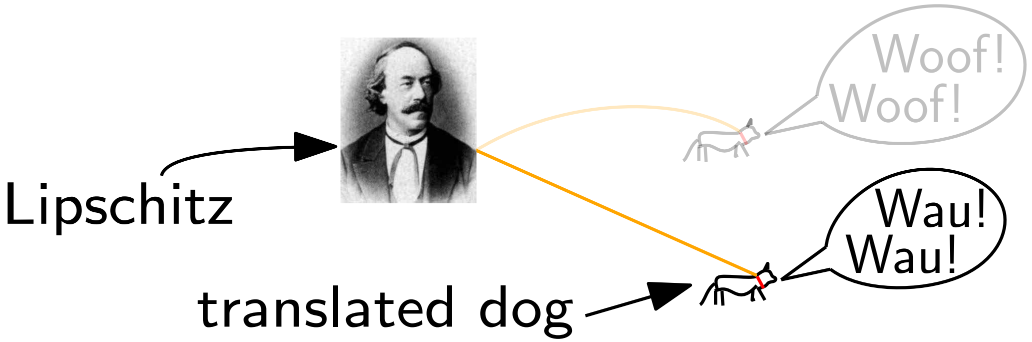
max planck institut
informatik

When Lipschitz Walks Your Dog: Algorithm Engineering of the Discrete Fréchet Distance under Translation

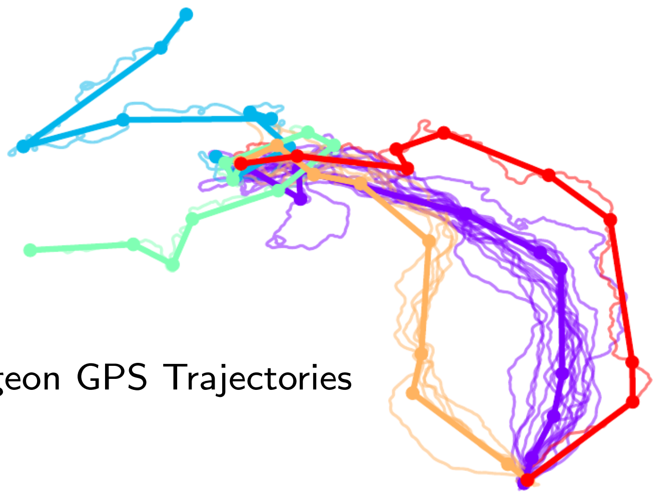
Karl Bringmann, Marvin Künnemann, and **André Nusser**

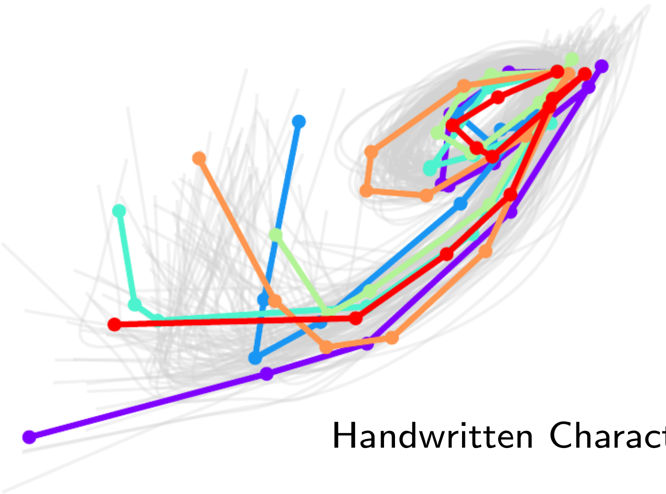


Teaser

Teaser

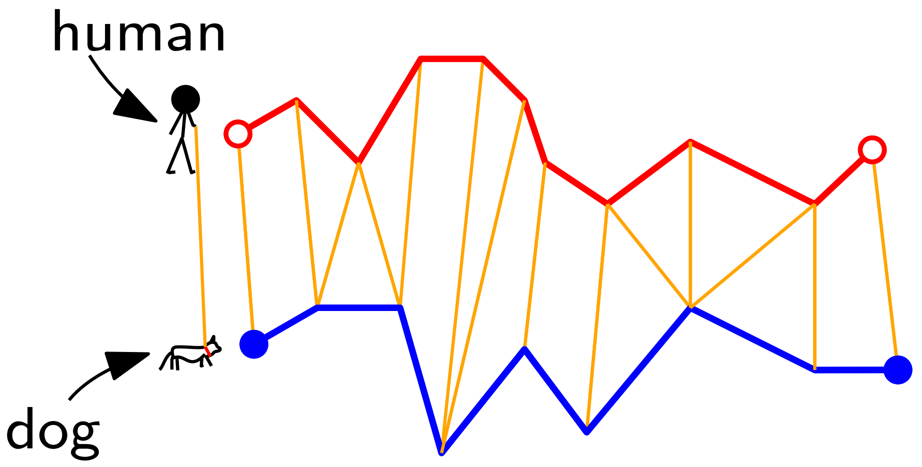
Trajectory Similarity:





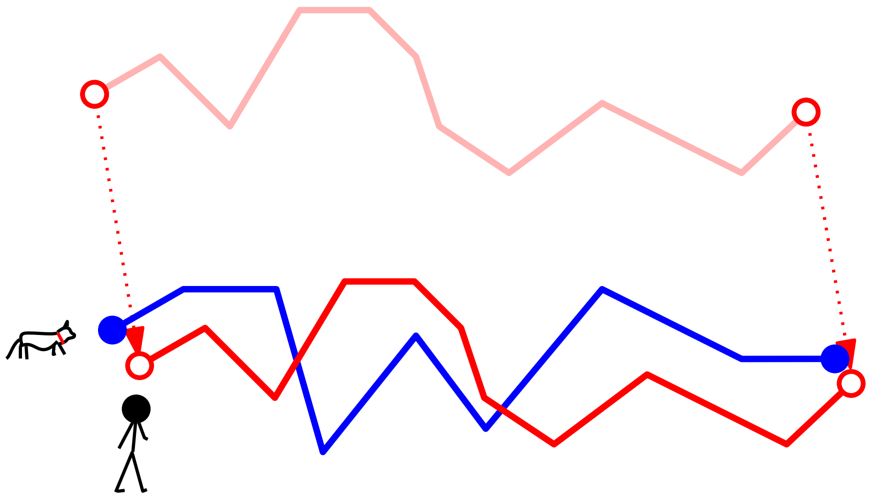
Fréchet Distance:

- traversal based
- fast in practice



Fréchet Distance Under Translation:

- traversal based
- only impractical algorithms (before)
- translation invariant



Karl Bringmann, Marvin Künnemann, and **André Nusser**

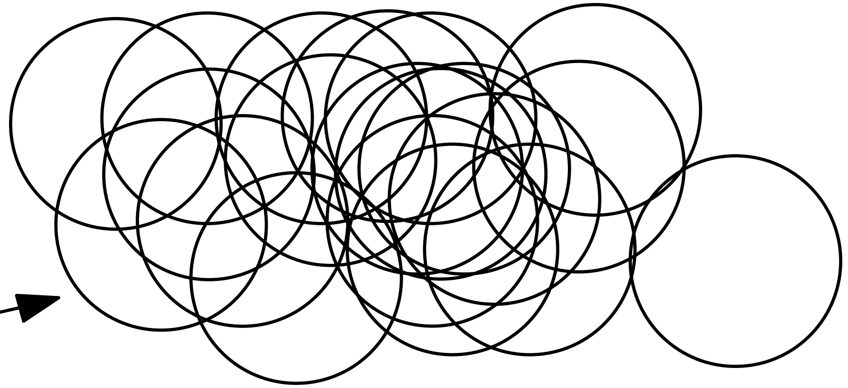
max planck institut informatik

Teaser

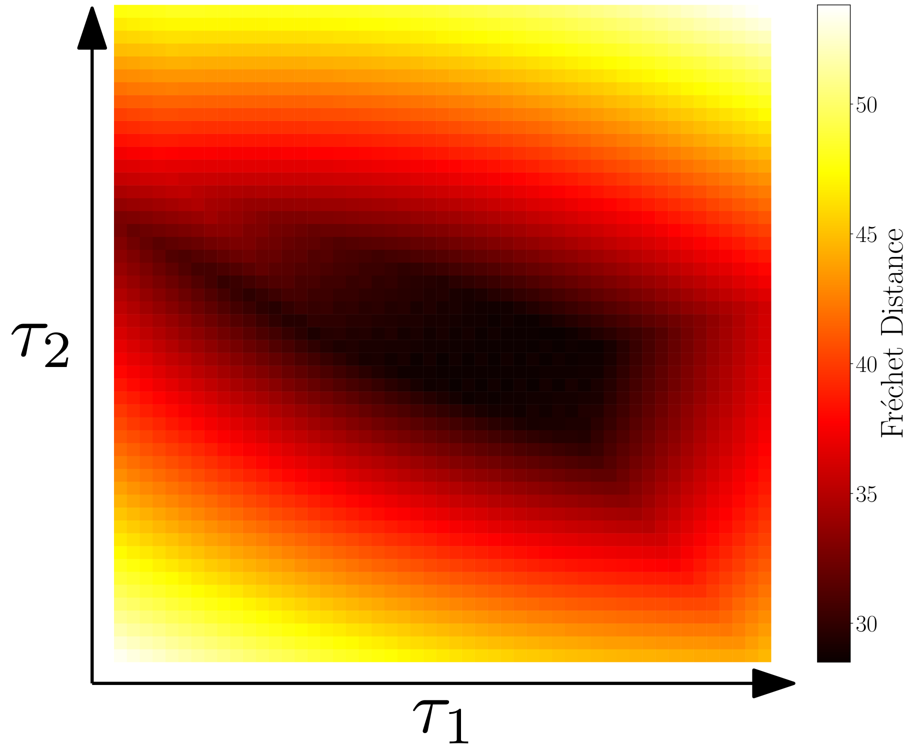
Best algorithm: $\mathcal{O}(n^{4.66})$

Conditional lower bound: $n^{4-o(1)}$

↳ All algorithms build $\mathcal{O}(n^4)$ arrangement!



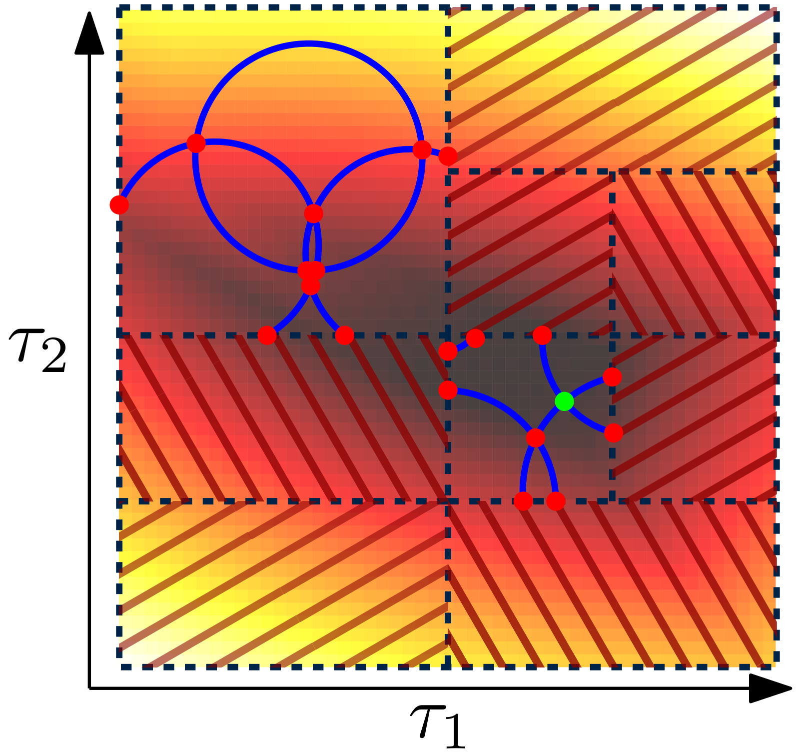
Fréchet under translation is 1-Lipschitz in τ!



Use continuous optimization:

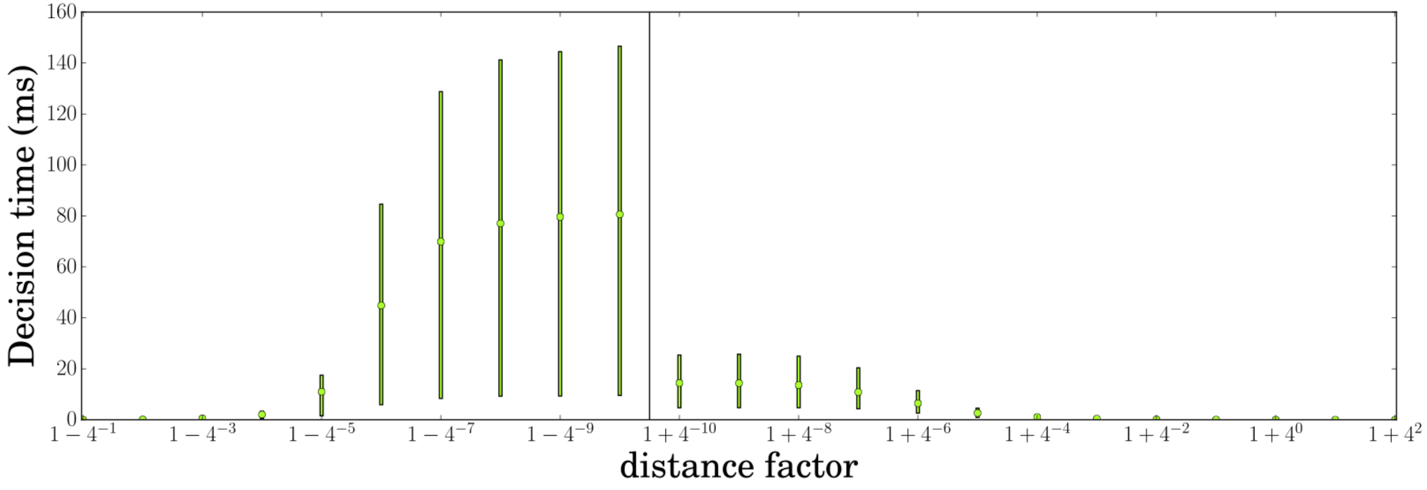
- branch & bound!

Lipschitz Meets Fréchet:

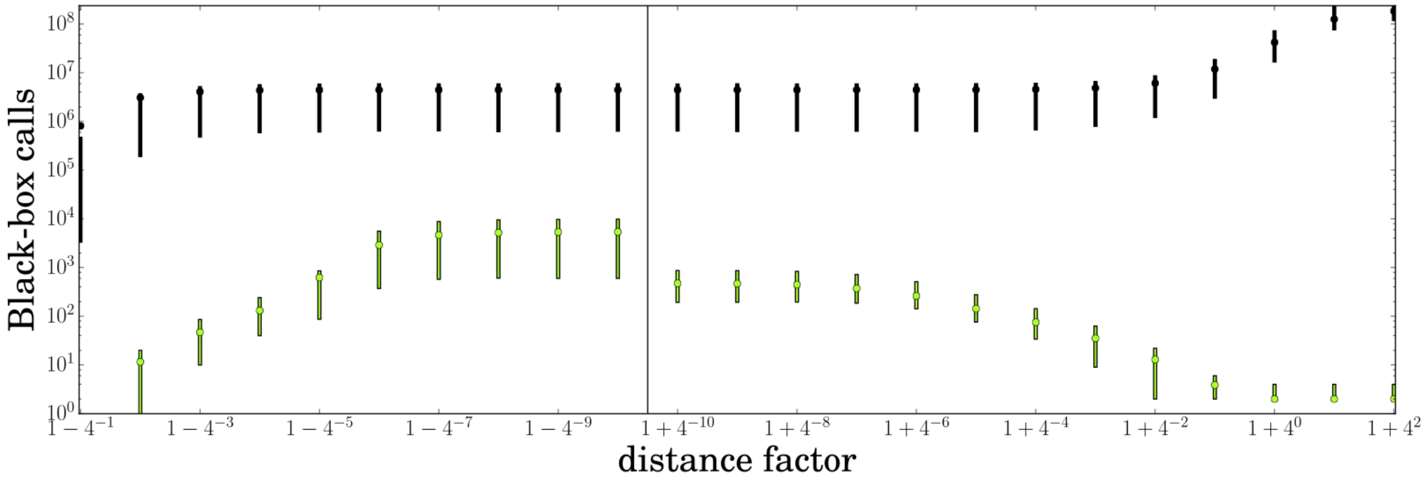


Teaser

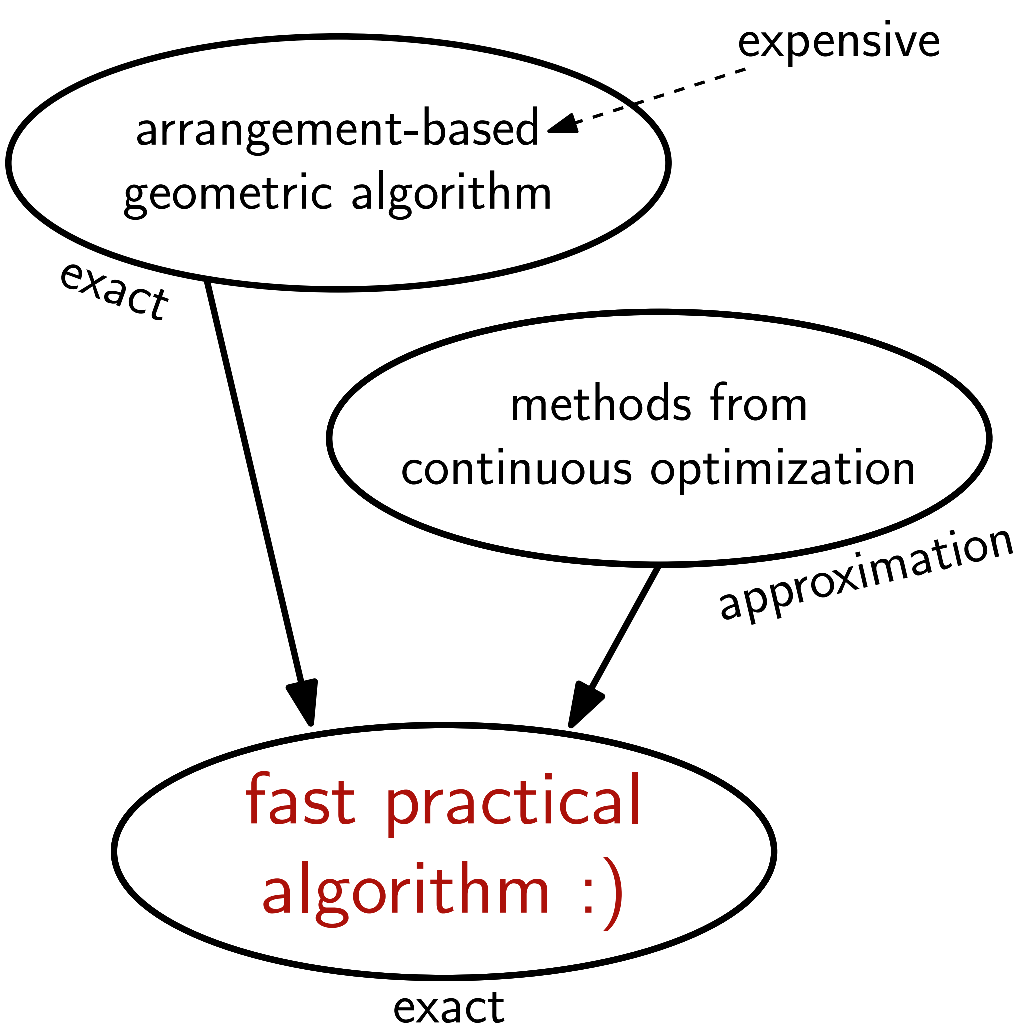
SAME-CHARACTERS:



SAME-CHARACTERS:



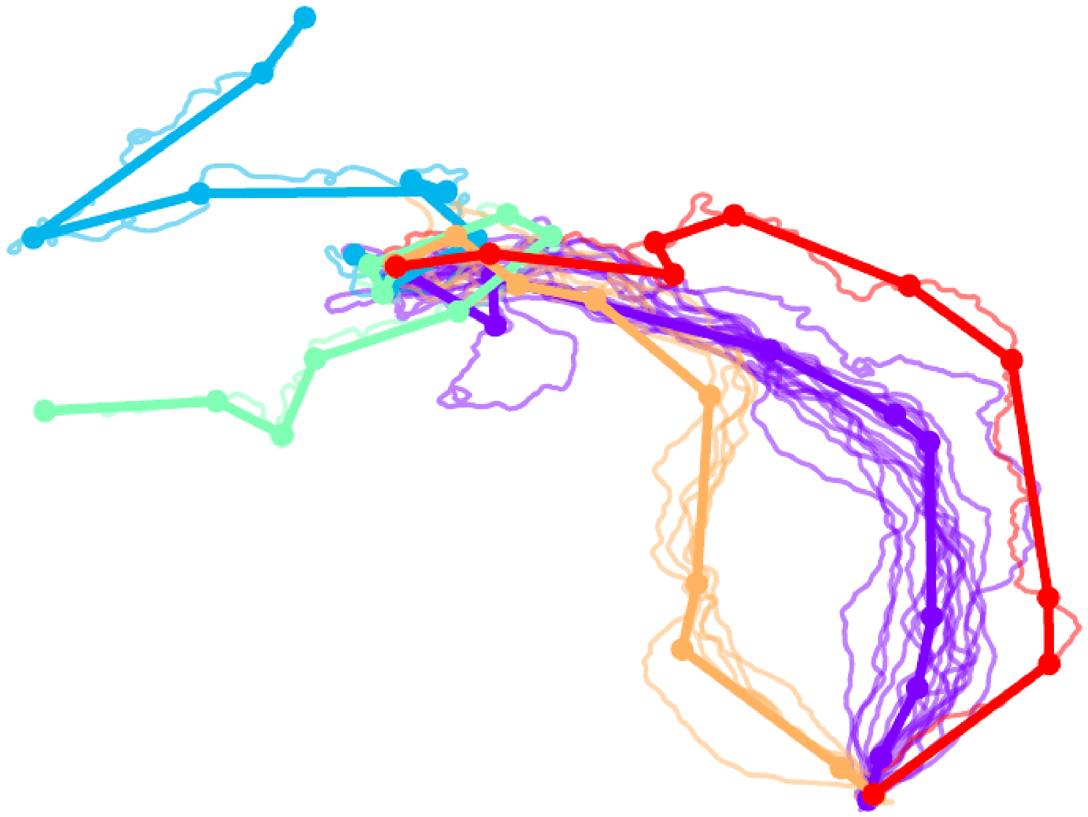
Take-Home Message:



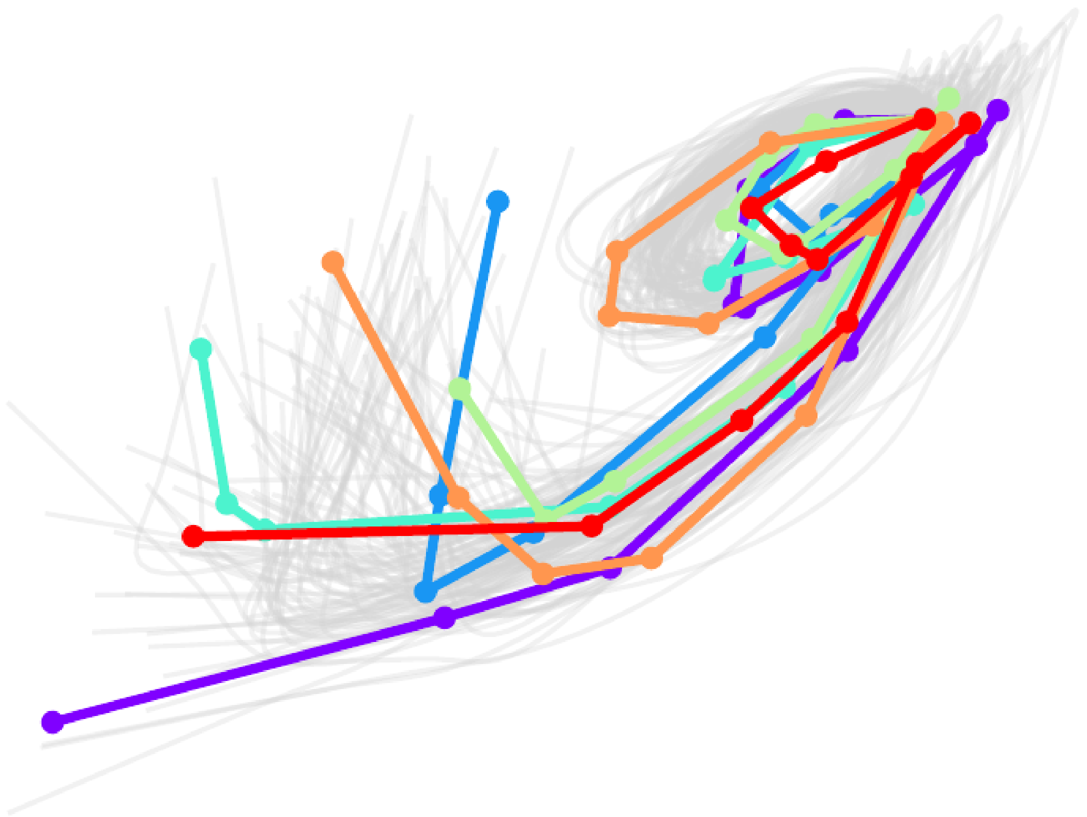
End of Teaser

Why Trajectory Similarity?

