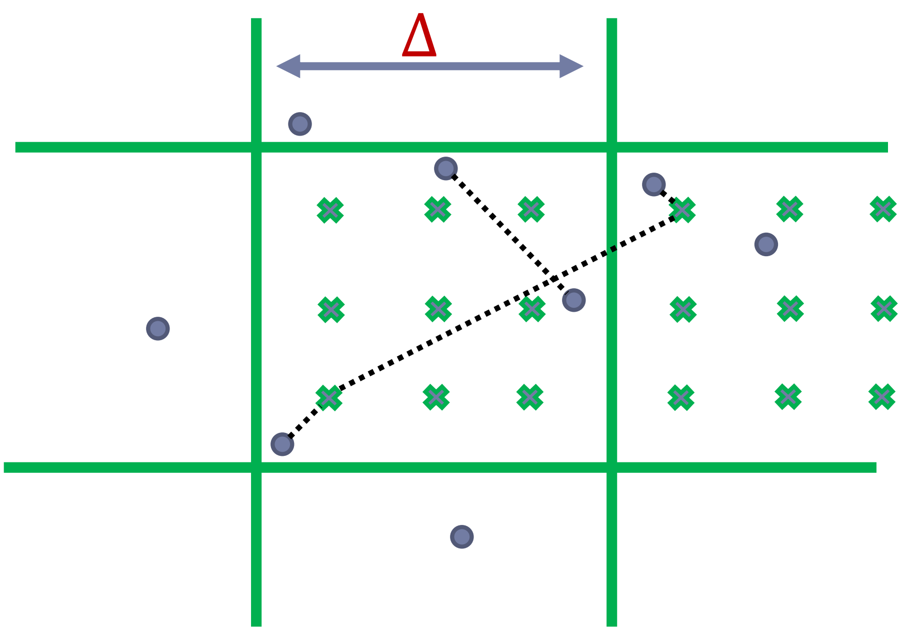
Not optimal MST:

- Quad tree can cut MST edges
  - forcing irrevocable decisions
- Choose a wrong representative



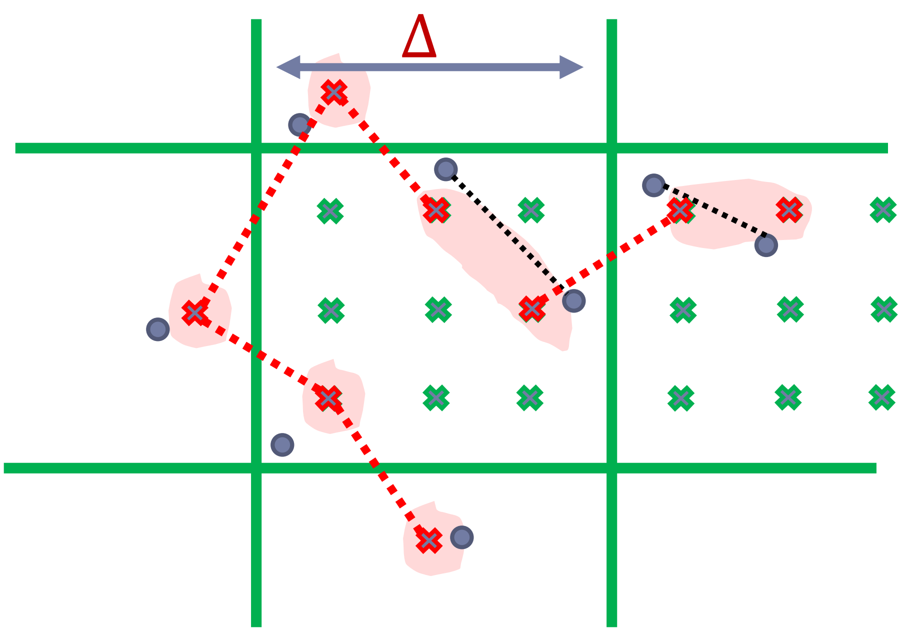
New Partition: Grid Distance

- Randomly shifted grid [Arora'98, …]
- Each cell has an $\epsilon\Delta$-net N
- Net points are entry/exit portals for the cell
- Claim: all distances preserved up to $1 + \epsilon$ in expectation