Pigeons' GPS Trajectories:

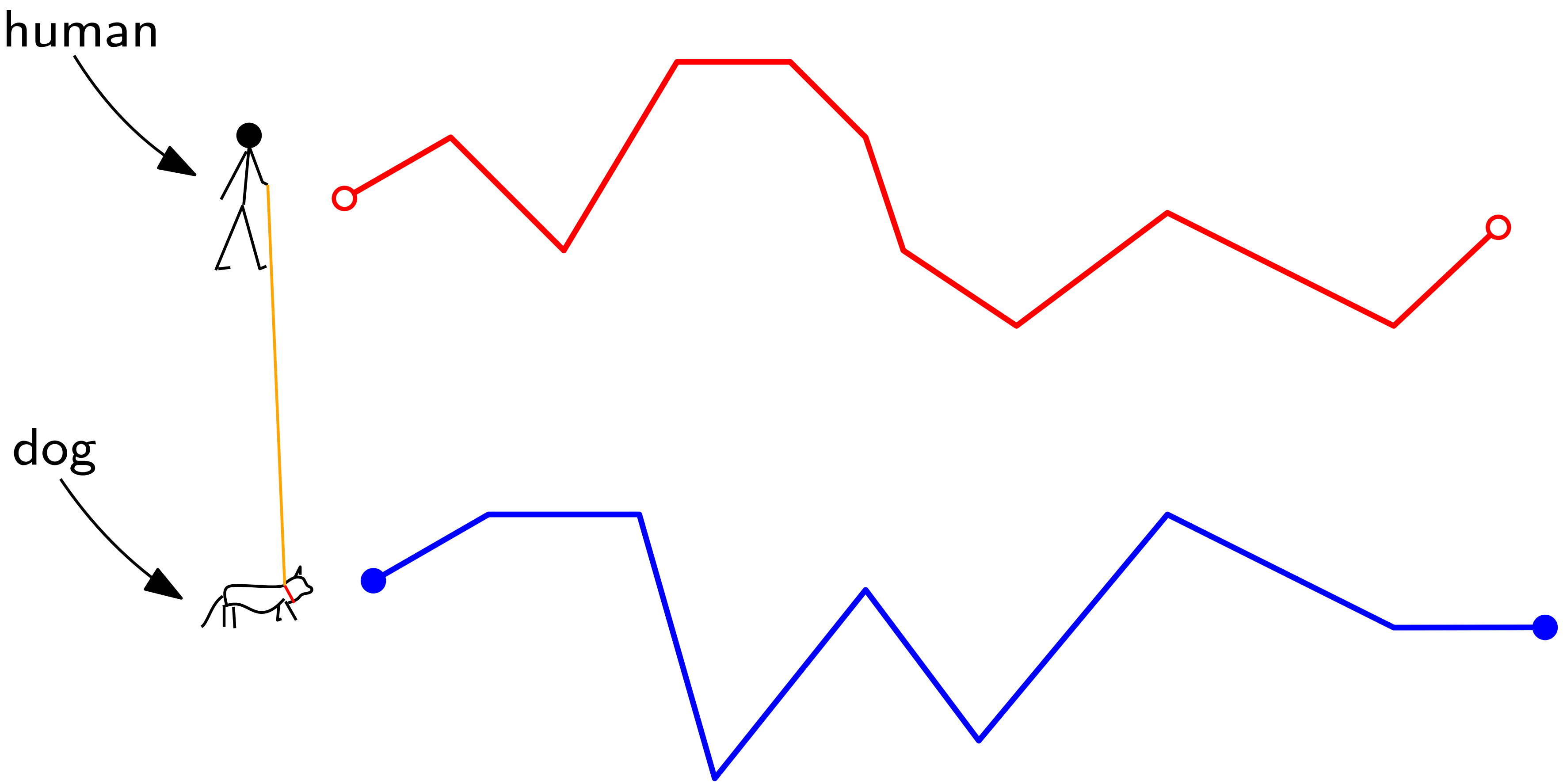


Handwritten Character Trajectories:



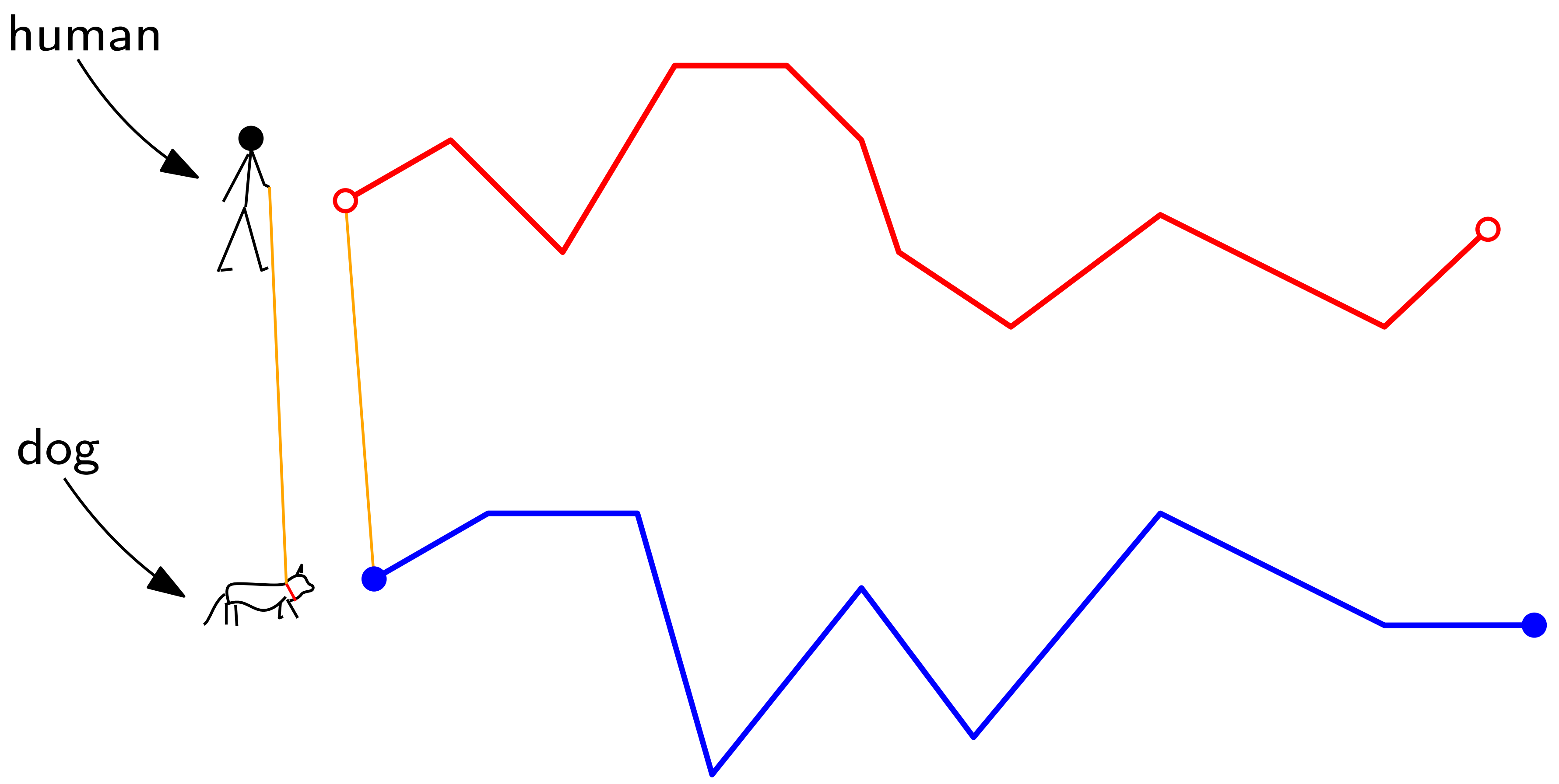
Discrete Fréchet Distance

Intuition



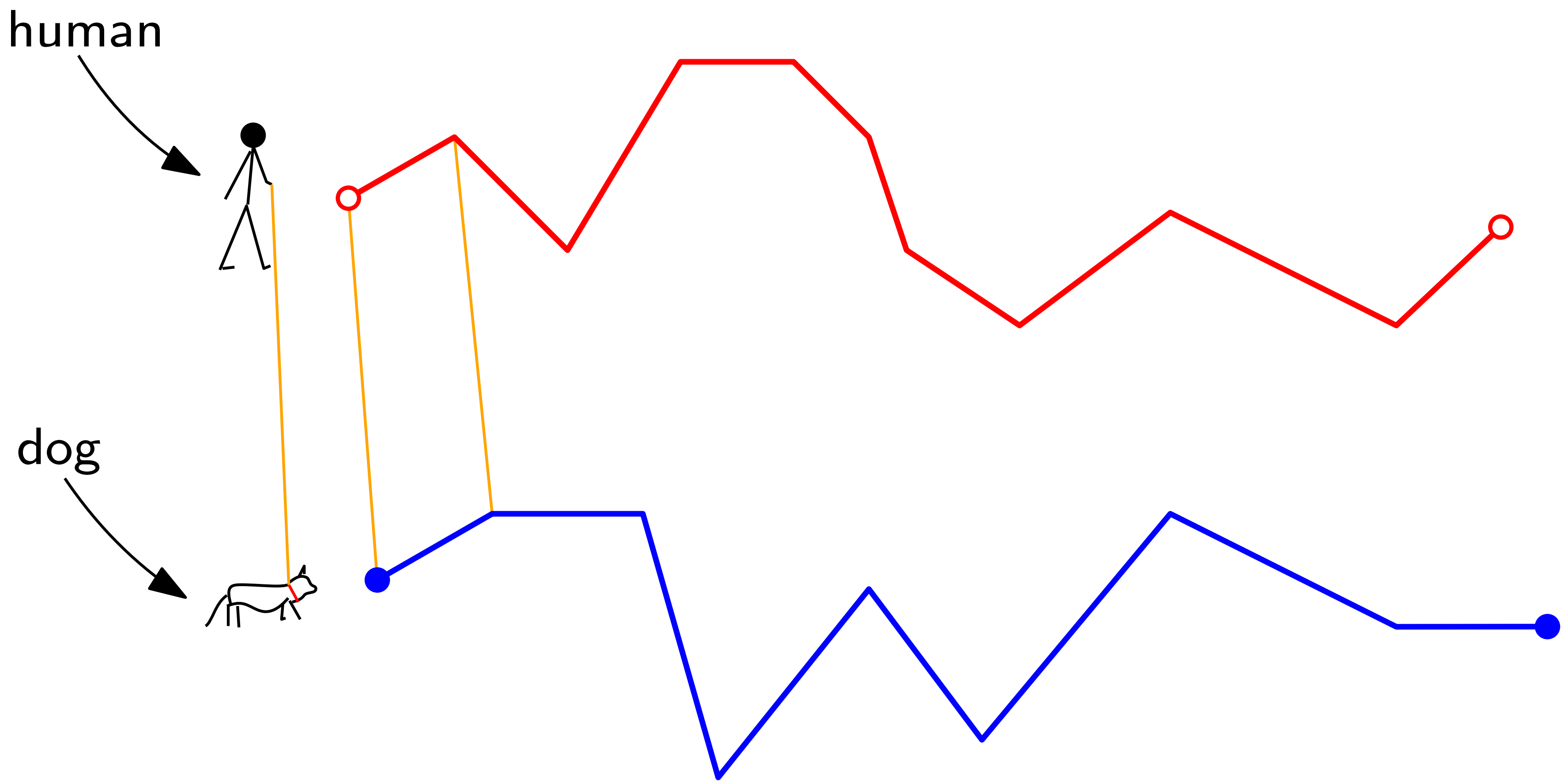
Discrete Fréchet Distance

Intuition



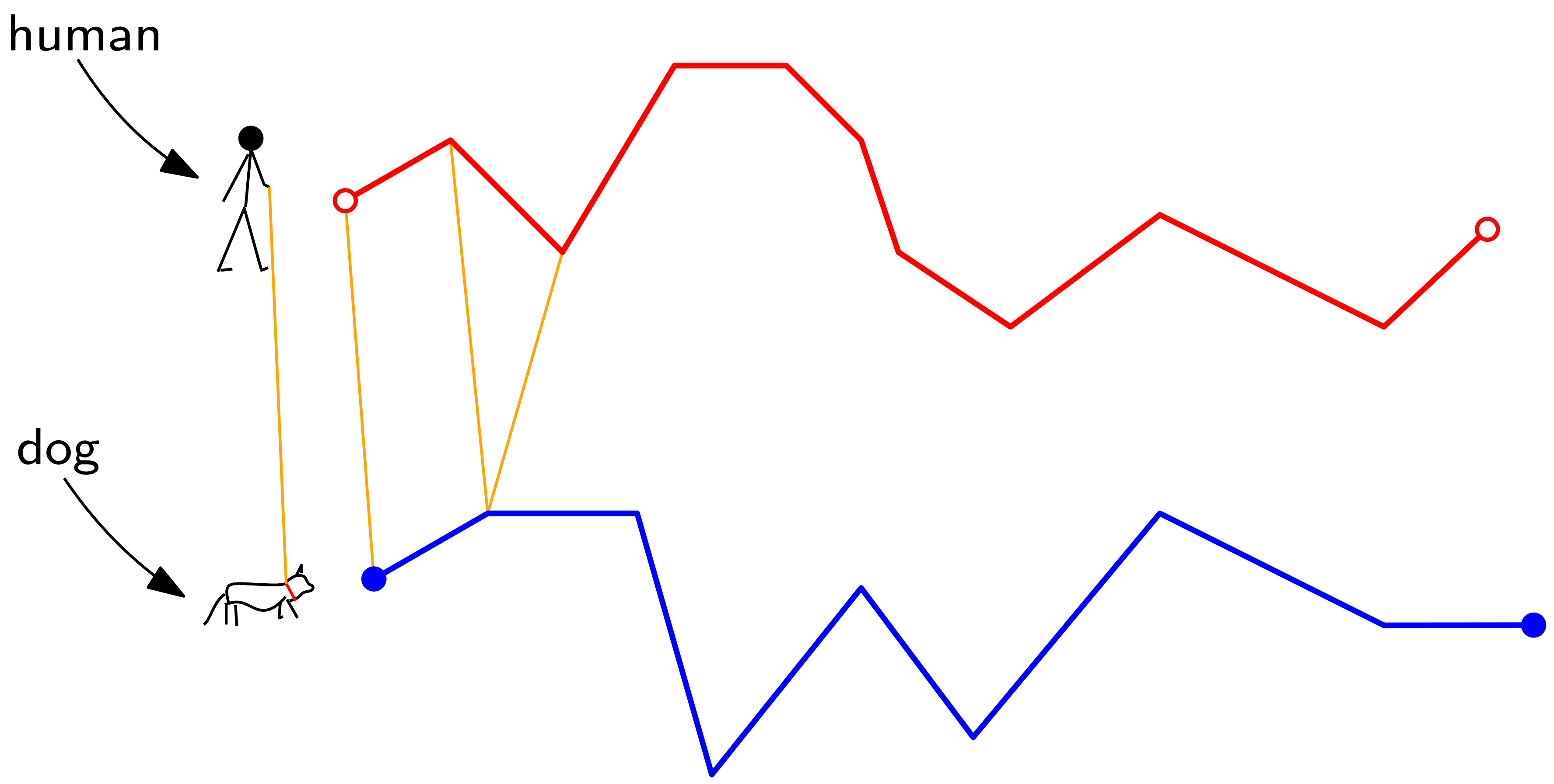
Discrete Fréchet Distance

Intuition



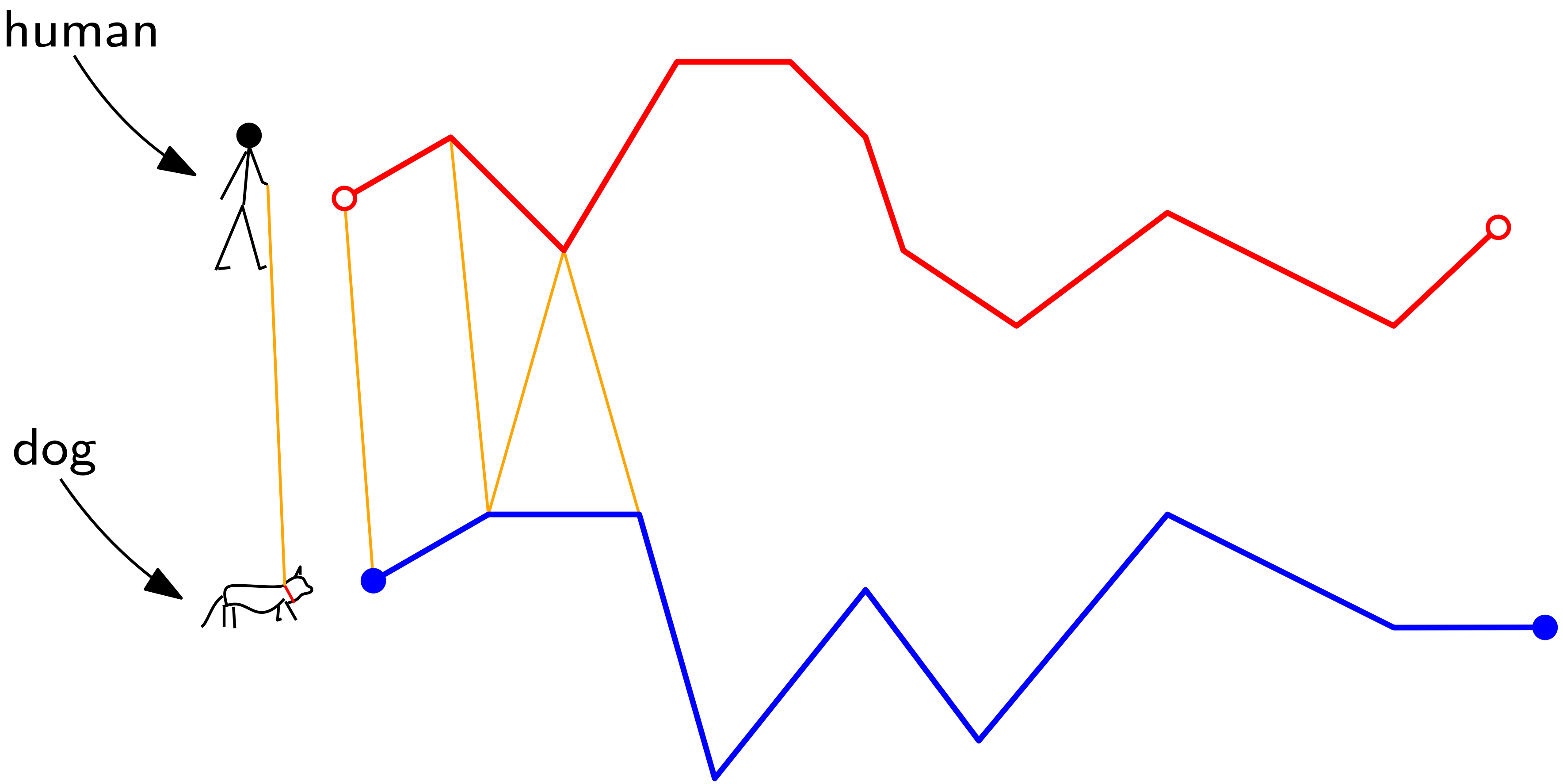
Discrete Fréchet Distance

Intuition



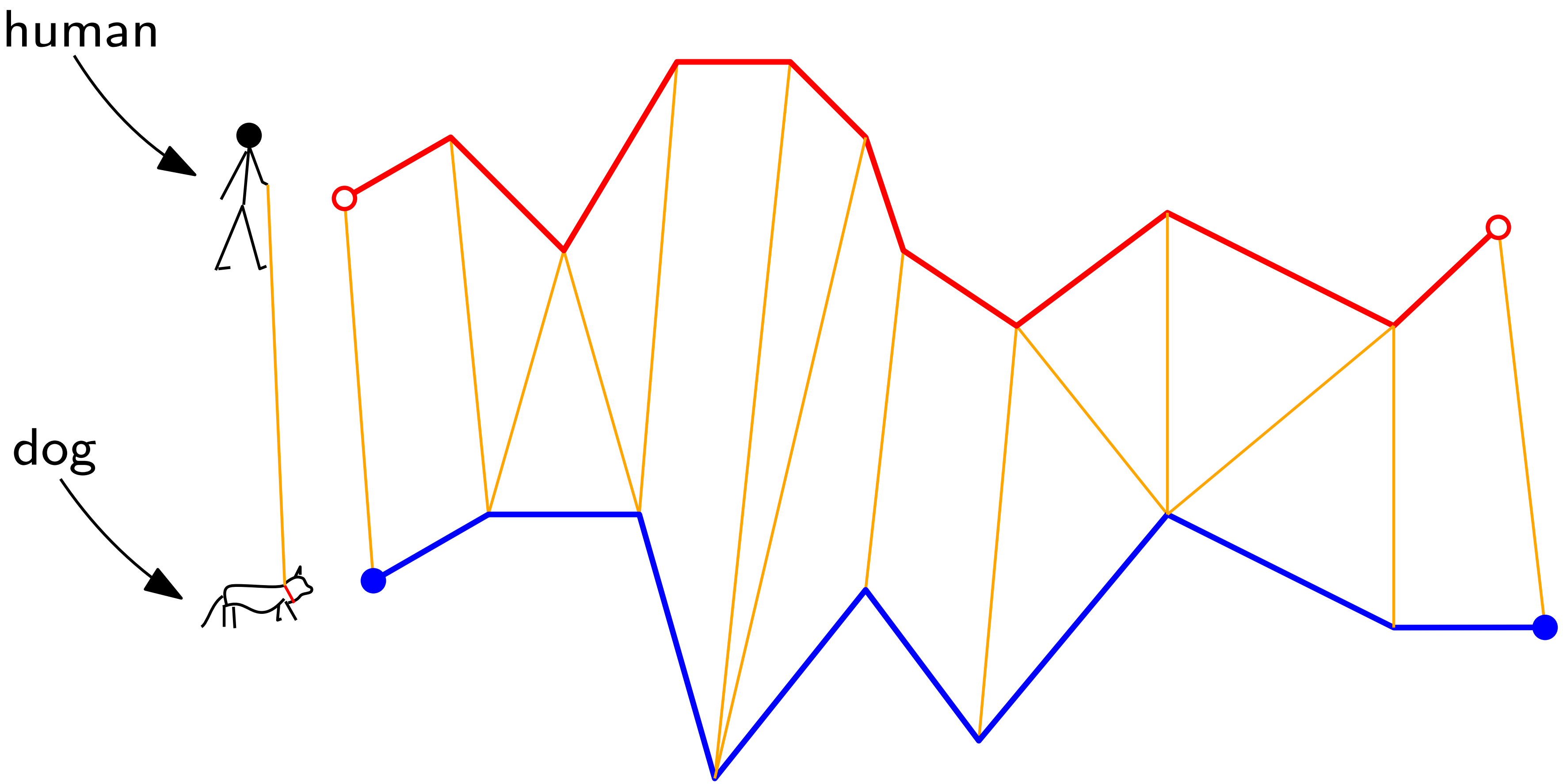
Discrete Fréchet Distance

Intuition



Discrete Fréchet Distance

Intuition



Question: What is the traversal that achieves the shortest leash length?

Discrete Fréchet Distance

Formal Definition

$$\delta_F(\pi, \sigma) := \min_{f,g \in \mathcal{T}} \max_{t \in [0,1]} \left\| \pi_{f(t)} - \sigma_{g(t)} \right\|$$

$\pi, \sigma =$ polygonal curves of length n
$\mathcal{T} =$ set of monotone and surjective functions from $[0,1]$ to $\{1, \ldots, n\}$

Discrete Fréchet Distance under Translation

Definition

Intuition: Allow arbitrary translations $\tau \in \mathbb{R}^2$ of curve σ.

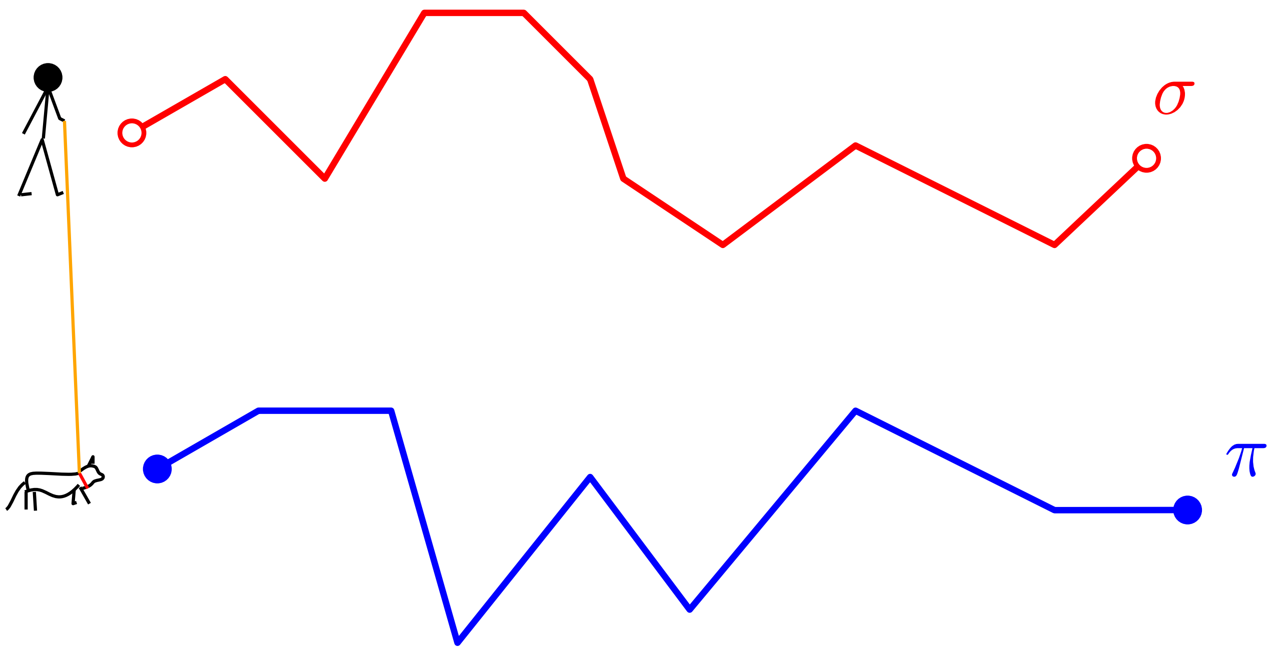
$$\delta_T(\pi, \sigma) := \min_{\tau \in \mathbb{R}^2} \delta_F(\pi, \sigma + \tau)$$

Discrete Fréchet Distance under Translation

Definition

Intuition: Allow arbitrary translations $\tau \in \mathbb{R}^2$ of curve σ.

$$\delta_T(\pi, \sigma) := \min_{\tau \in \mathbb{R}^2} \delta_F(\pi, \sigma + \tau)$$

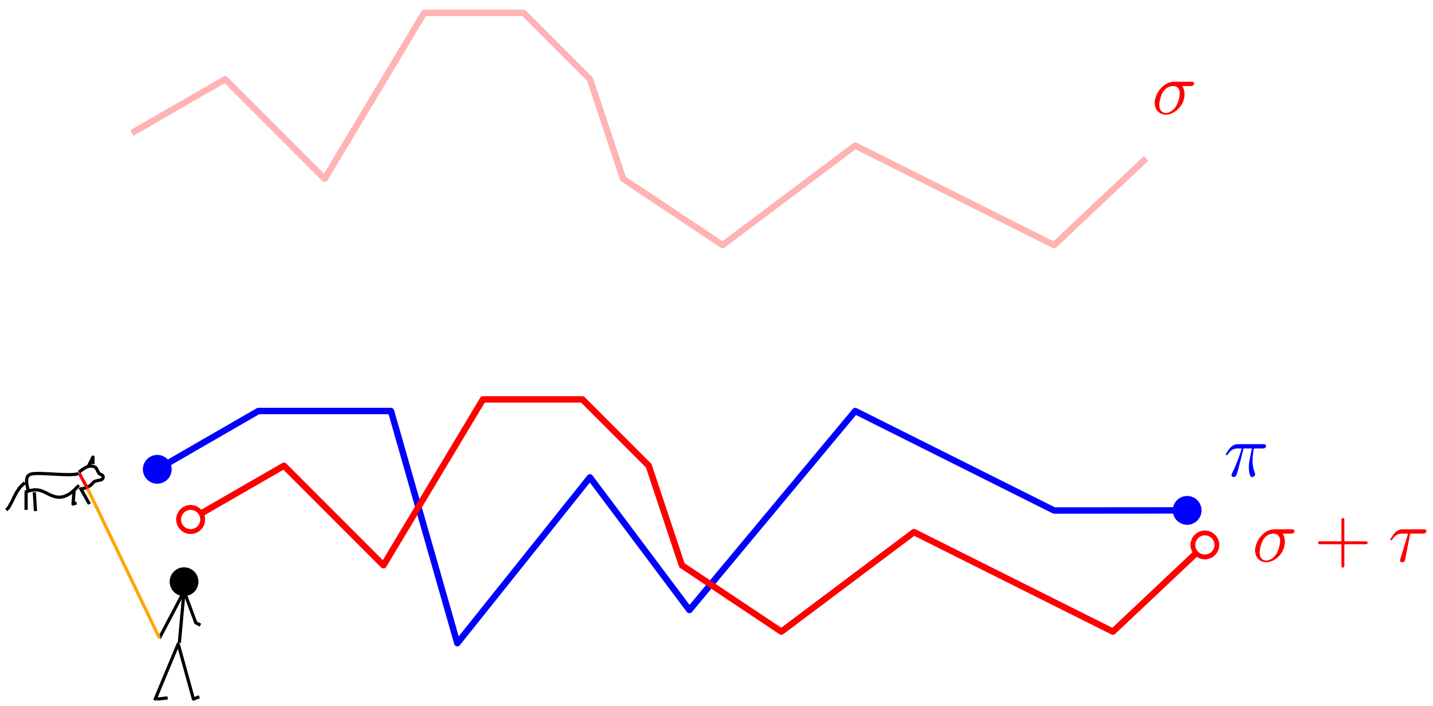


Discrete Fréchet Distance under Translation

Definition

Intuition: Allow arbitrary translations $\tau \in \mathbb{R}^2$ of curve σ.

$$\delta_T(\pi, \sigma) := \min_{\tau \in \mathbb{R}^2} \delta_F(\pi, \sigma + \tau)$$

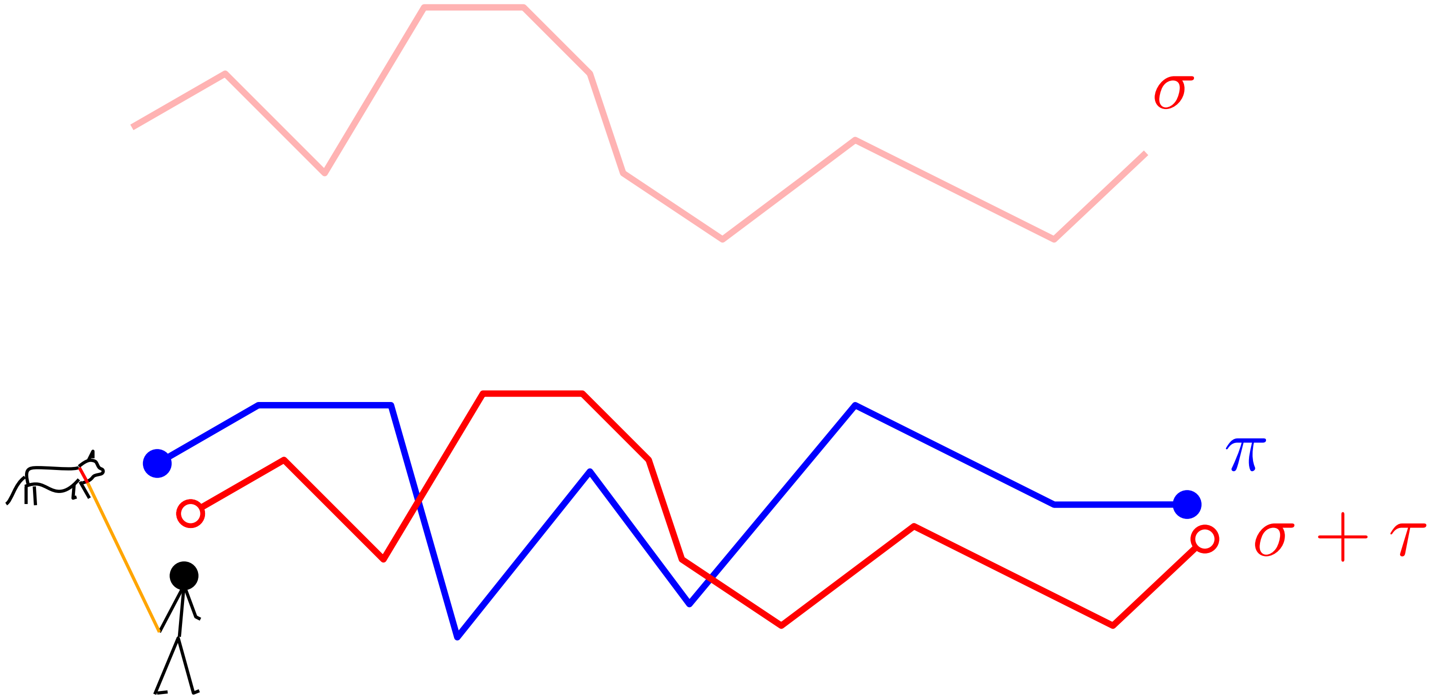


Discrete Fréchet Distance under Translation

Definition

Intuition: Allow arbitrary translations $\tau \in \mathbb{R}^2$ of curve σ.

$$\delta_T(\pi, \sigma) := \min_{\tau \in \mathbb{R}^2} \delta_F(\pi, \sigma + \tau)$$



Decision Problem:

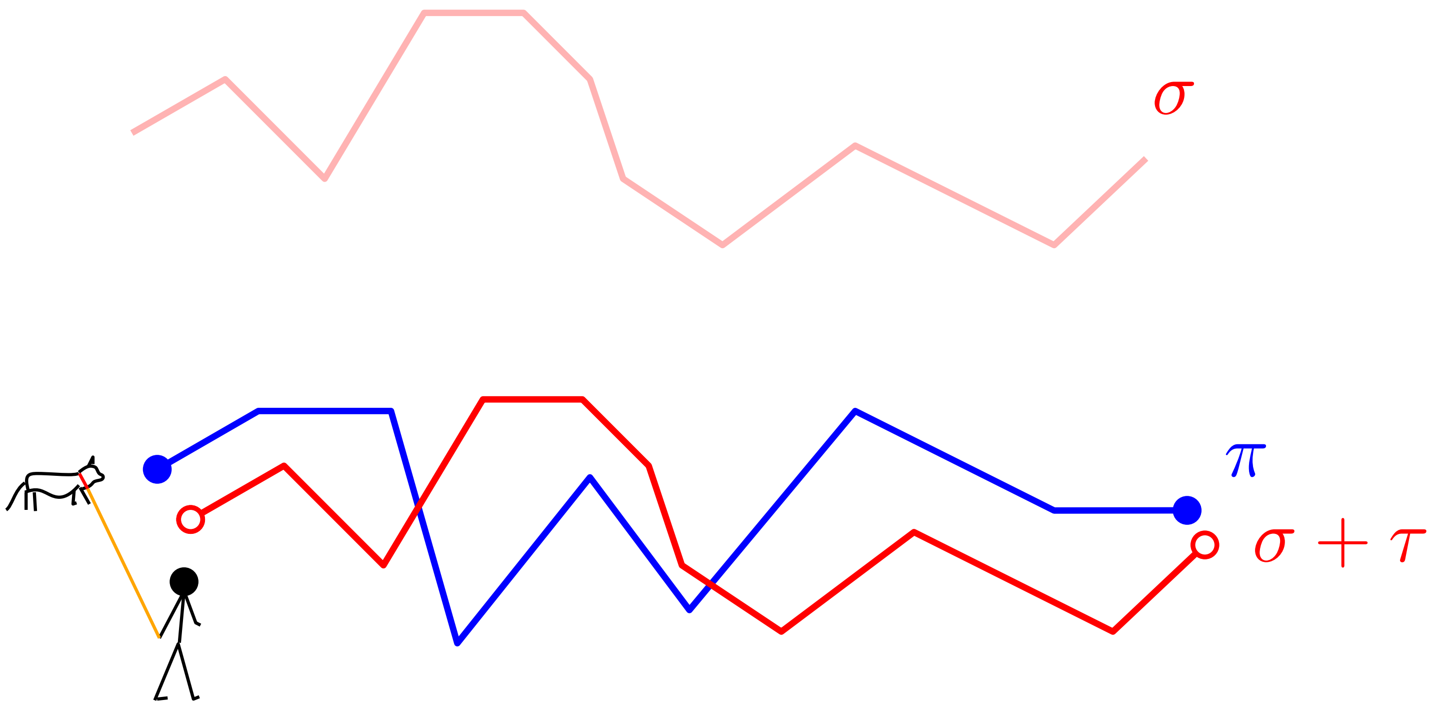
- Given π, σ, δ
- $\delta_T(\pi, \sigma) \leq \delta$?

Discrete Fréchet Distance under Translation

Definition

Intuition: Allow arbitrary translations $\tau \in \mathbb{R}^2$ of curve σ.

$$\delta_T(\pi, \sigma) := \min_{\tau \in \mathbb{R}^2} \delta_F(\pi, \sigma + \tau)$$



Decision Problem:

- Given π, σ, δ
- $\delta_T(\pi, \sigma) \le \delta$?

Focus on this in the talk!

Goal:

Performant implementation
computing the discrete Fréchet distance under translation
on practical inputs.

Related Work

Theory:

curve length

- Discrete Fréchet distance under translation in $\tilde{\mathcal{O}}(n^5)$
  [Agarwal, Ben Avraham, Kaplan, Sharir arXiv'15]
- Discrete Fréchet distance under translation in $\tilde{\mathcal{O}}(n^{4.66})$
  [Bringmann, Künnemann, N. SODA'19]
- SETH based lower bound of $n^{4-o(1)}$ for discrete Fréchet distance under translation
  [Bringmann, Künnemann, N. SODA'19]

Related Work

Theory:

curve length

- Discrete Fréchet distance under translation in $\tilde{\mathcal{O}}(n^5)$
  [Agarwal, Ben Avraham, Kaplan, Sharir arXiv'15]
- Discrete Fréchet distance under translation in $\tilde{\mathcal{O}}(n^{4.66})$
  [Bringmann, Künnemann, N. SODA'19]
- SETH based lower bound of $n^{4-o(1)}$ for discrete Fréchet distance under translation
  [Bringmann, Künnemann, N. SODA'19]

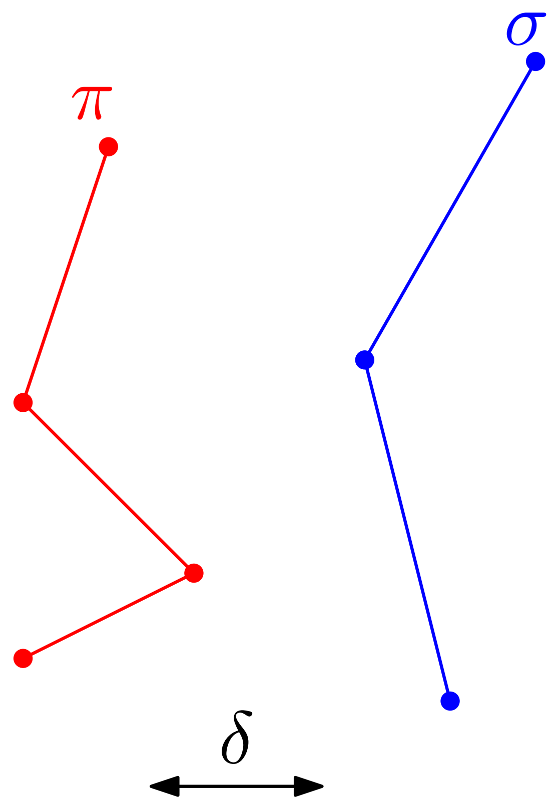
Practice:

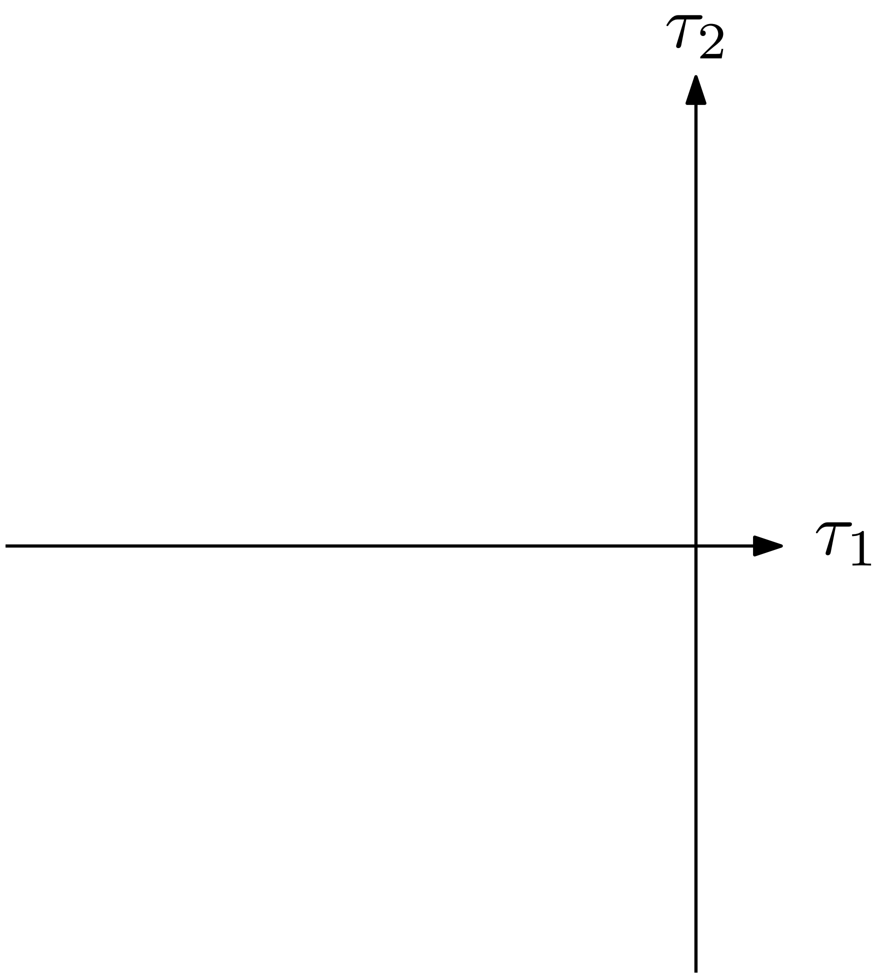
- GIS Cup on (fixed-translation) Fréchet distance near neighbors search
  [Werner, Oliver; Baldus et al.; Buchin et al.; Dütsch et al. SIGSPATIAL'17]
- State of the art (fixed-translation) Fréchet distance implementation
  [Bringmann, Künnemann, N. SoCG'19]

Approach I: Discrete Algorithms

Arrangement

- Idea: Partition the plane into equivalent regions.

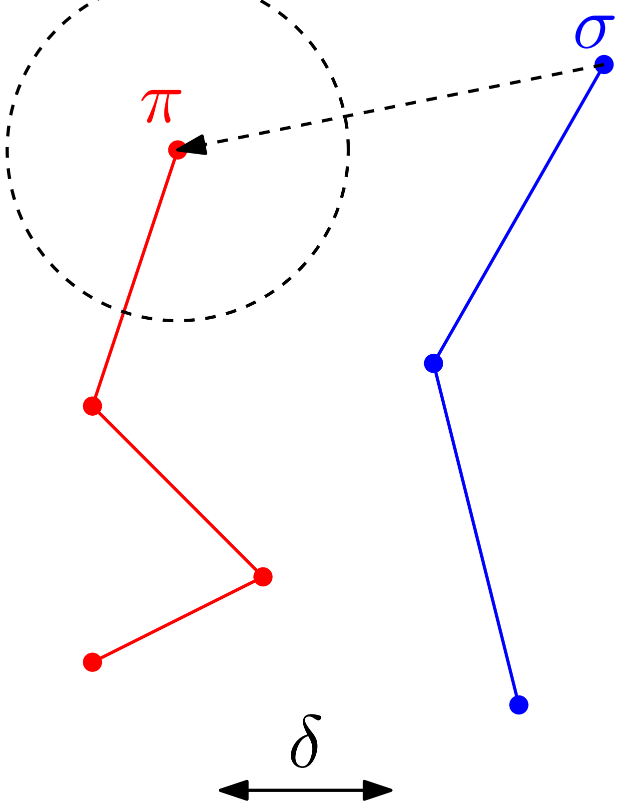


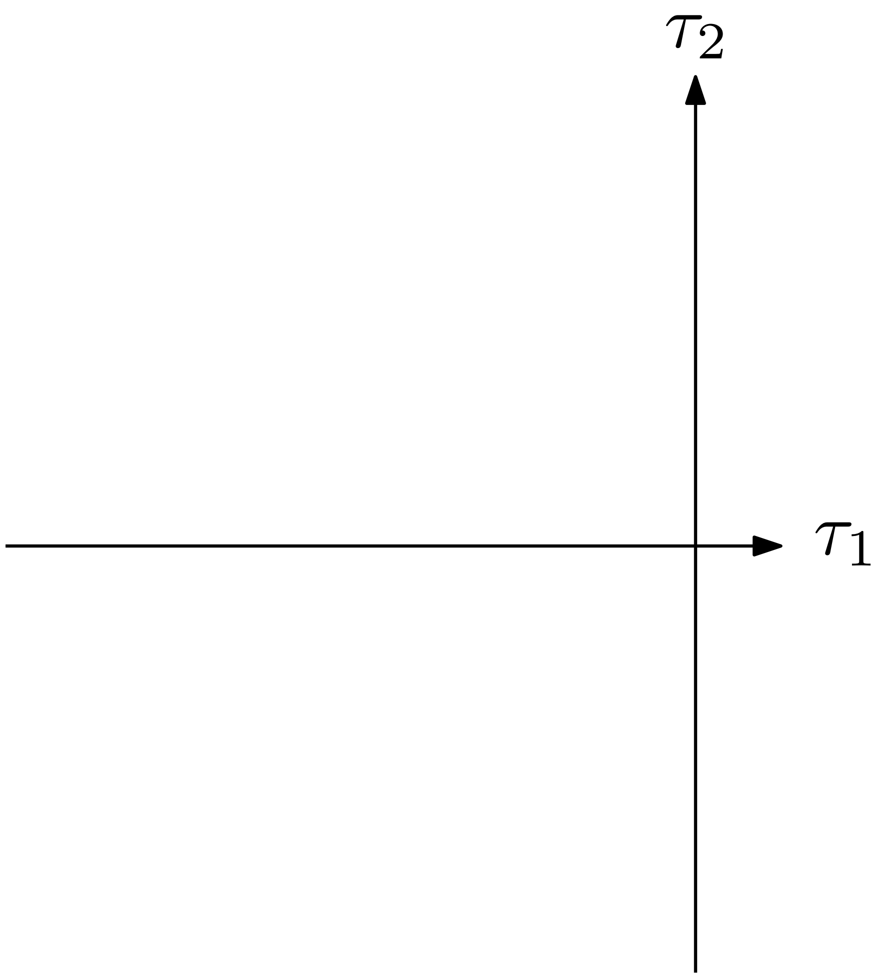


Approach I: Discrete Algorithms

Arrangement

- Idea: Partition the plane into equivalent regions.

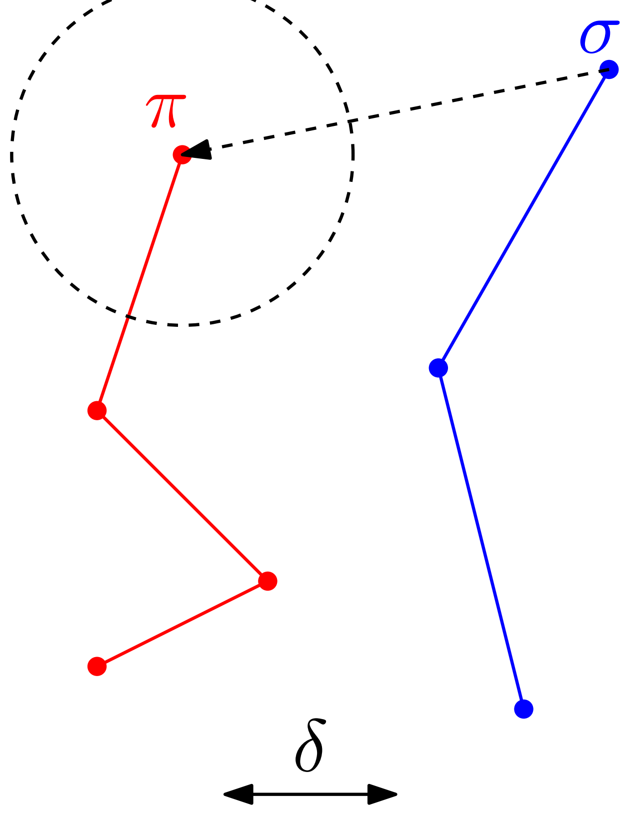


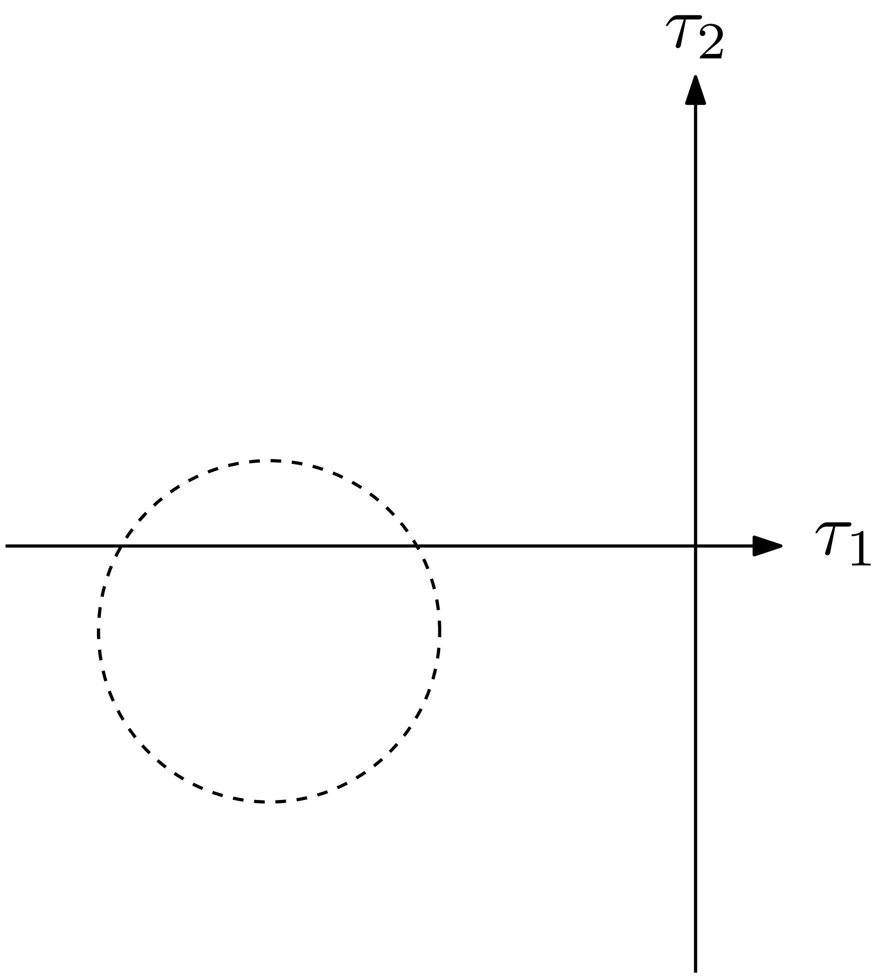


Approach I: Discrete Algorithms

Arrangement

- Idea: Partition the plane into equivalent regions.