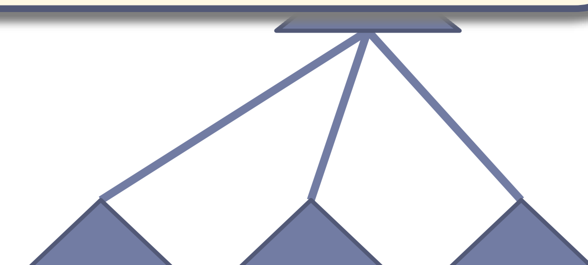
MST: S-a-S algorithm

- Assume entire pointset in a square of size $O(\sqrt{n}) \times O(\sqrt{n})$
- & partition into 2 levels only:
  - Randomly-shifted grid with $\Delta = n^{1/4}$
- Local solution:
  - Run Kruskal for edges up to length $\epsilon\Delta$
- Sketch of the local solution:
  - Snap points to $\epsilon^2\Delta$-net, and store their connectivity $\Rightarrow$ size $s = O\left(\frac{1}{\epsilon^4}\right)$



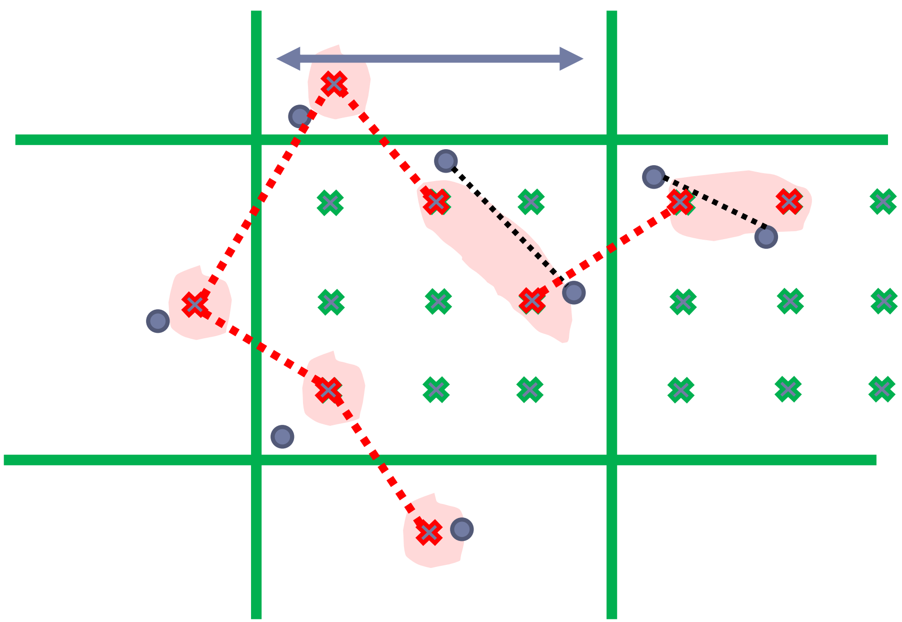
Runtime:

- $\sqrt{n}$ leaf cells: $O(\Delta^4)$ time each
- root: $O\left(\left((\sqrt{n}/s)^2 \cdot s\right)^2\right)$
- total: $O(n \cdot \sqrt{n})$



MST Analysis

- Equivalent to running Kruskal on the *grid distance*
- Any distance between cells is $\geq \epsilon\Delta$
  - 1) Safe to run Kruskal "locally" inside each cell up to this threshold!
  - 2) Snapping to $\epsilon^2\Delta$-net points costs a little bit only

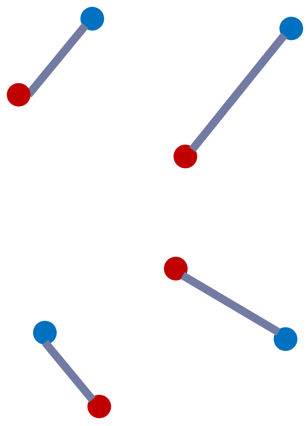


Runtime:

- $\sqrt{n}$ leaf cells: $O(\Delta^4)$ time each
- root: $O\left(\left((\sqrt{n}/s)^2 \cdot s\right)^2\right)$
- total: $O(n \cdot \sqrt{n})$

Back to Optimal Transport (EMD)

- Theorem:
  - $1+\epsilon$ cost approximation in $\mathbb{R}^d$ space
  - $n^{1+o(1)}$ time for constant ϵ, d



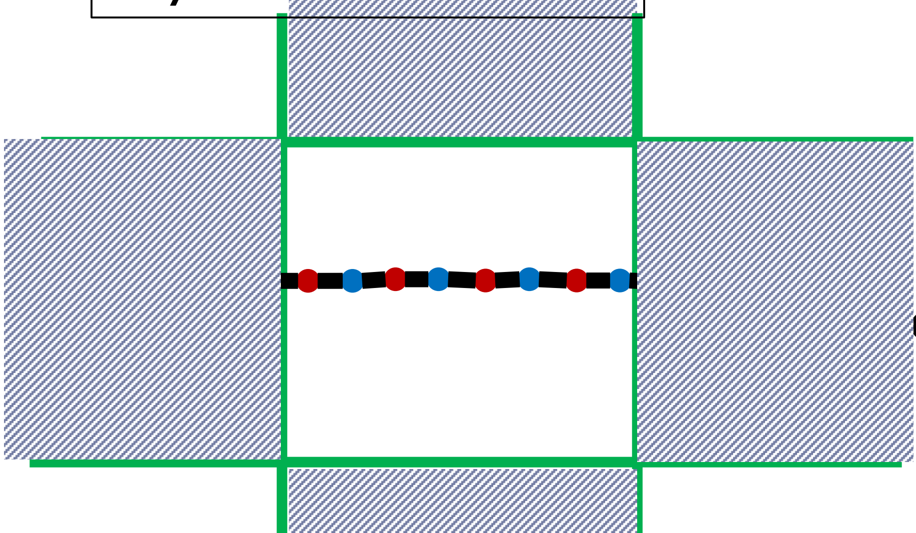
Solve-And-Sketch for EMD

- Partition the space hierarchically in a "nice way"
- In each part
  - Compute a "local solution" for the local view
  - Sketch the solution using small space
  - Combine local sketches into (more) global solution

all potential local solutions

cannot precompute *any* "local solution"

fat quad-tree (as before) & use *grid distance*

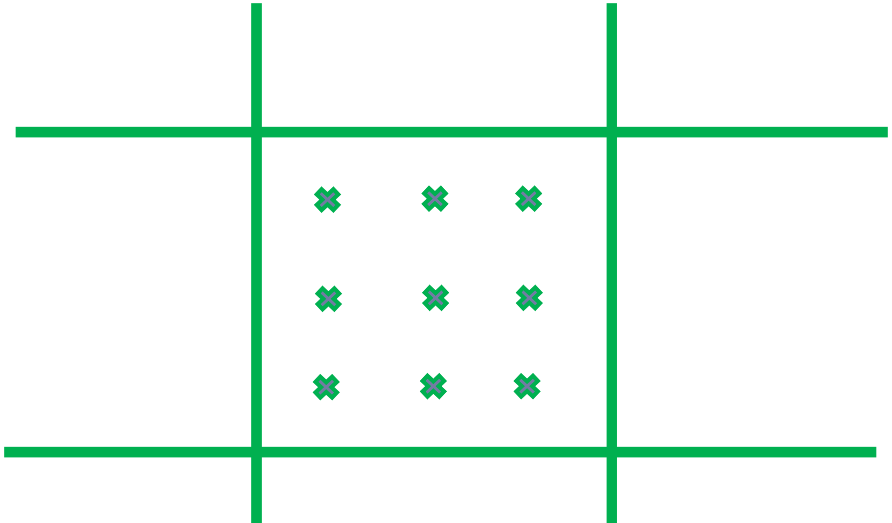


committing to a wrong alternation,
ot get <2 approximation!