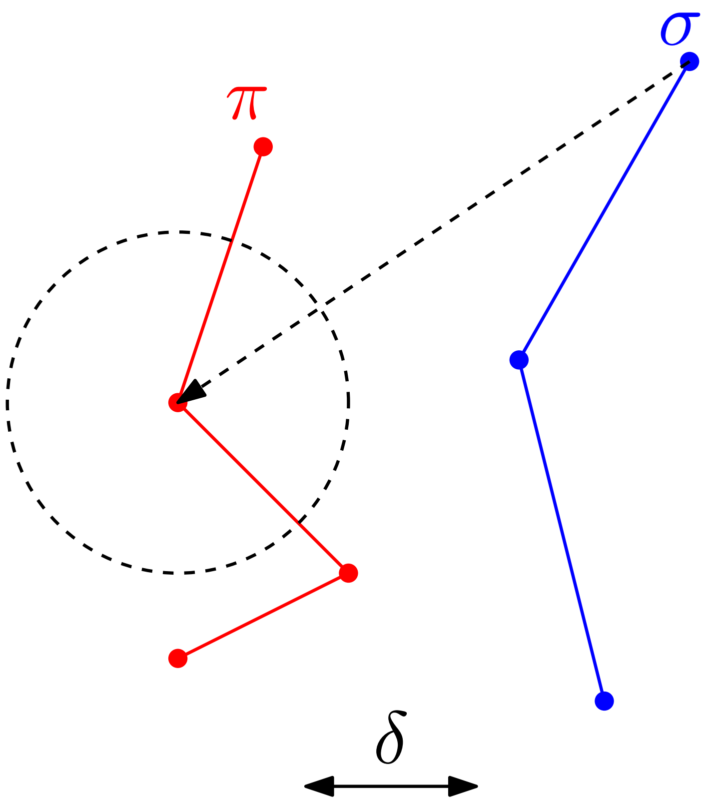


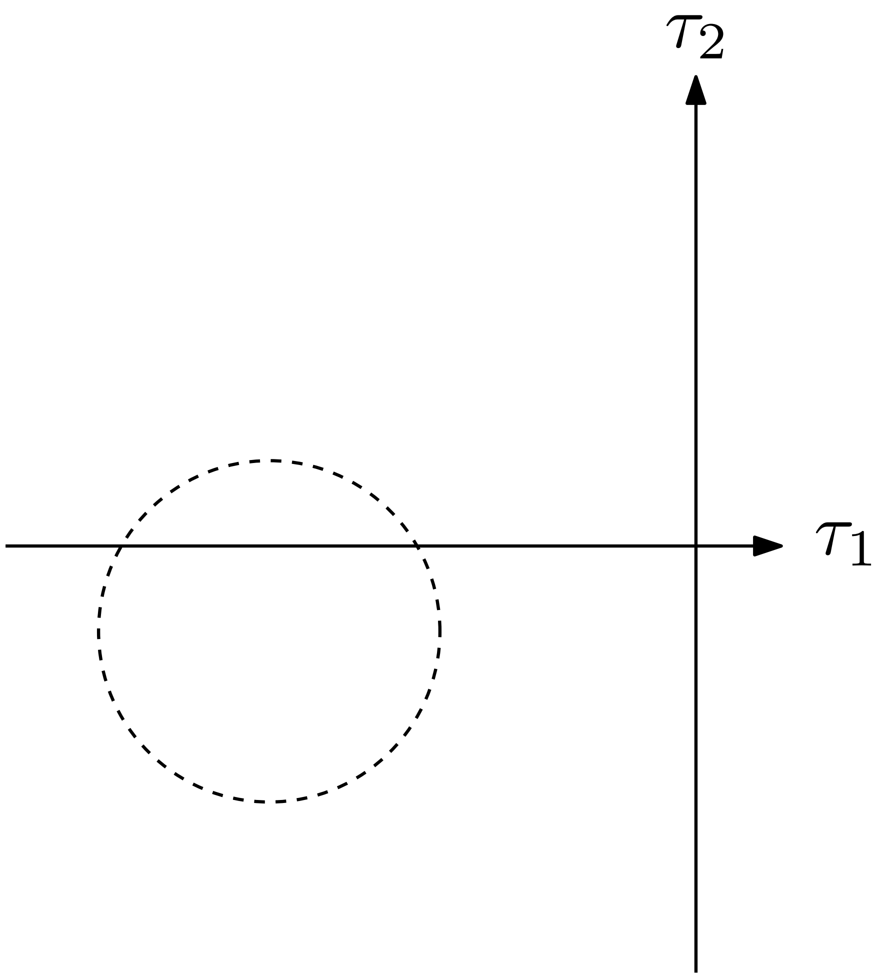


Approach I: Discrete Algorithms

Arrangement

- Idea: Partition the plane into equivalent regions.

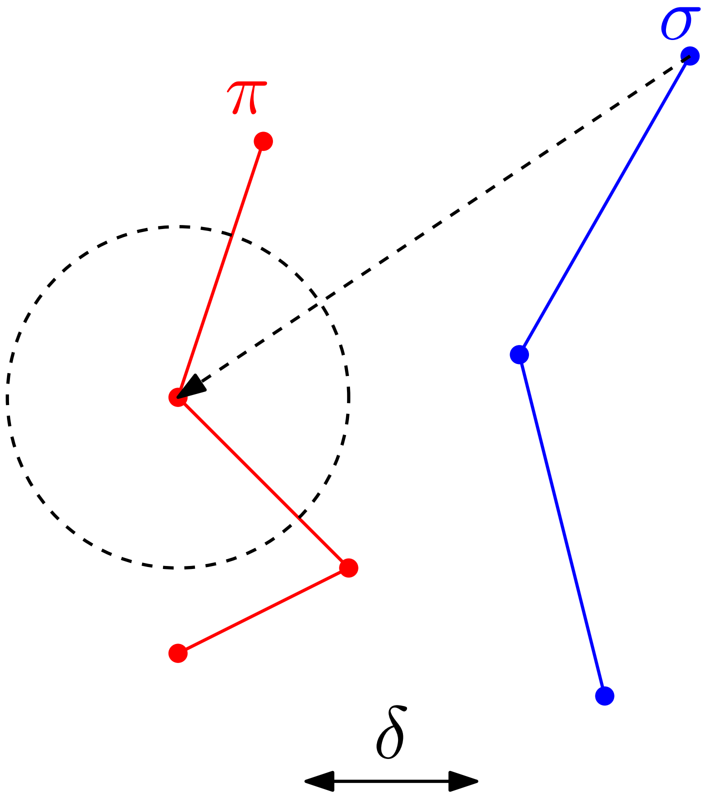


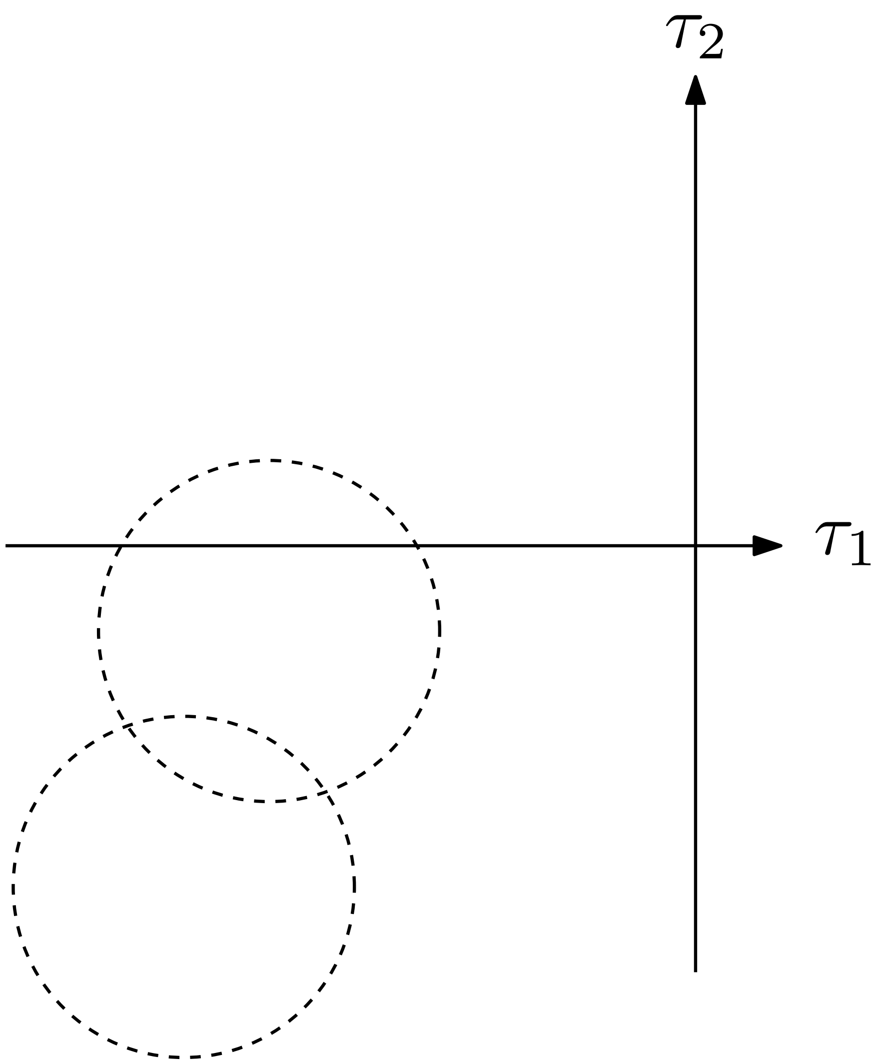


Approach I: Discrete Algorithms

Arrangement

- Idea: Partition the plane into equivalent regions.

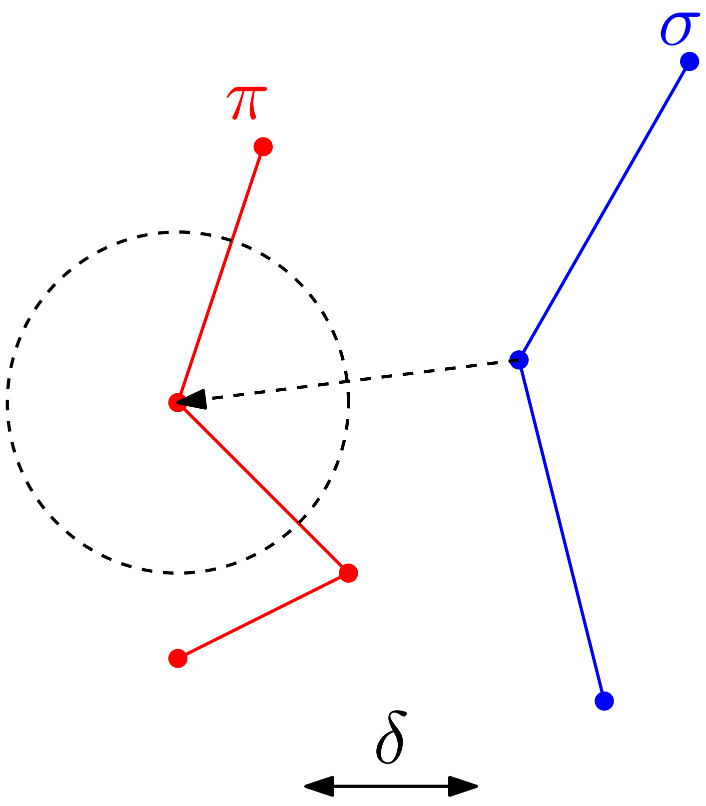


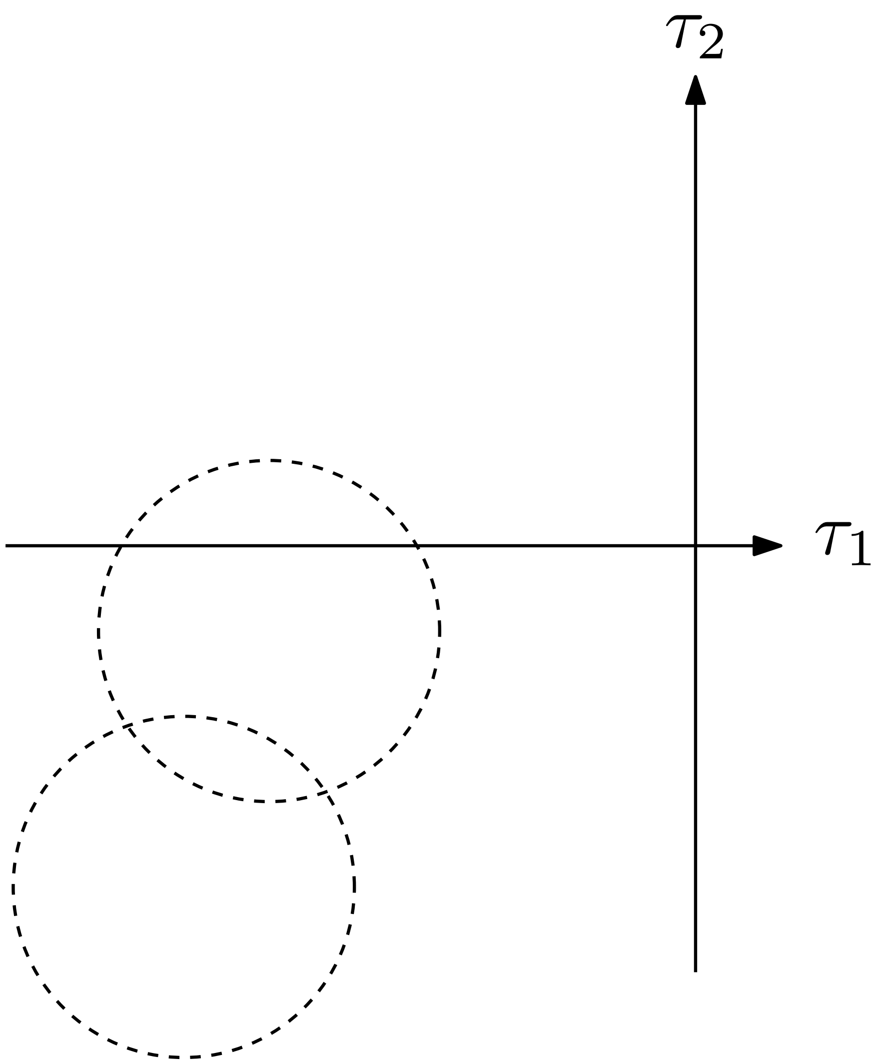


Approach I: Discrete Algorithms

Arrangement

- Idea: Partition the plane into equivalent regions.

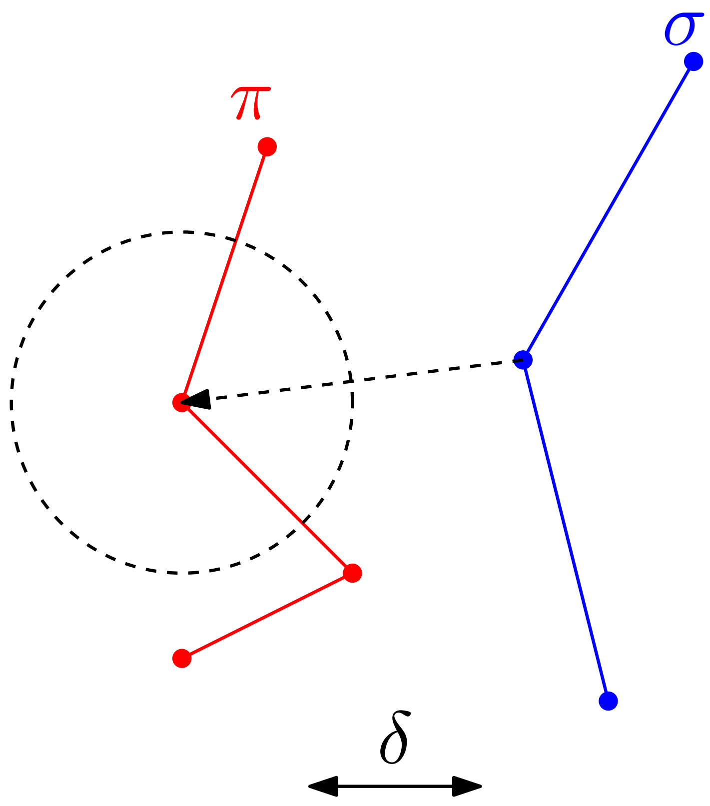


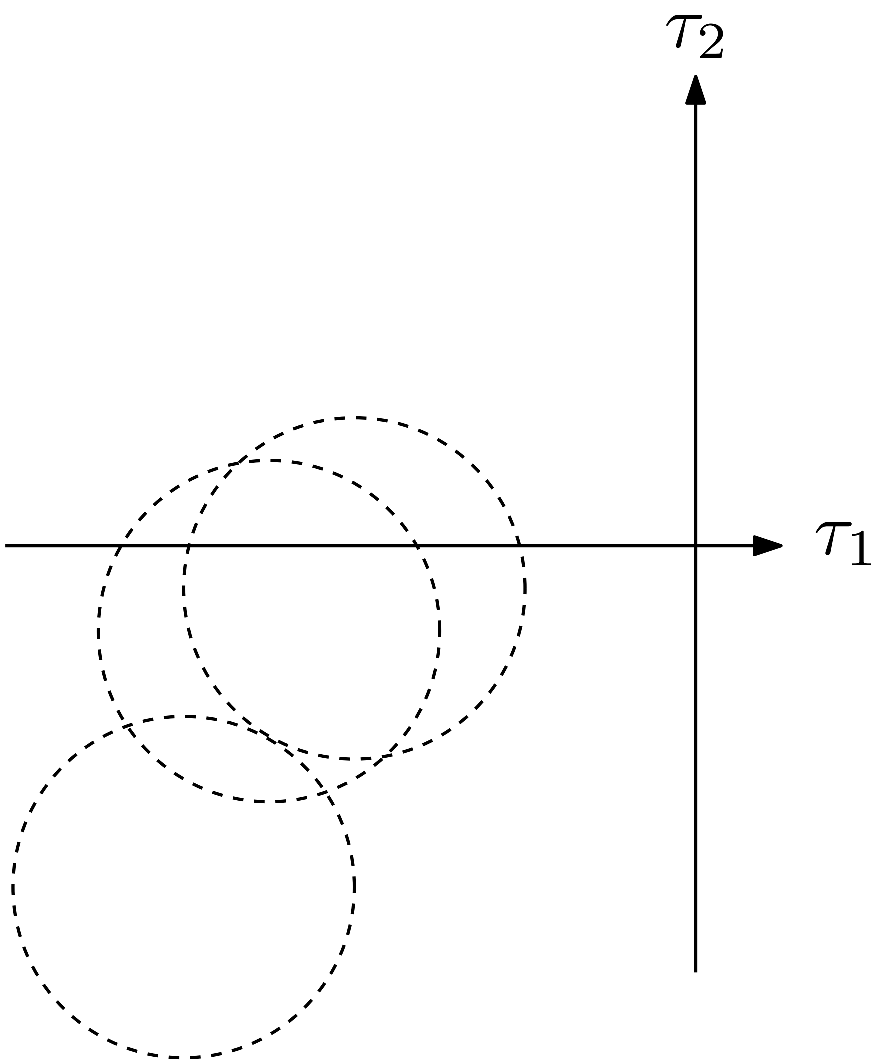


Approach I: Discrete Algorithms

Arrangement

- Idea: Partition the plane into equivalent regions.

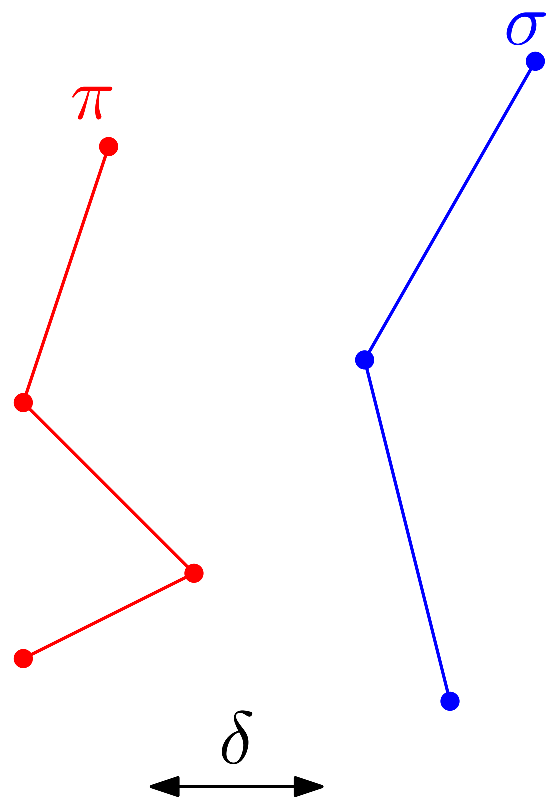


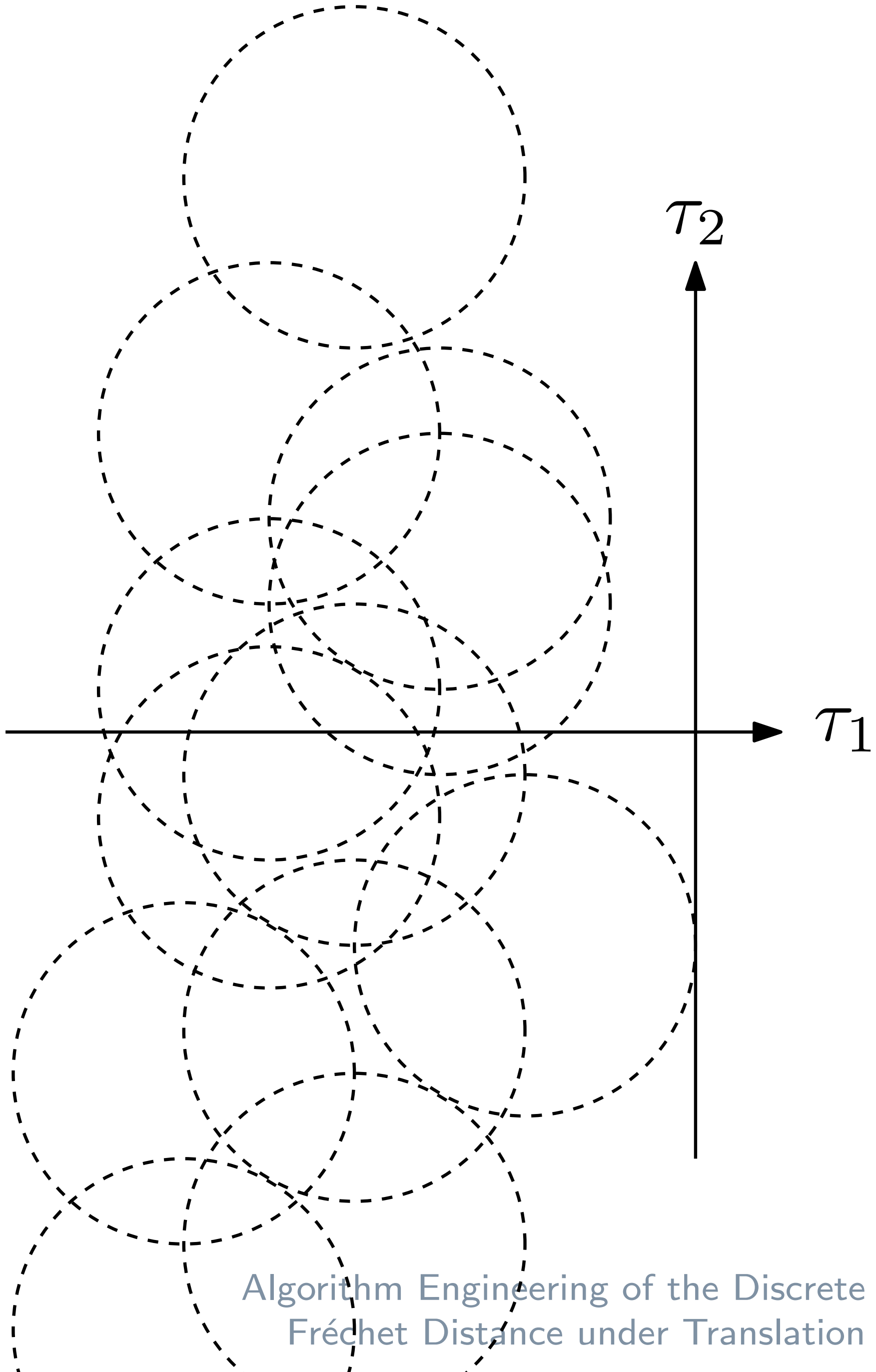


Approach I: Discrete Algorithms

Arrangement

- Idea: Partition the plane into equivalent regions.

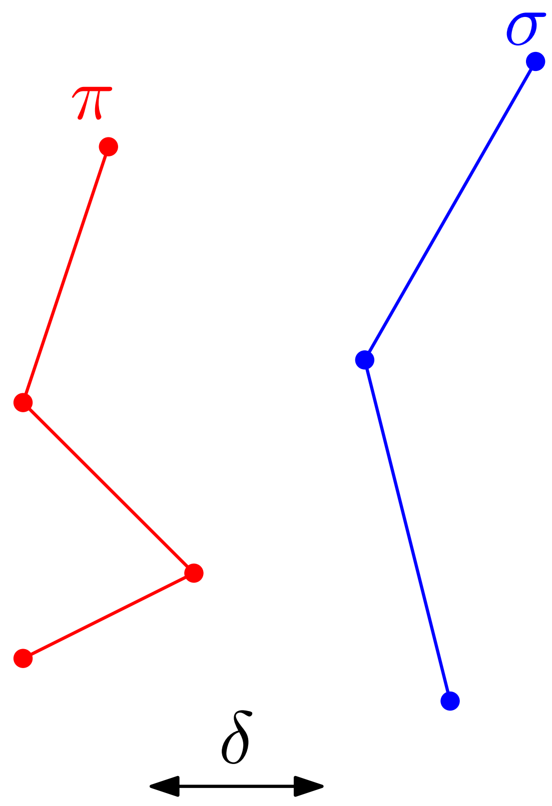




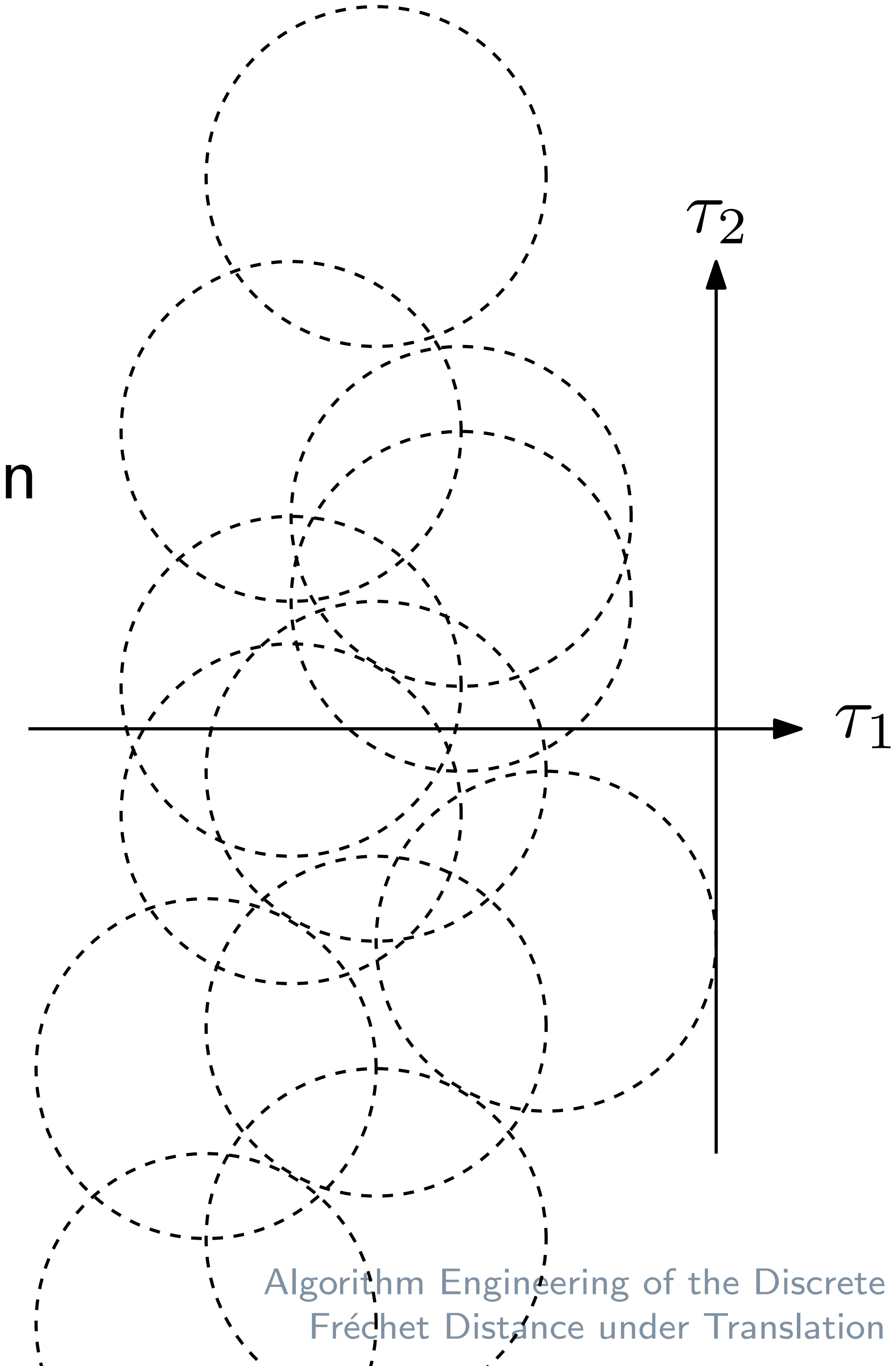
Approach I: Discrete Algorithms

Arrangement

- Idea: Partition the plane into equivalent regions.



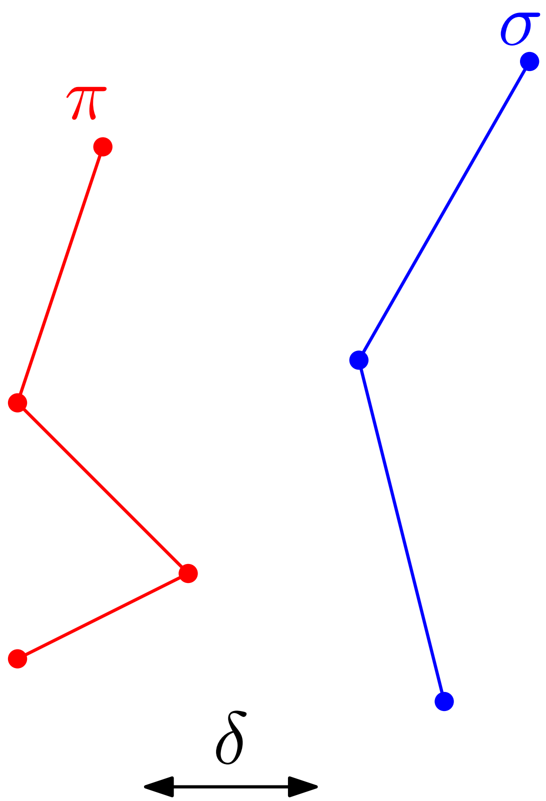
Observation: All translations in a cell of the arrangement have the same closeness relation.



Approach I: Discrete Algorithms

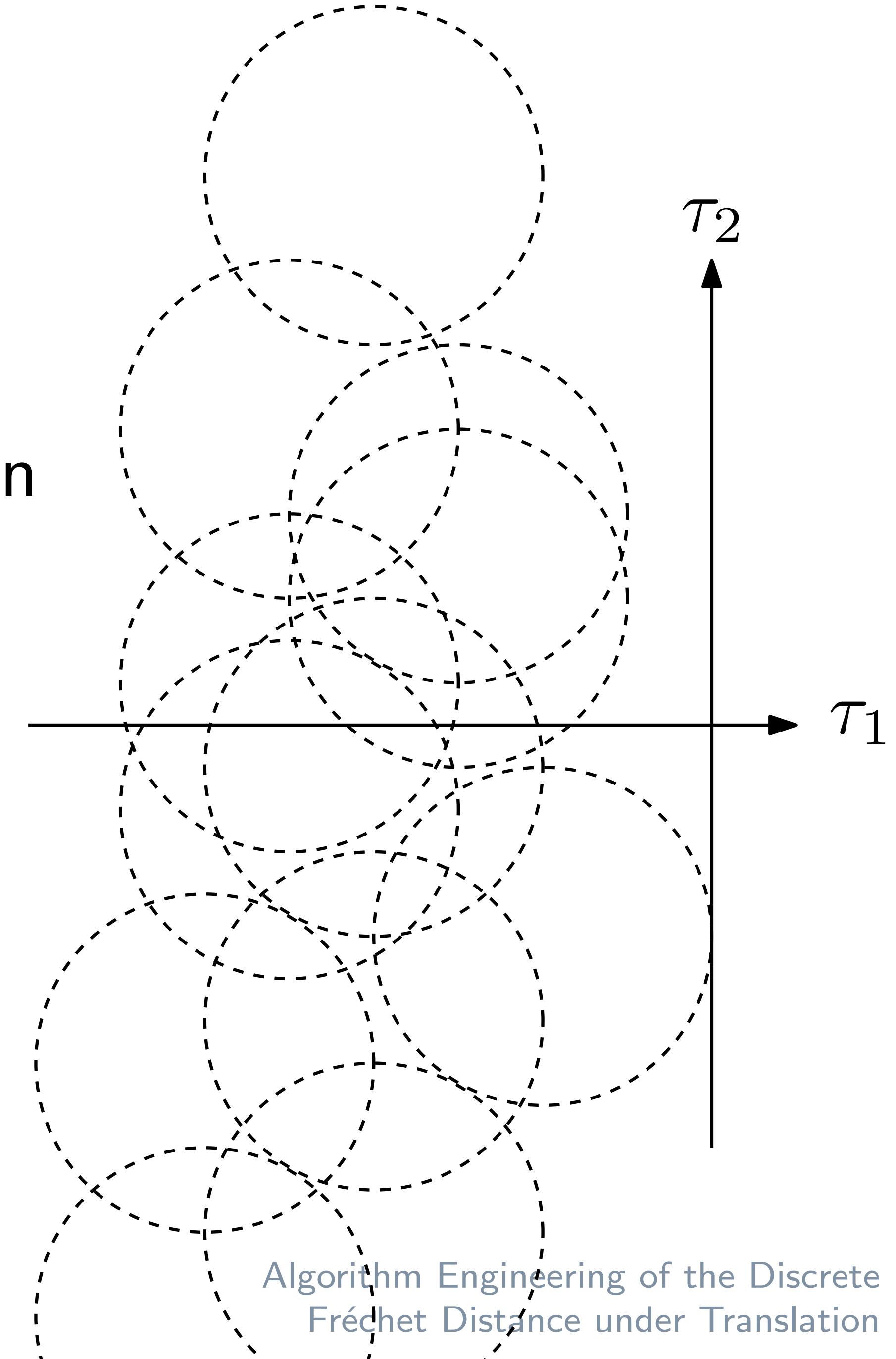
Arrangement

- Idea: Partition the plane into equivalent regions.



Observation: All translations in a cell of the arrangement have the same closeness relation.

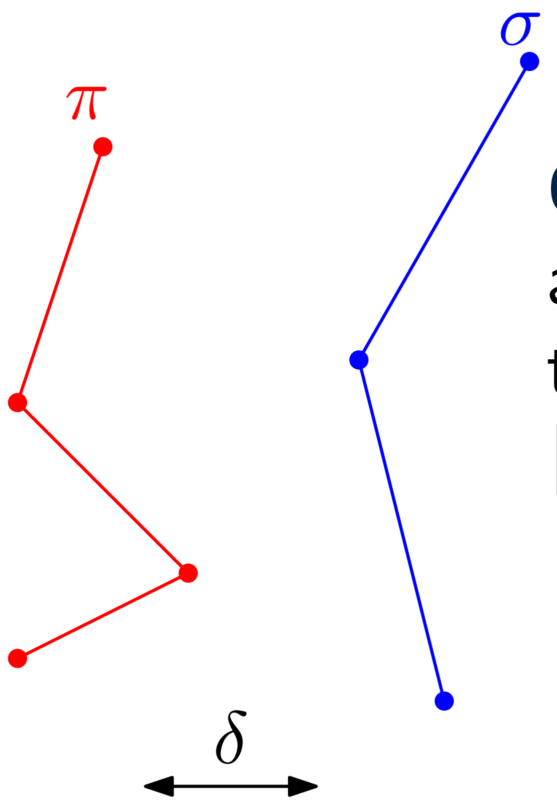
→ for each cell, pick some τ and check $d_F(\pi, \sigma + \tau)$



Approach I: Discrete Algorithms

Arrangement

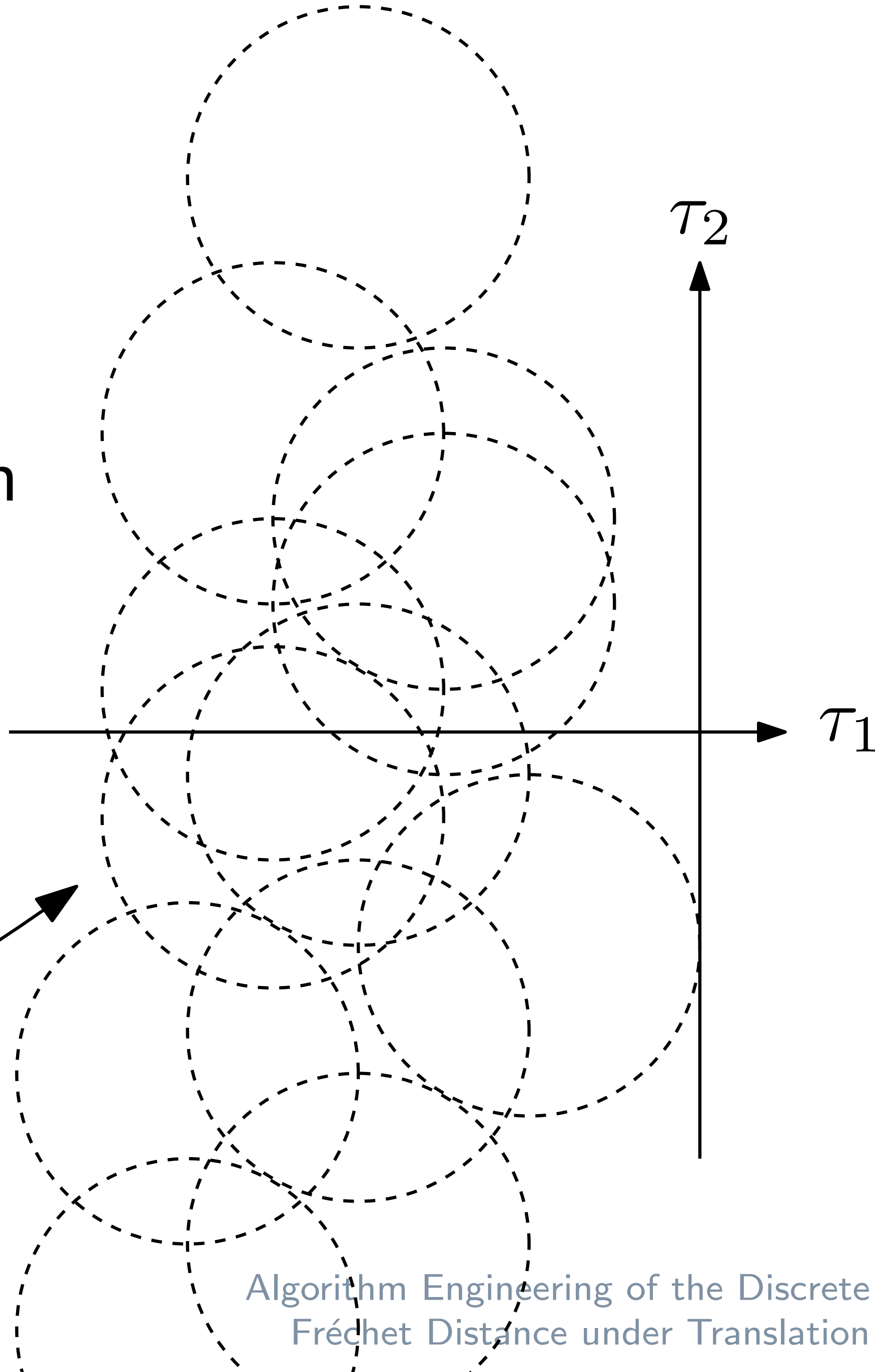
- Idea: Partition the plane into equivalent regions.



Observation: All translations in a cell of the arrangement have the same closeness relation.

$\rightarrow$ for each cell, pick some τ and check $d_F(\pi, \sigma + \tau)$

$\mathcal{O}(n^4)$ complexity

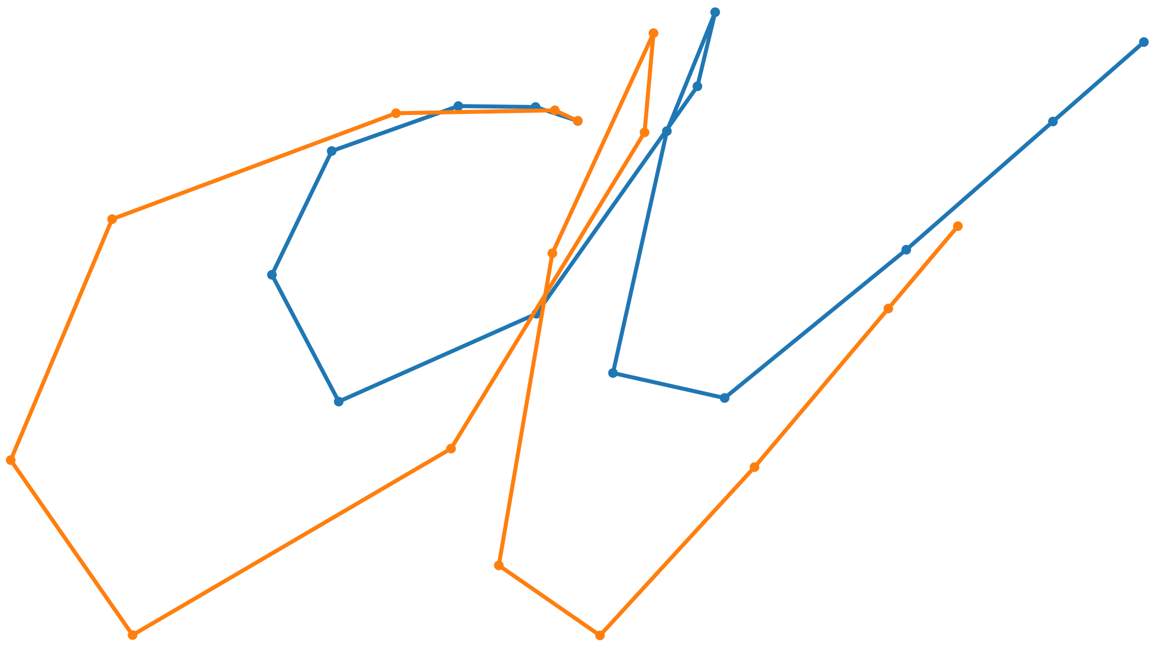


Approach I: Discrete Algorithms

- All known algorithms build an $\mathcal{O}(n^4)$ arrangement.

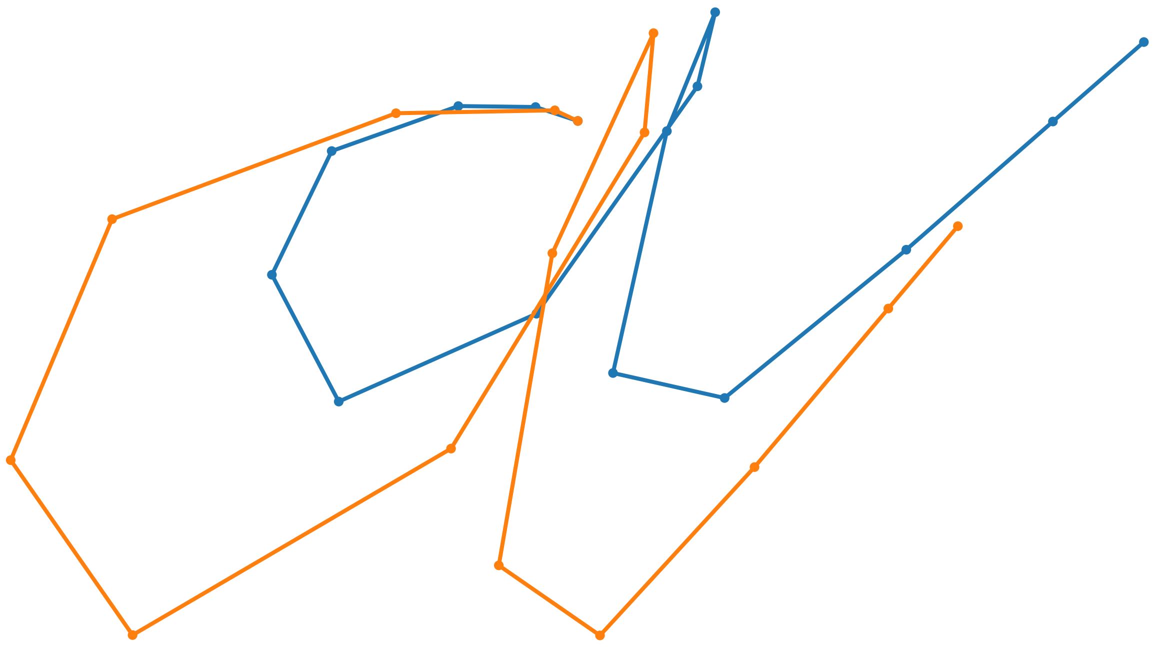
Approach I: Discrete Algorithms

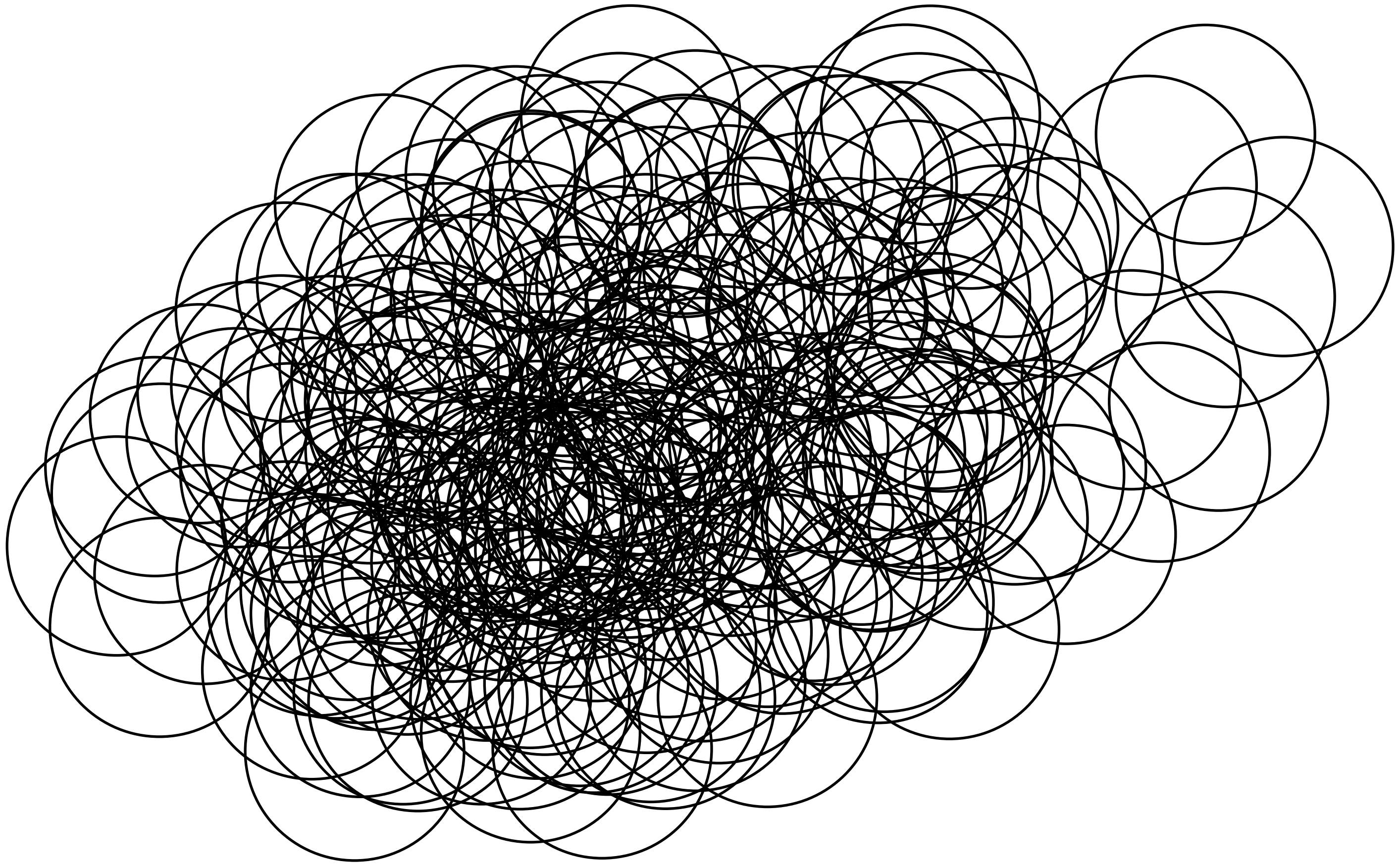
- All known algorithms build an $\mathcal{O}(n^4)$ arrangement.