Sketching ALL local solutions

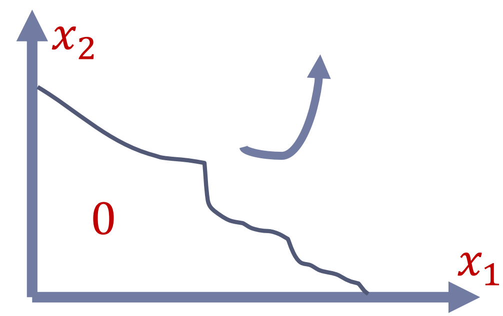
- Let k =size of ϵ-net (“portal” points)
- Define “solution function” $F: \mathbb{R}^k \to \mathbb{R}_+$
  - interpret $x \in \mathbb{R}^k$ as the *interface flow* (matching) at the portals
  - $F(x) =$ min-cost matching assuming flow x at portals

- Goal: sketch entire function F !



Sketching solution function F

- We do not know if is possible…
- Approach ?
  - Prove "cool properties" about F
  - Show: $\forall F$ with "cool properties" => sketchable
- May require $\Omega(n)$ even for (generic) F satisfying:
  - convex
  - Δ-Lipshitz
  - $k = 2$ inputs



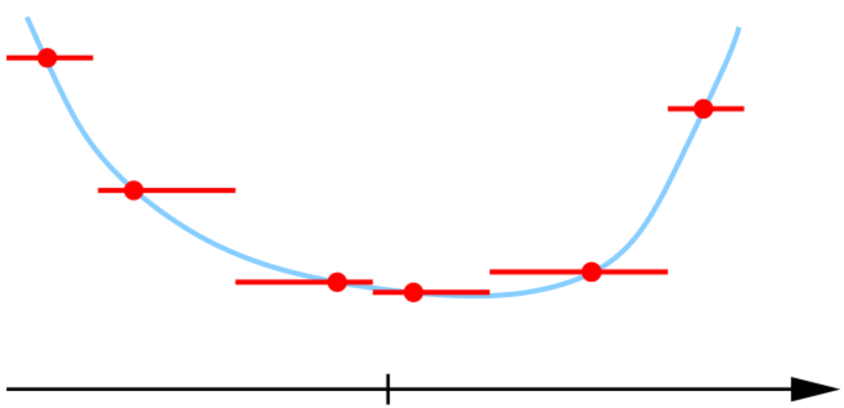
But can for:

$$F'(x) = F(x) + \epsilon\Delta \cdot ||x||_1$$

Sketching solution function F'

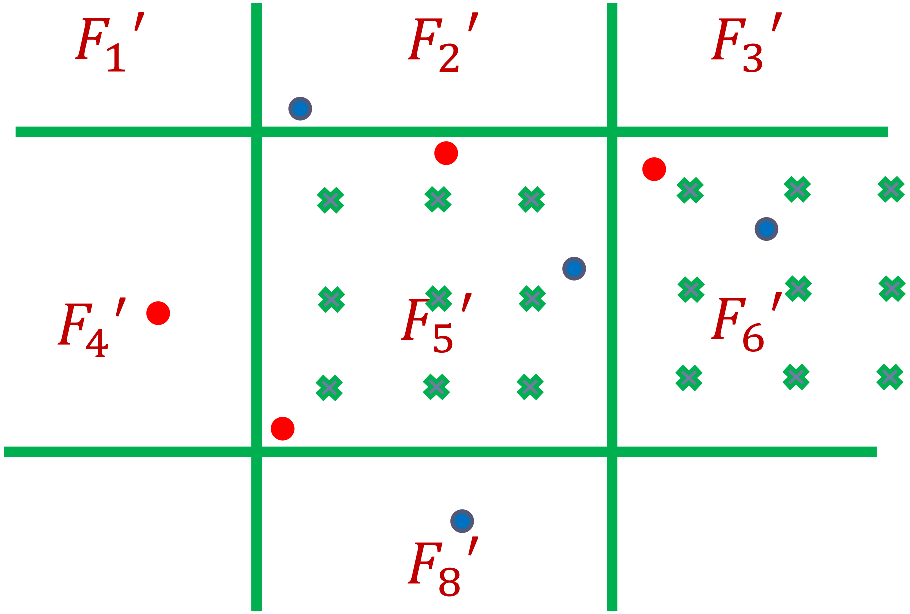
$$F'(x) = F(x) + \epsilon\Delta \cdot ||x||_1$$