Approach I: Discrete Algorithms

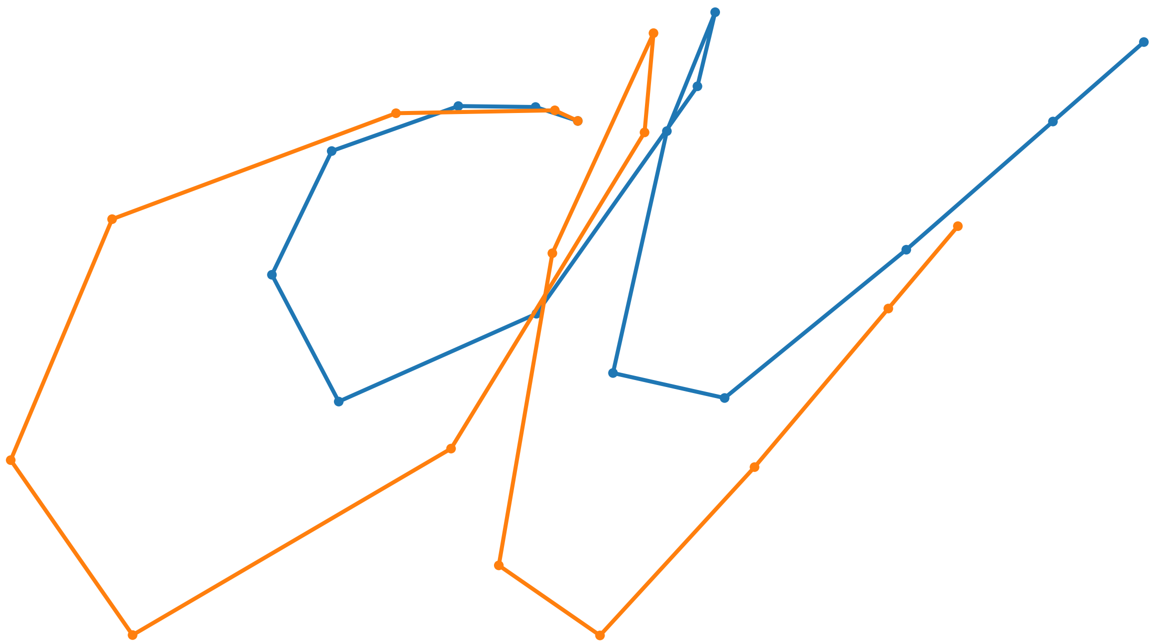
- All known algorithms build an $\mathcal{O}(n^4)$ arrangement.

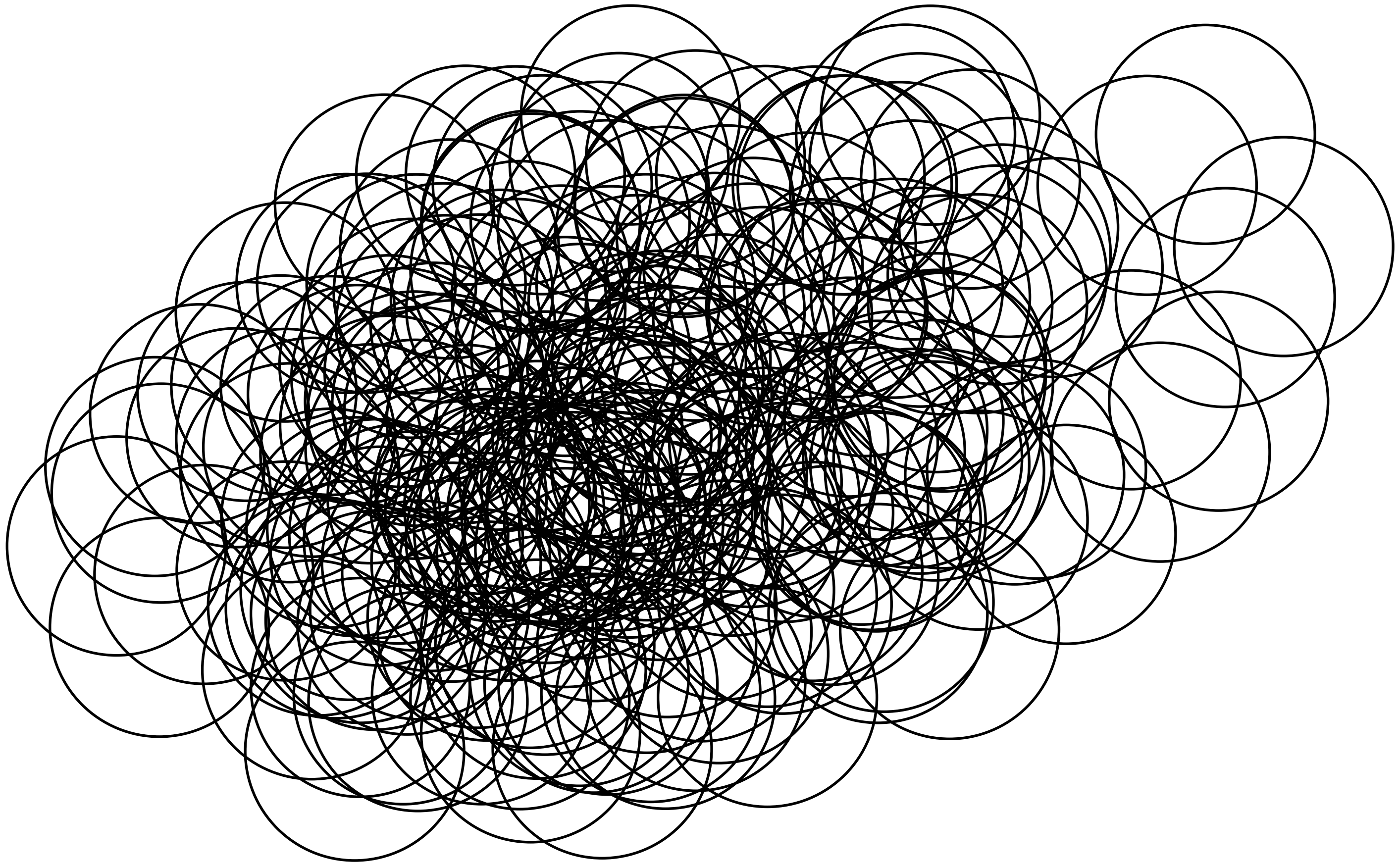




Approach I: Discrete Algorithms

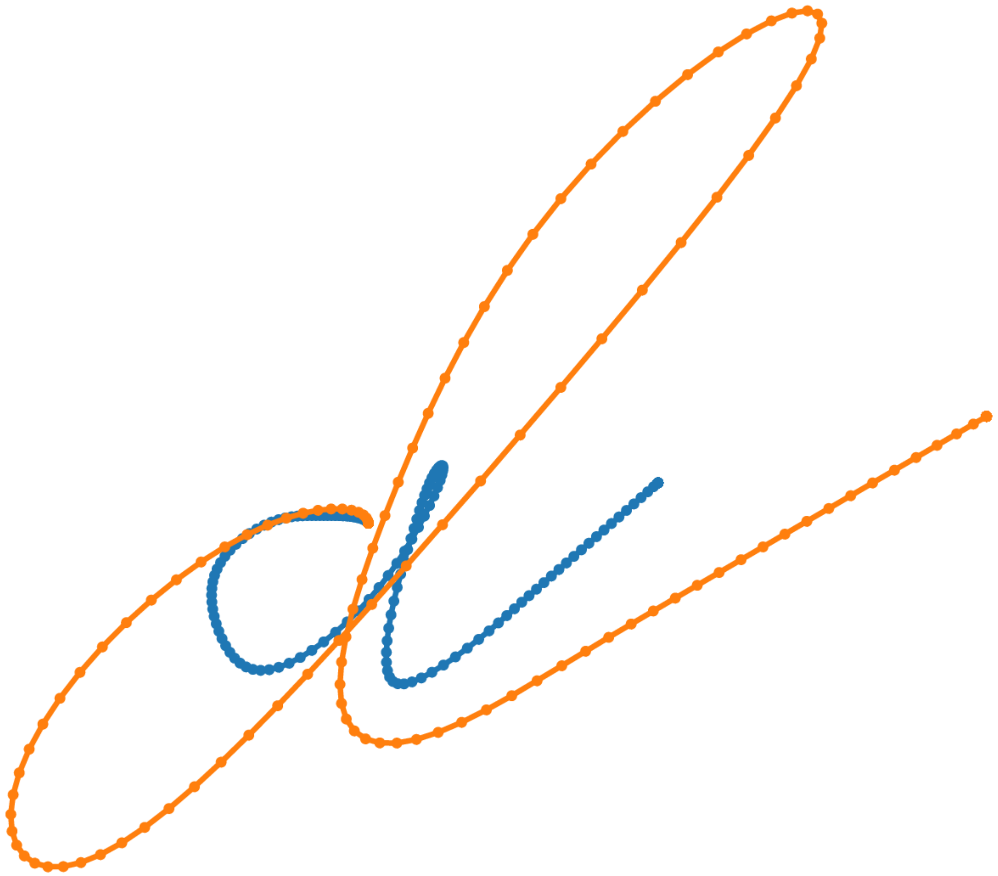
- All known algorithms build an $\mathcal{O}(n^4)$ arrangement.



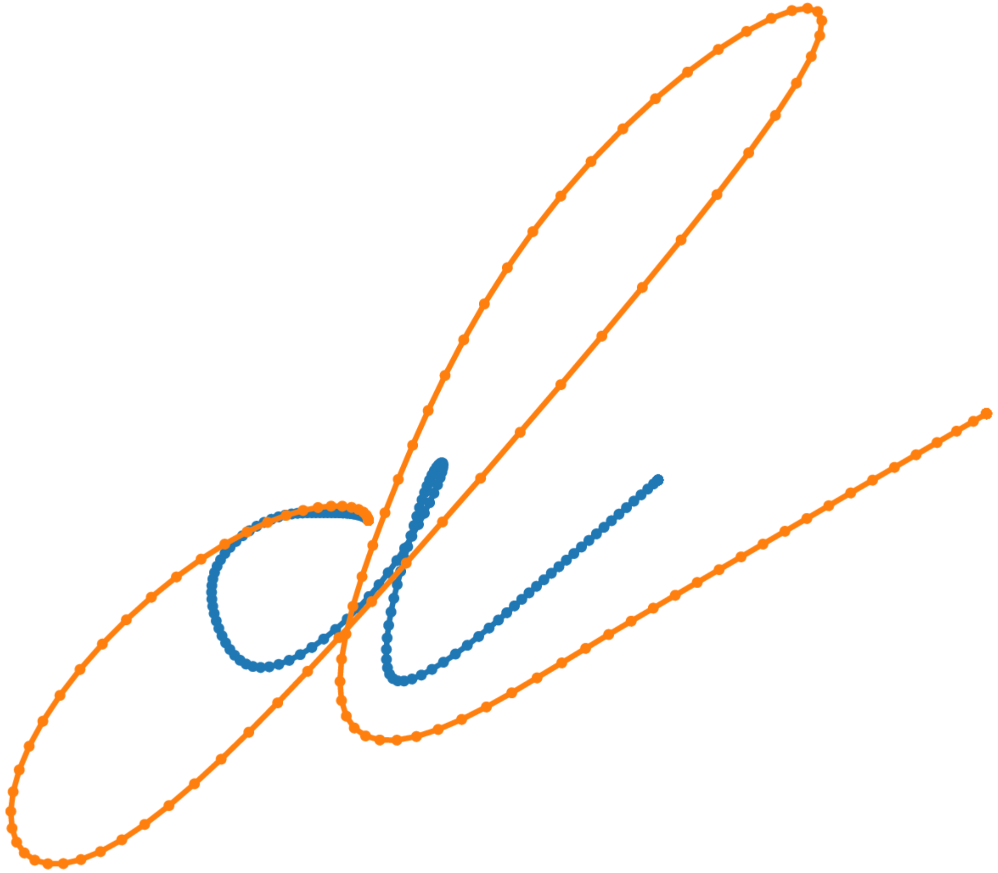


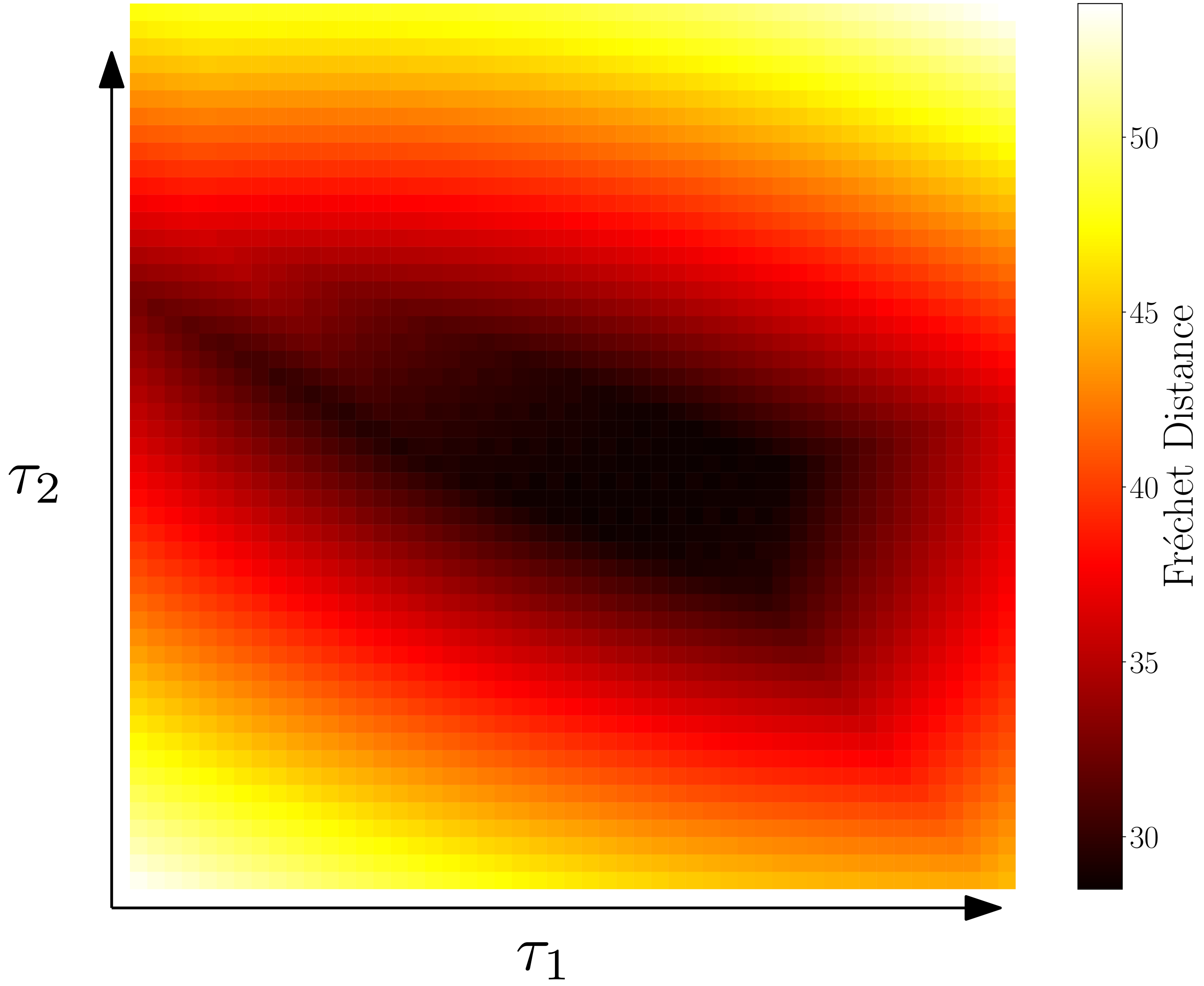
Ewwww.....

Approach II: Continuous Optimization



Approach II: Continuous Optimization





Approach II: Continuous Optimization

Observation: Fréchet under Translation is 1-Lipschitz, i.e.,

$$|d_F(\pi, \sigma + \tau) - d_F(\pi, \sigma + \tau')| \leq \|\tau - \tau'\|$$

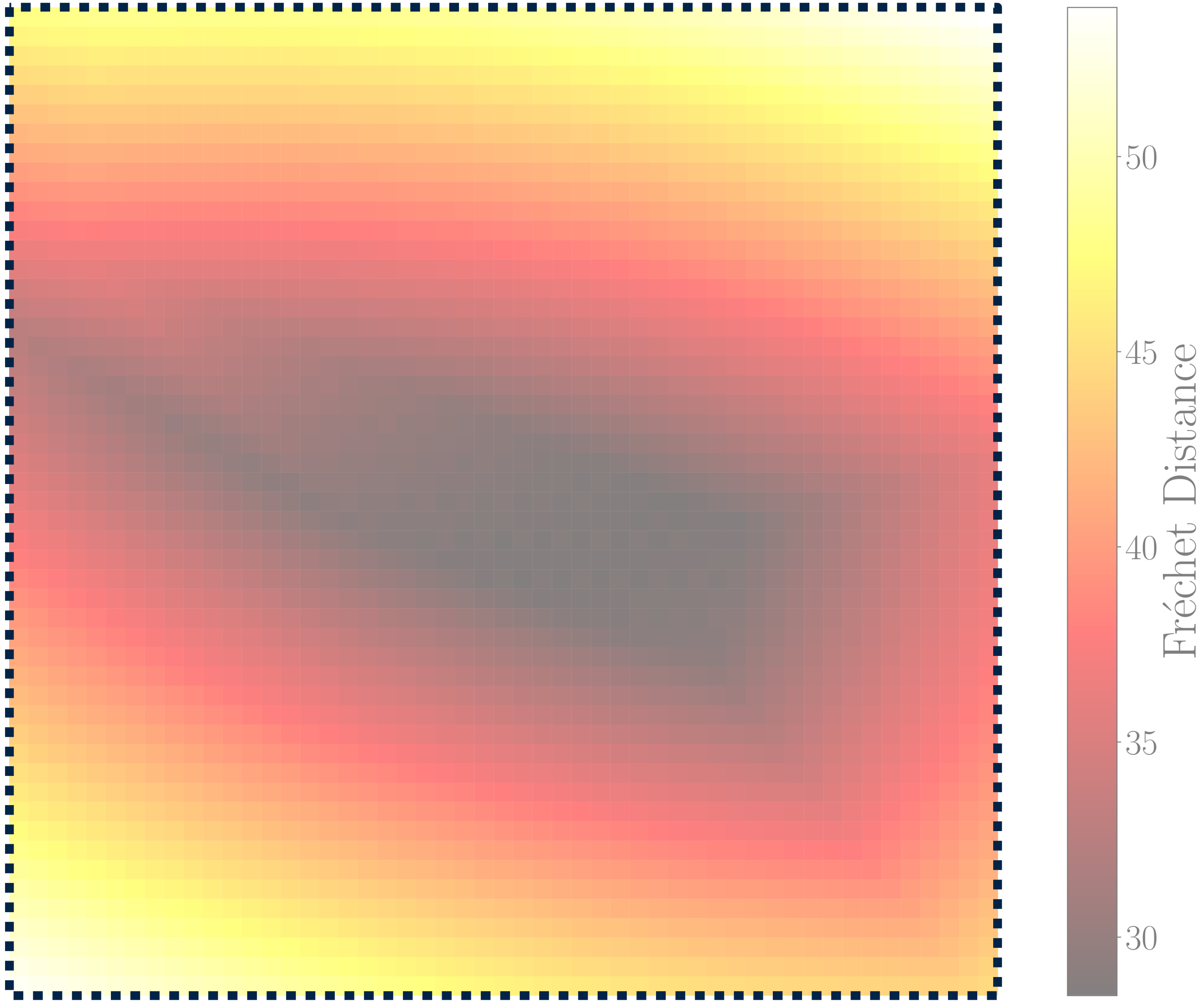
Approach II: Continuous Optimization

Lipschitz Optimization

Approach:

- branch & bound

For each box:



Approach II: Continuous Optimization

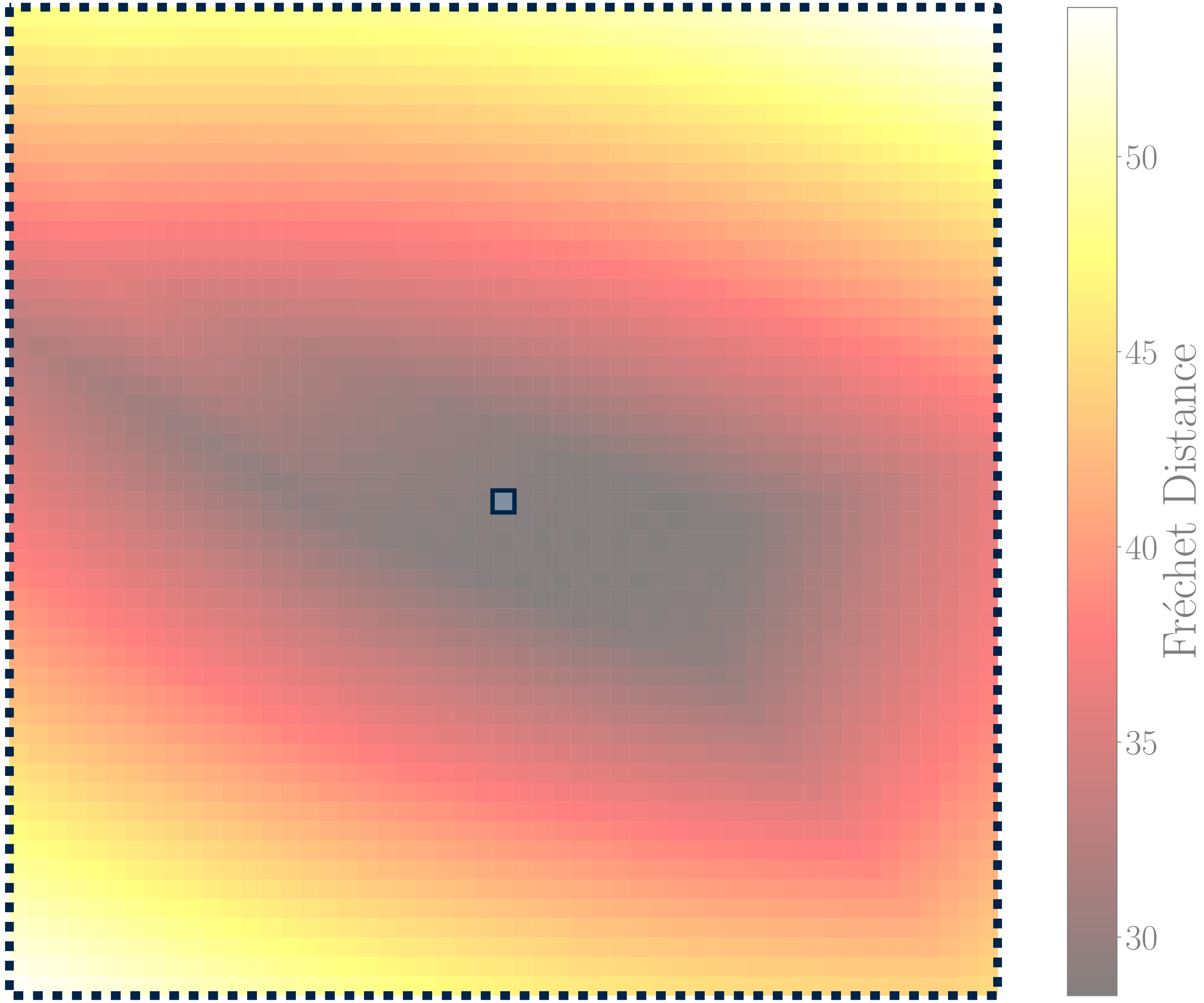
Lipschitz Optimization

Approach:

- branch & bound

For each box:

- if $d_F(\pi, \sigma + \blacksquare) \leq \delta$
  - return LESS



Approach II: Continuous Optimization

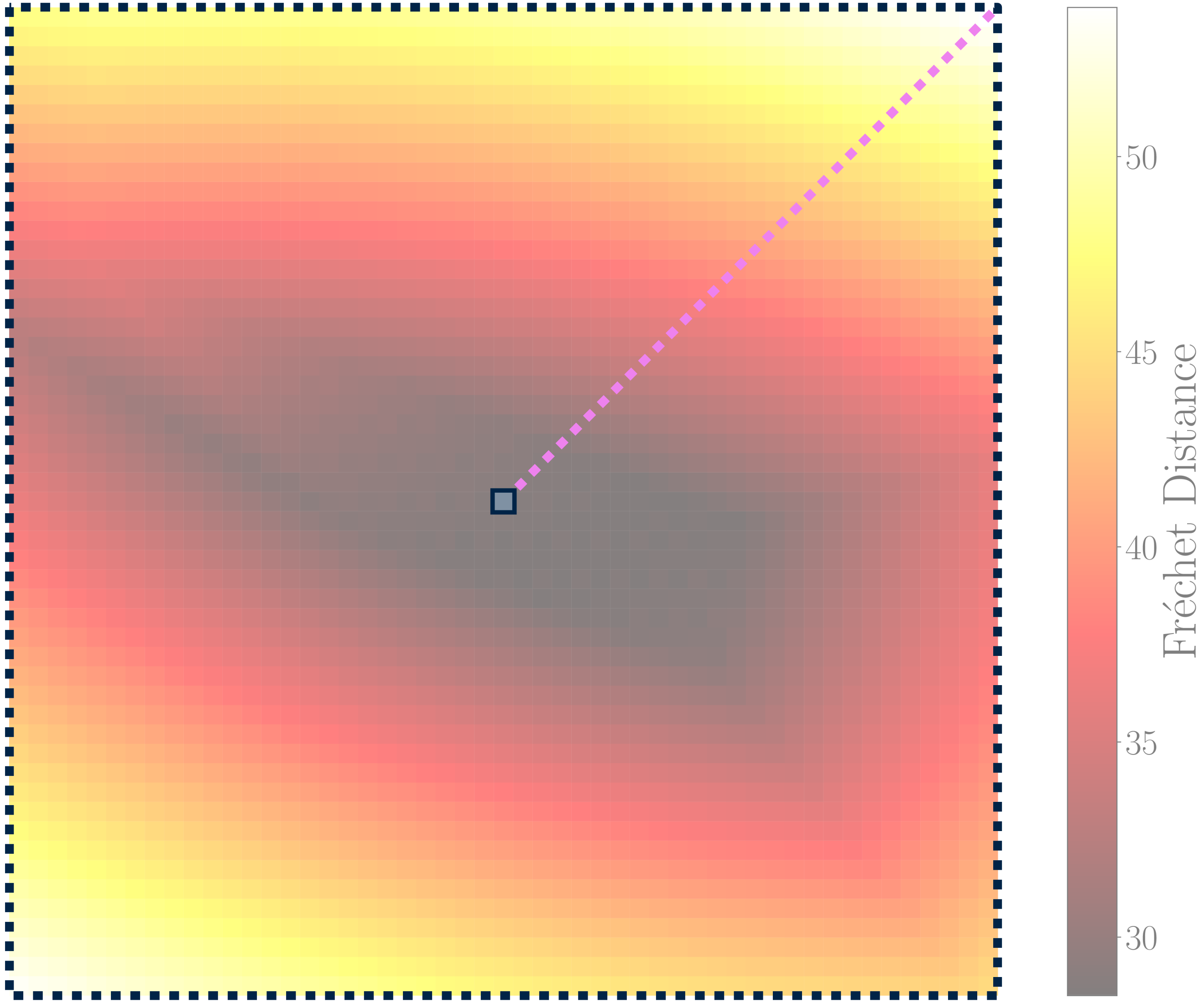
Lipschitz Optimization

Approach:

- branch & bound

For each box:

- if $d_F(\pi, \sigma + \blacksquare) \leq \delta$
  - return LESS
- if $d_F(\pi, \sigma + \blacksquare) > \delta +$
  - skip box



Approach II: Continuous Optimization

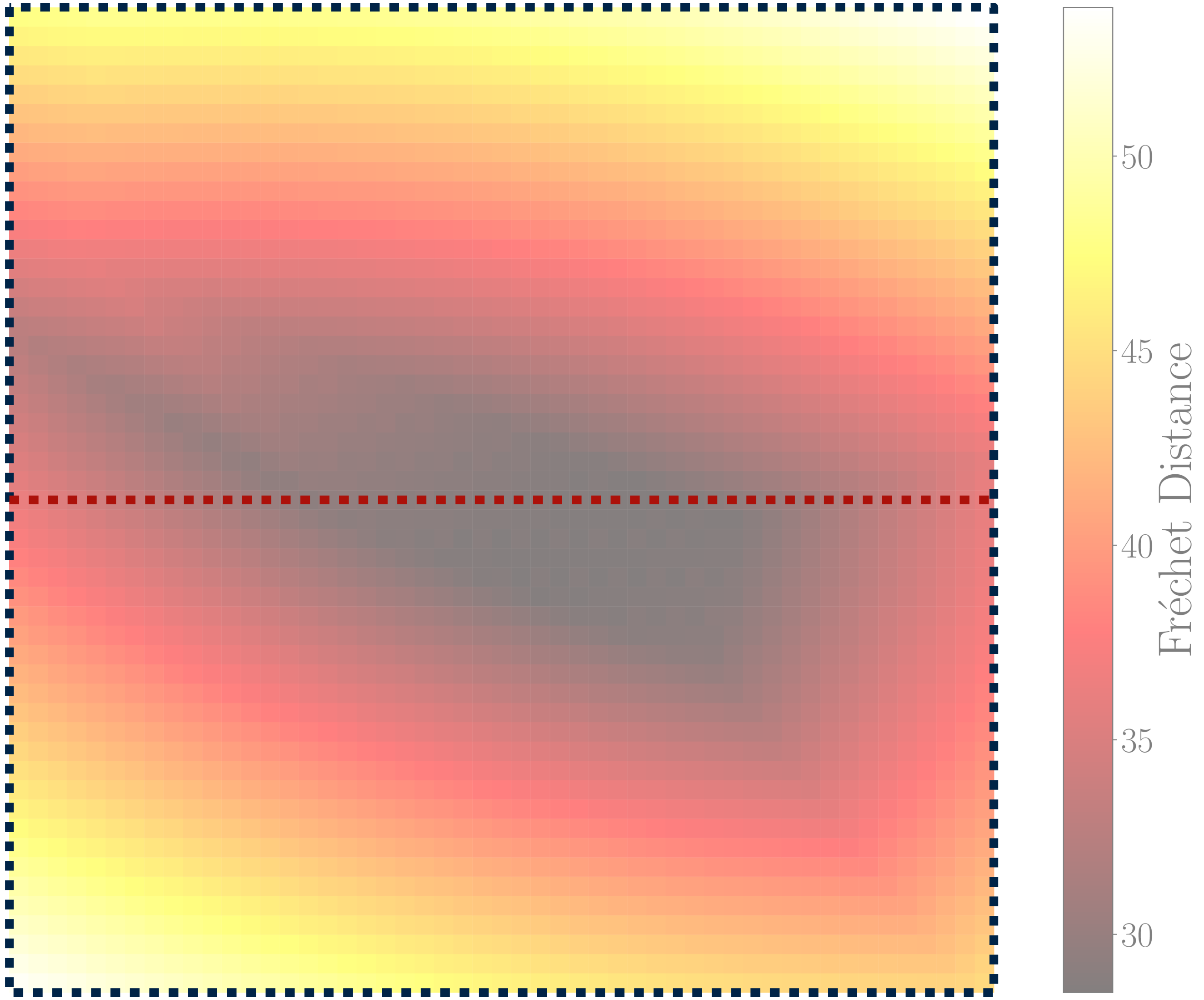
Lipschitz Optimization

Approach:

- branch & bound

For each box:

- if $d_F(\pi, \sigma + \blacksquare) \leq \delta$
  - return LESS
- if $d_F(\pi, \sigma + \blacksquare) > \delta +$
  - skip box
- if both fail: split



Approach II: Continuous Optimization

Issues

- In general, only approximate decisions possible.

Approach II: Continuous Optimization

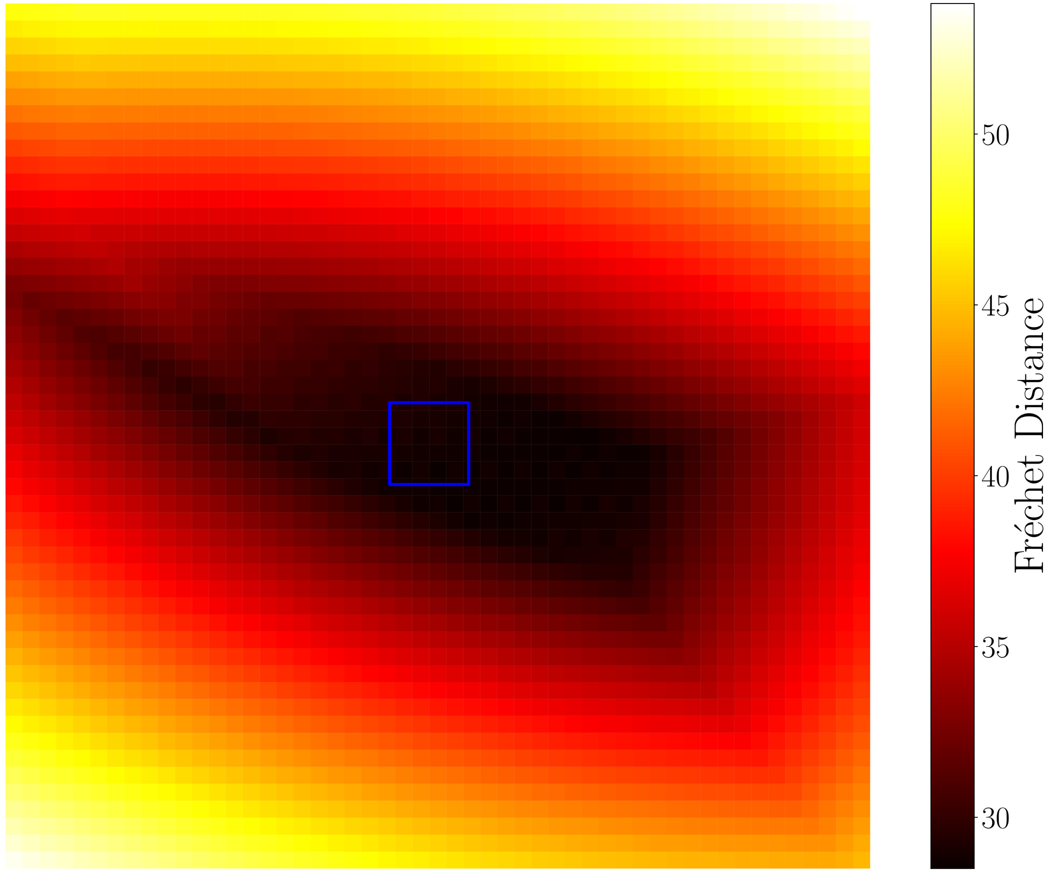
Issues

- In general, only approximate decisions possible.
- Locally highly non-convex:

Approach II: Continuous Optimization

Issues

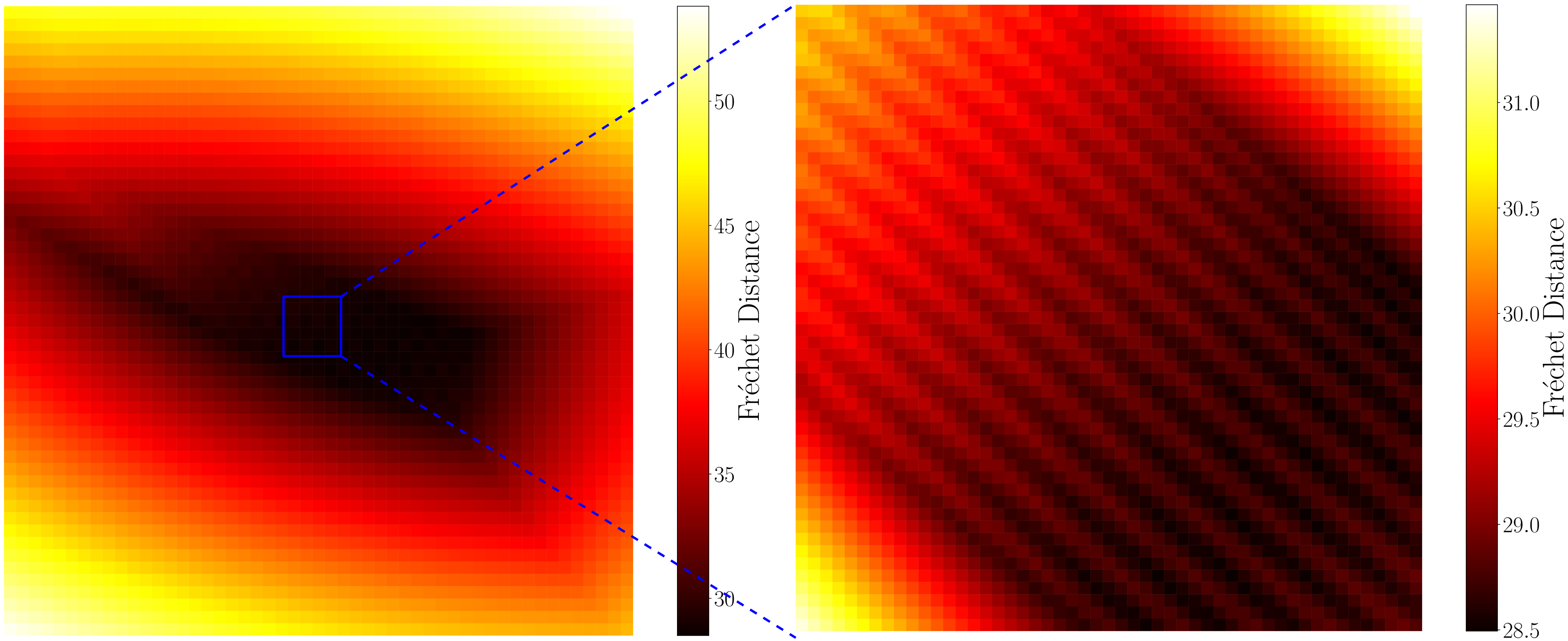
- In general, only approximate decisions possible.
- Locally highly non-convex:



Approach II: Continuous Optimization

Issues

- In general, only approximate decisions possible.
- Locally highly non-convex:



Combine Both Approaches!

Combine Both Approaches!

1) Use Lipschitz optimization to identify important regions

Core Idea

Combine Both Approaches!

1) Use Lipschitz optimization to identify important regions
2) Use arrangement algorithm inside these regions

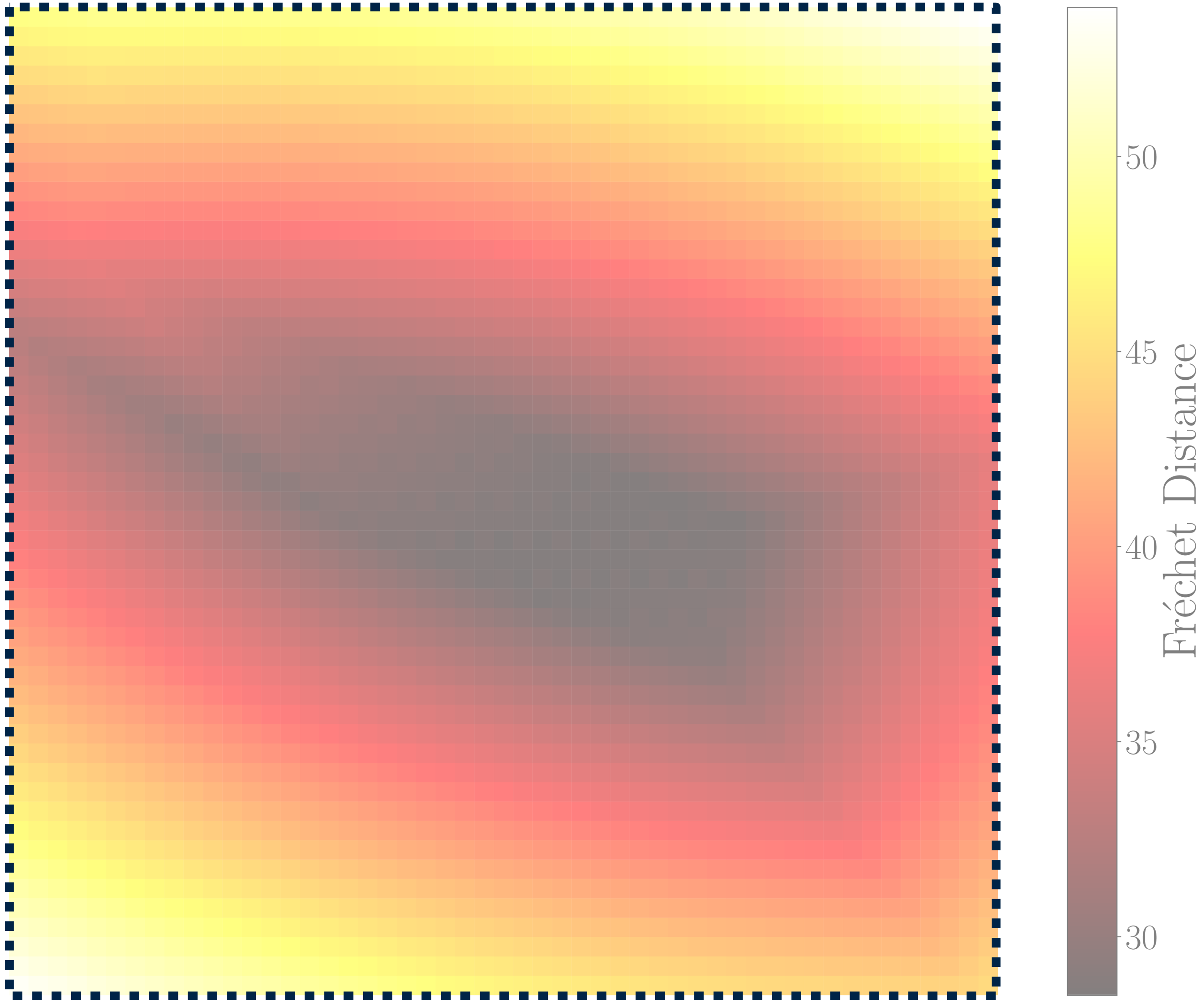
Contribution I: Exact Decider

Contribution I: Exact Decider

Approach:

- augment branch & bound approach
- for each box:
  - estimate arrangement size
- if it is small: build arrangement

exact decision!

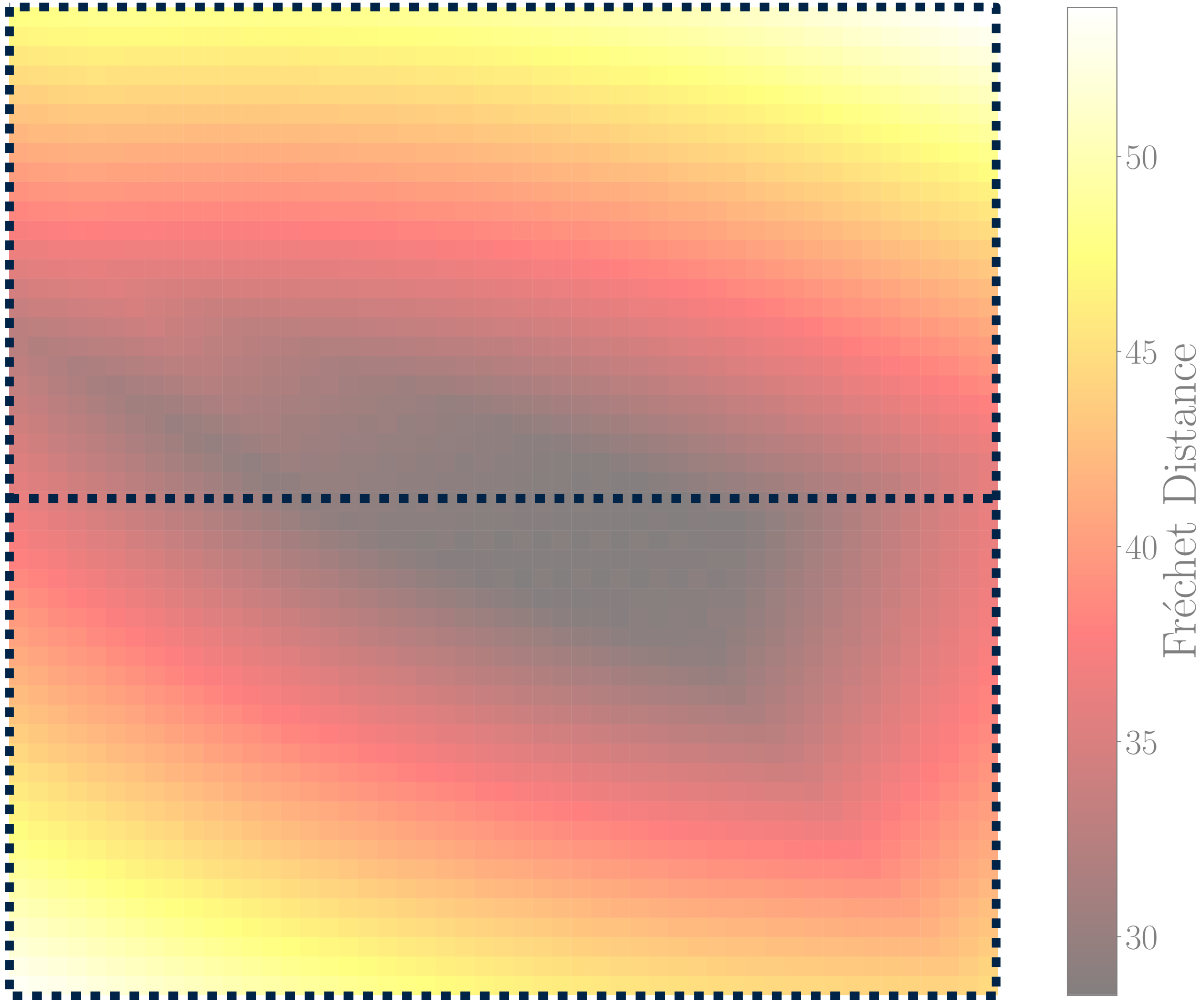


Contribution I: Exact Decider

Approach:

- augment branch & bound approach
- for each box:
  - estimate arrangement size
- if it is small: build arrangement

exact decision!

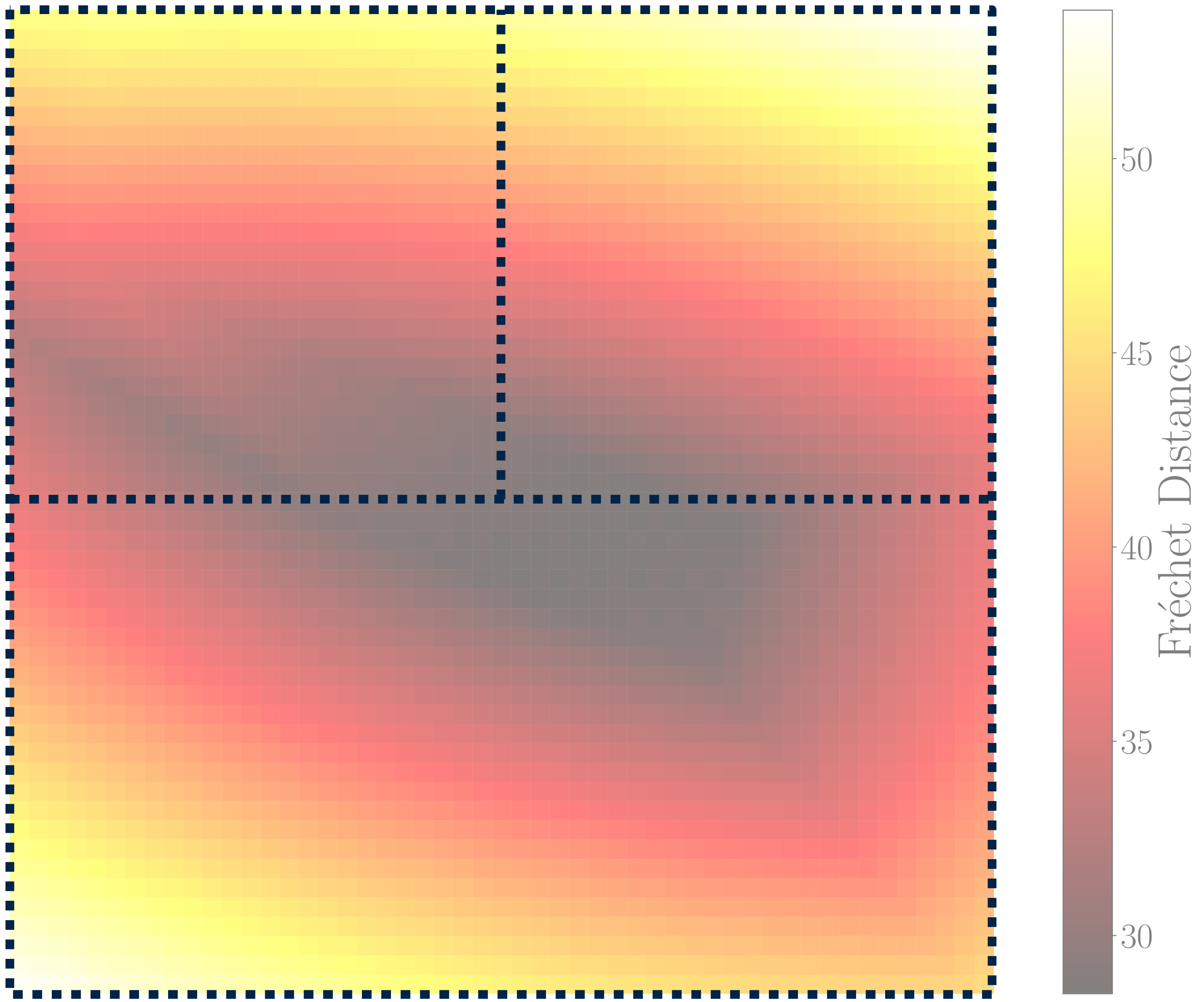


Contribution I: Exact Decider

Approach:

- augment branch & bound approach
- for each box:
  - estimate arrangement size
- if it is small: build arrangement

exact decision!