- Lemma: can $(1+\epsilon)$-sketch F' using space $\left(\frac{\log n}{\epsilon}\right)^{O(k)}$
  - sketch: $F'(x)$ for x with each coordinate $x_i =$ power of $1+\epsilon$
- Why is regularization term ok?
  - part of the cost at next level up: distance between cells $\geq \epsilon\Delta$
- Done! F captures all information from the cell
  - if we don't care about runtime…



Local runtime: polynomial

- When combining: input = sketches *F'*
  - need to extrapolate *F'* from sketches
- Solution 1: formulate as a convex program
  - extrapolate F' by computing lower convex hull
- Solution 2: lower convex hull+LP = fancier LP



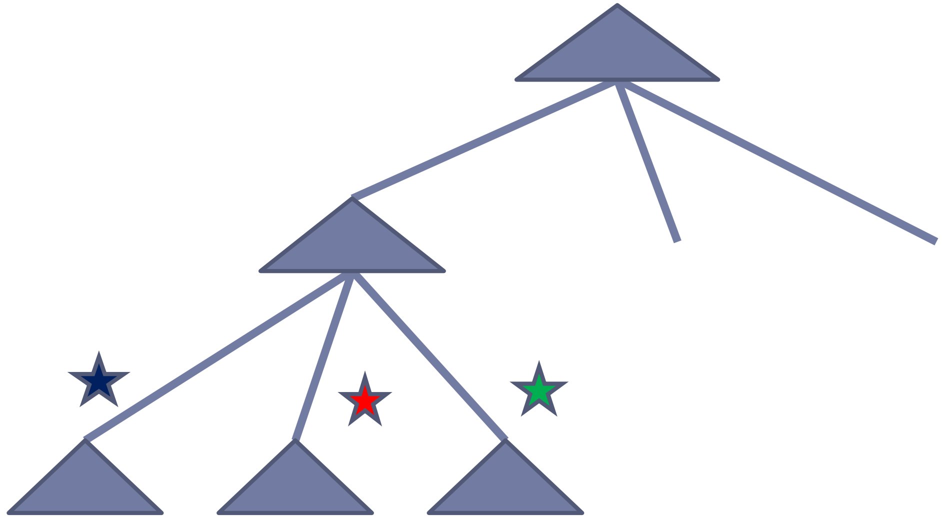
variables: flows between portals

min $\sum_i F_i'$(projection of vars on cell i)
subject to:
flow preservation conditions

size: $poly(S)$

Overall algorithm

- Decompose the LP into many small ones
  - small ones have size $\leq n^{o(1)}$
  - ok to use any poly-time LP solver
  - recompose hierarchically (divide and conquer)
- Enabling tool: sketching the solution function!



Wrap-up

- Optimal low-d transport in near-linear time
  - Solve-And-Sketch framework = divide-and-conquer the LP
  - Parallelizable
  - Generalizes to d-dimensional case (even doubling)
- Some open questions:
  - Sketch "solution function" in $poly\left(\frac{\log n}{\epsilon}\right)$ space?
  - Near-linear time for 2-Wasserstein metric?
    - Best known: $\tilde{O}(n^{1.5})$ [Phillips-Agarwal'06, Agarwal-Sharathkumar'14]
    - Evidence it is a harder problem, for $\mathbb{R}^3$ [A-Naor-Neiman'16]
  - Other problems? Which LPs are decomposable?