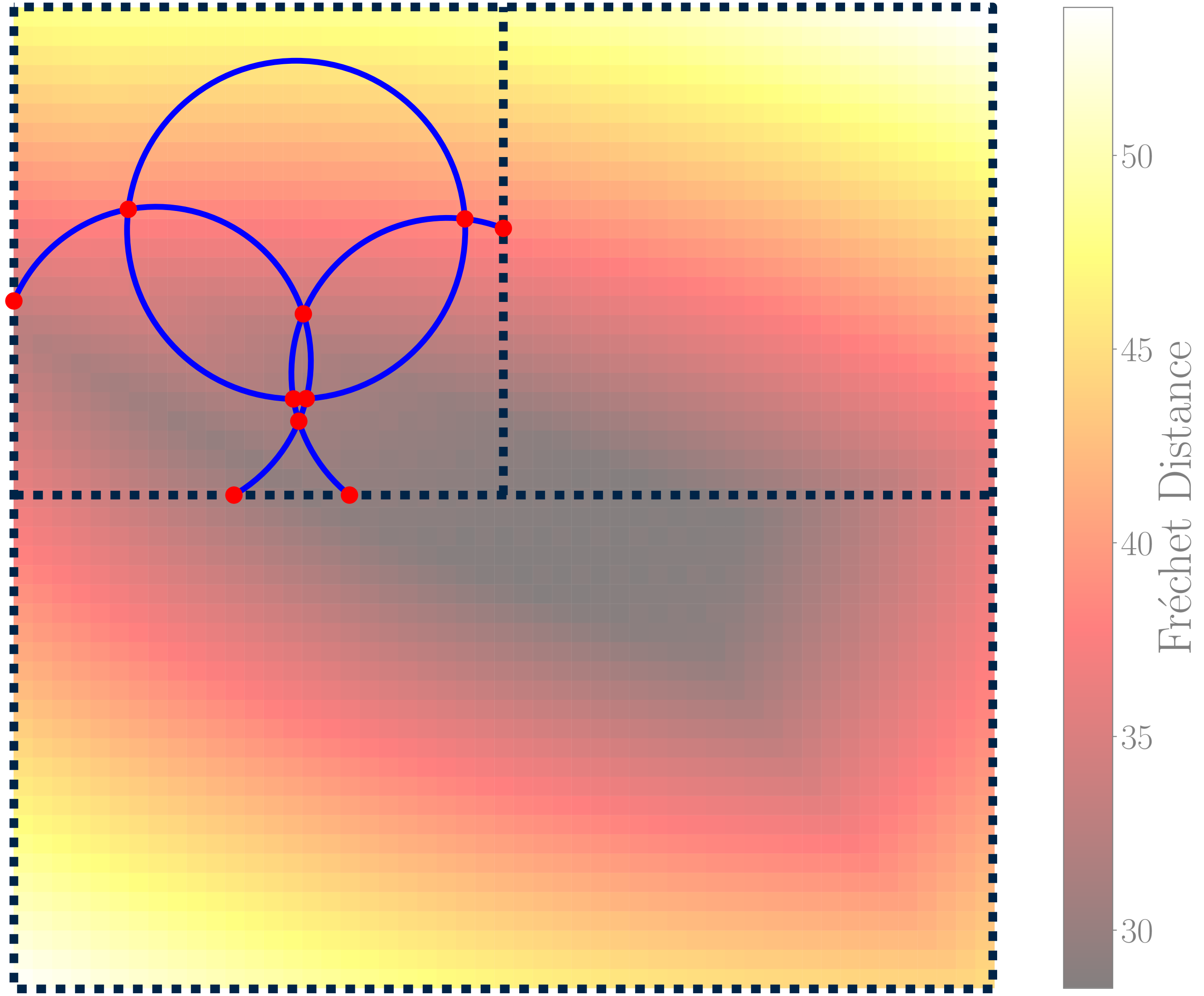


Contribution I: Exact Decider

Approach:

- augment branch & bound approach
- for each box:
  - estimate arrangement size
- if it is small: build arrangement

exact decision!

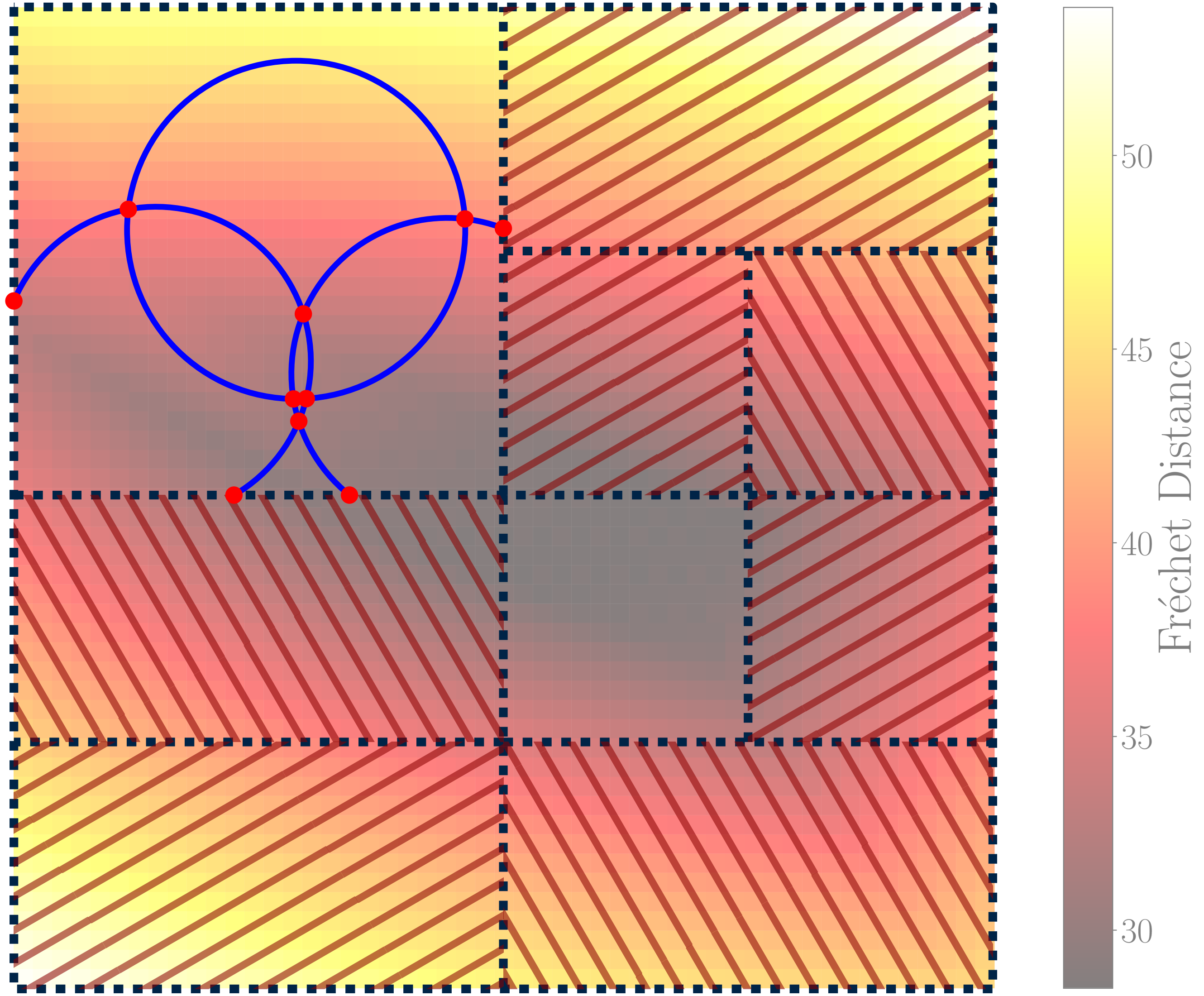


Contribution I: Exact Decider

Approach:

- augment branch & bound approach
- for each box:
  - estimate arrangement size
- if it is small: build arrangement

exact decision!

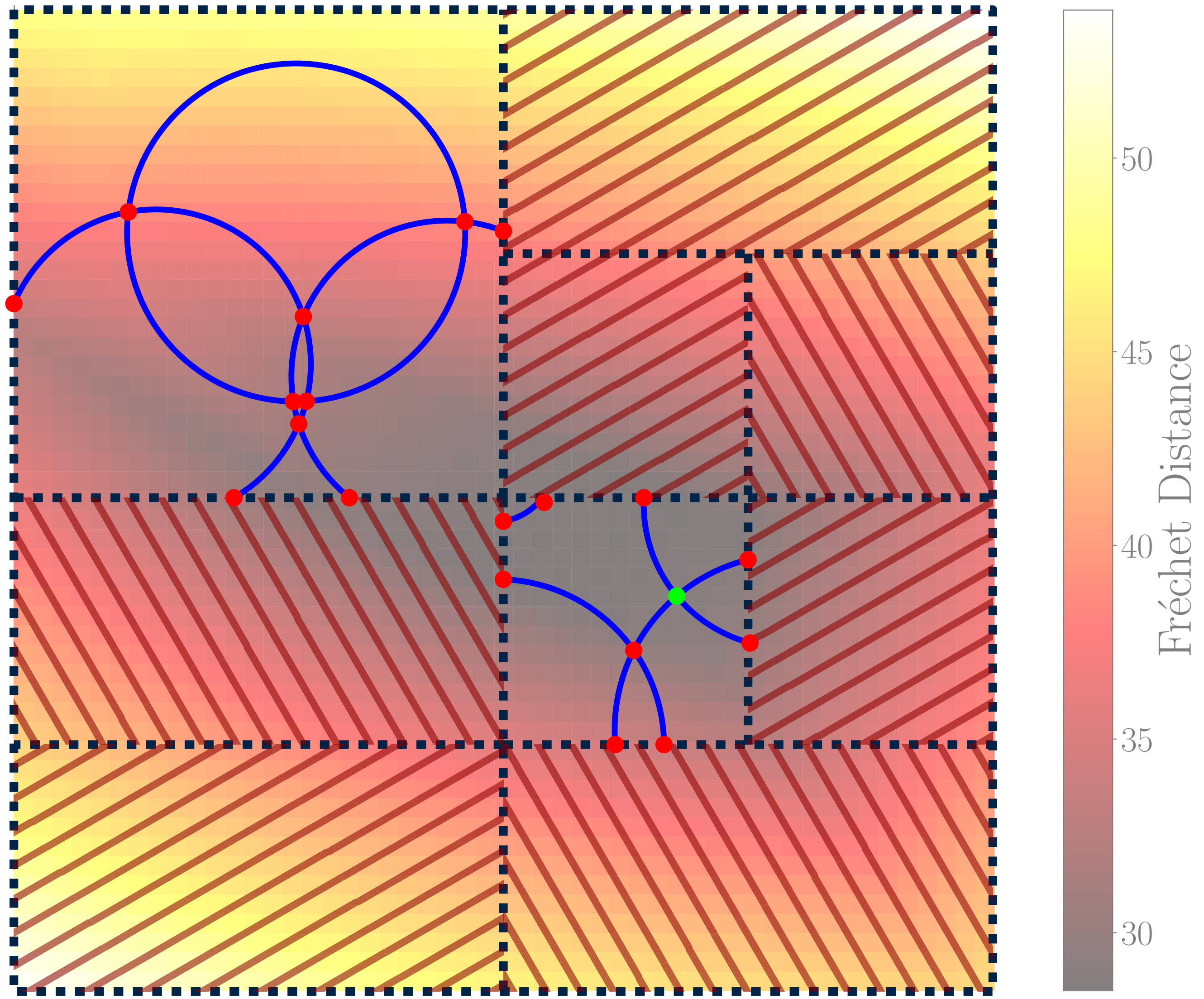


Contribution I: Exact Decider

Approach:

- augment branch & bound approach
- for each box:
  - estimate arrangement size
- if it is small: build arrangement

exact decision!

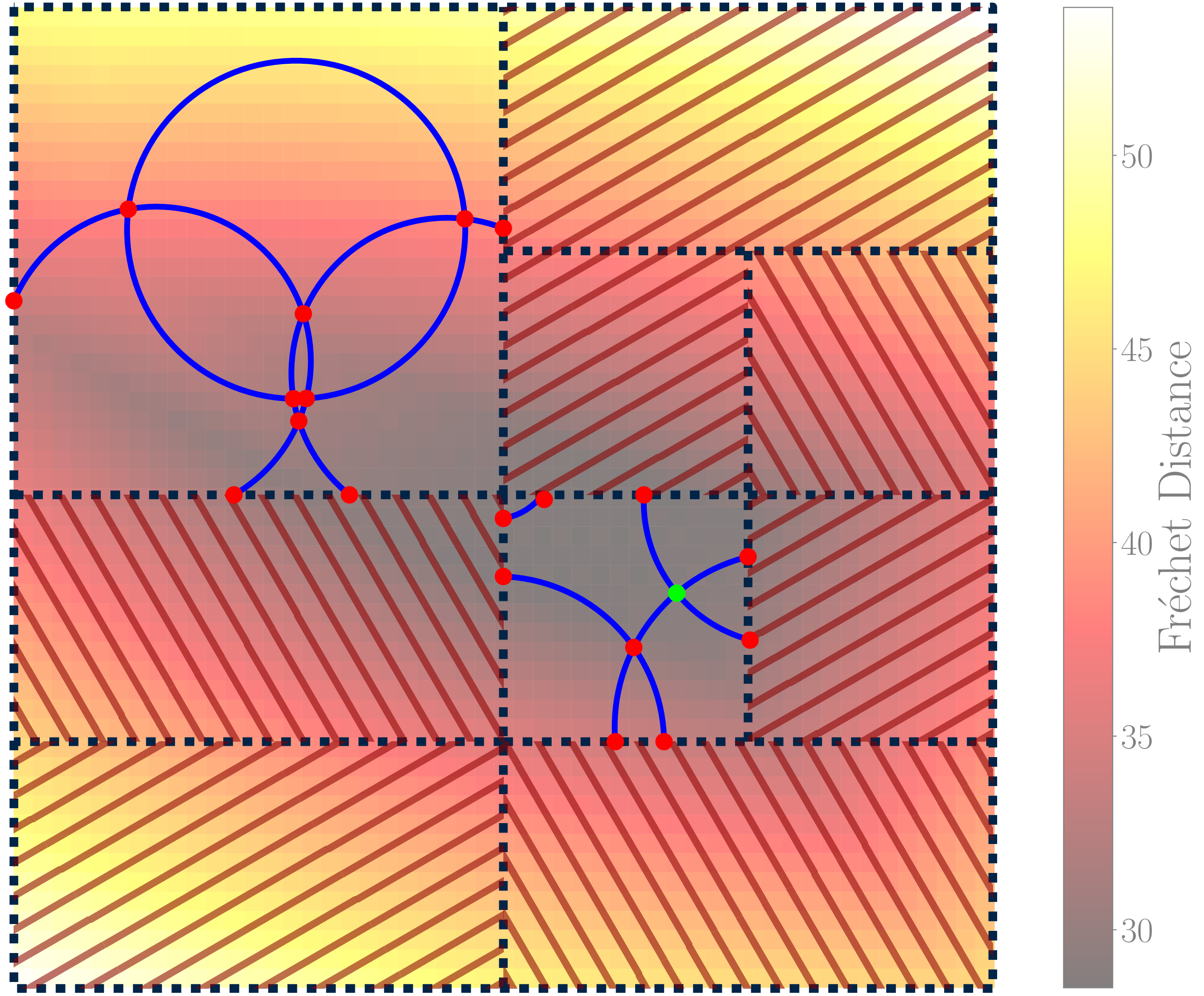


Contribution I: Exact Decider

Approach:

- augment branch & bound approach
- for each box:
  - estimate arrangement size
- if it is small: build arrangement

exact decision!



Issue: When to build the arrangement?

Main Ingredients:

1. Arrangement size estimation
2. Threshold parameter

Contribution I: Exact Decider

Approach:

- augment branch & bound approach
- for each box:
  - estimate arrangement size
- if it is small: build arrangement

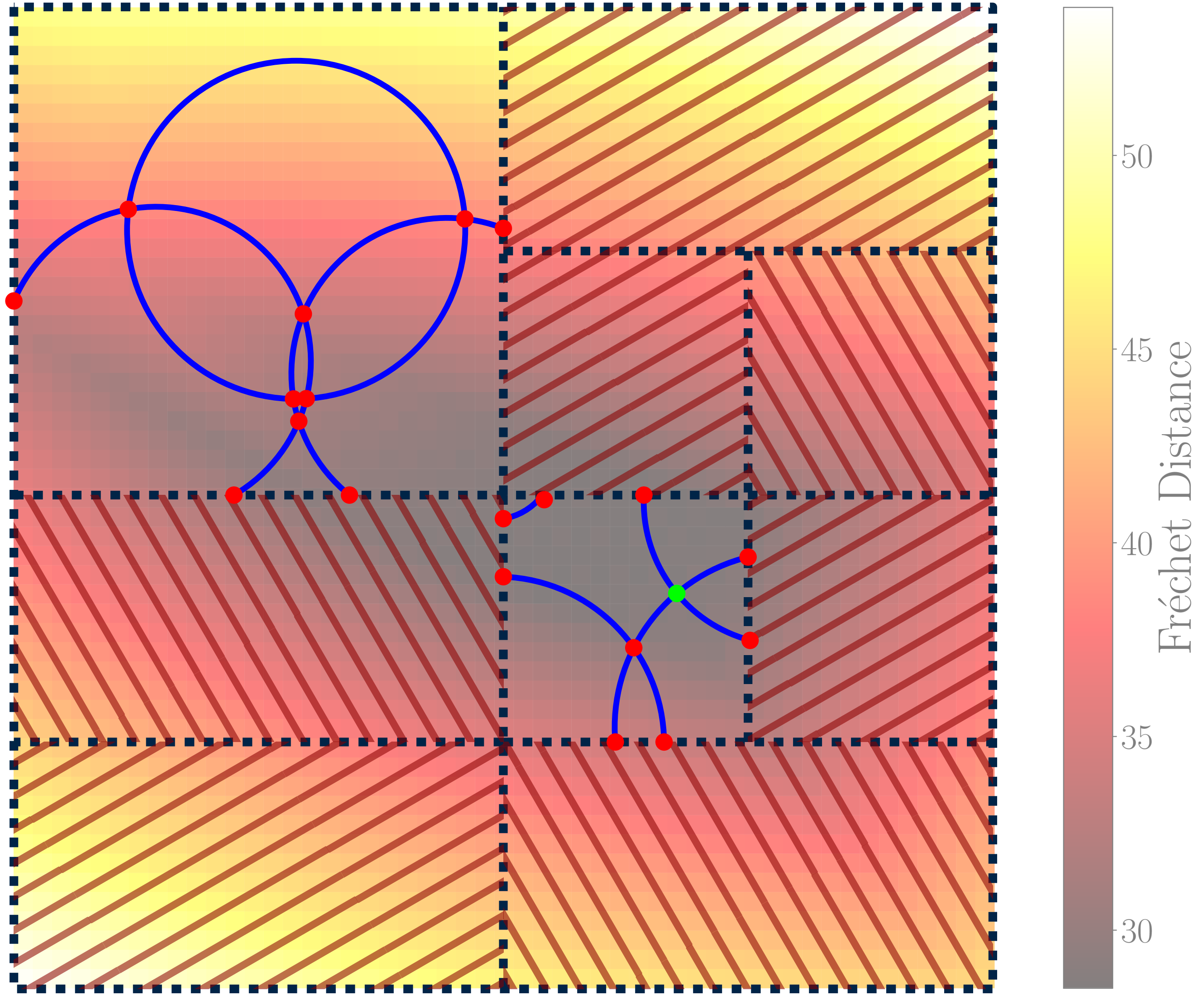
exact decision!

Issue: When to build the arrangement?

Main Ingredients:

1. Arrangement size estimation
2. Threshold parameter

modified kd-tree

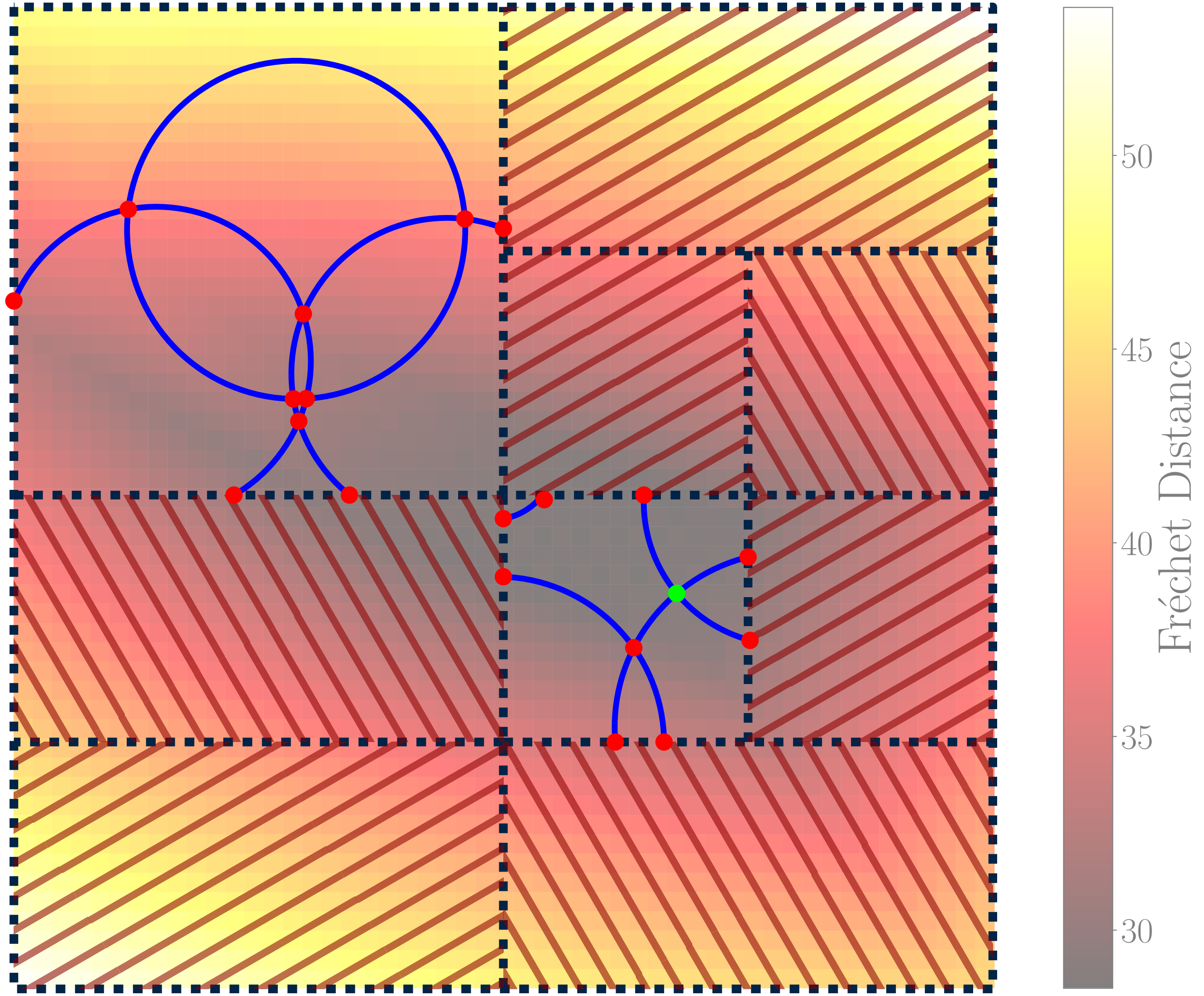


Contribution I: Exact Decider

Approach:

- augment branch & bound approach
- for each box:
  - estimate arrangement size
- if it is small: build arrangement

exact decision!



Issue: When to build the arrangement?

Main Ingredients:

1. Arrangement size estimation
2. Threshold parameter

empirically choose

modified kd-tree

Contribution I: Exact Decider

Implementation Details

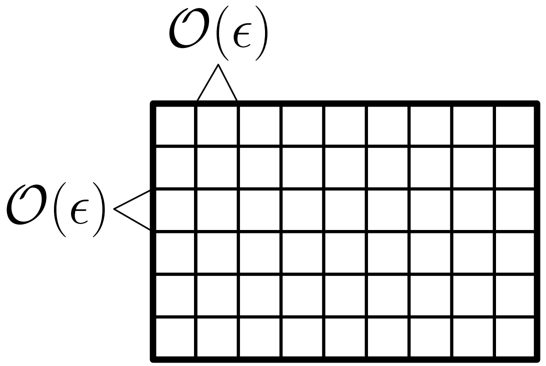
- Adaption of (fixed-translation) [SoCG'19] implementation to discrete case
- Lazy translation
- Parameter choice for arrangement size estimation

Contribution II: From Decider to Value Computation

Contribution II: From Decider to Value Computation

Approaches

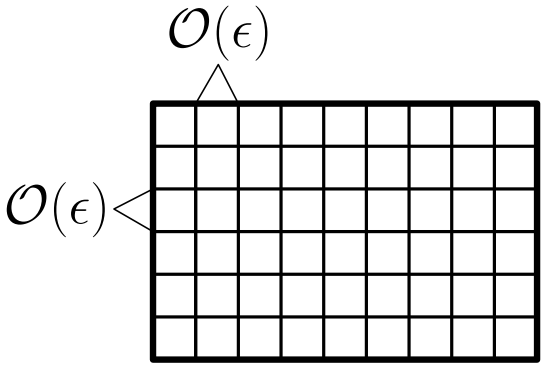
Epsilon-approximate Set:



Contribution II: From Decider to Value Computation

Approaches

Epsilon-approximate Set:



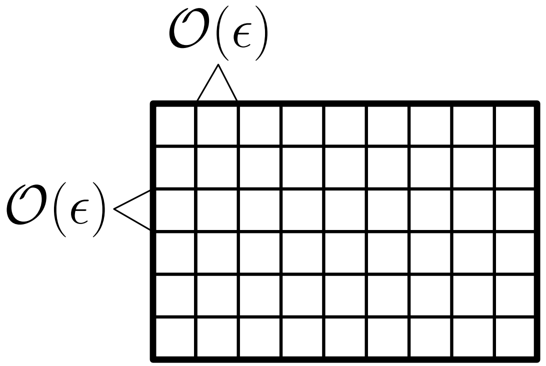
Binary Search via Decision Problem:

- Binary search over δ using decider

Contribution II: From Decider to Value Computation

Approaches

Epsilon-approximate Set:



Binary Search via Decision Problem:

- Binary search over δ using decider

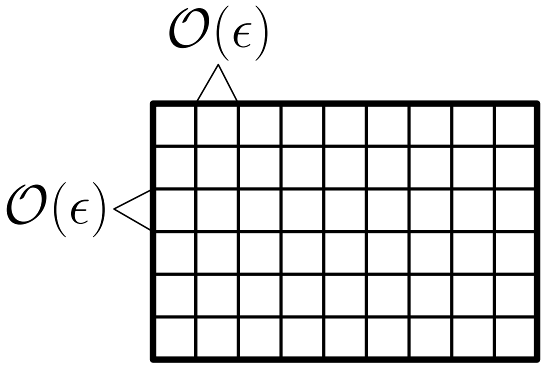
Lipschitz-only Optimization:

- Use plain Lipschitz optimization

Contribution II: From Decider to Value Computation

Approaches

Epsilon-approximate Set:



Binary Search via Decision Problem:

- Binary search over δ using decider

Lipschitz-only Optimization:

- Use plain Lipschitz optimization

Lipschitz-meets-Fréchet:

next slide

Contribution II: From Decider to Value Computation

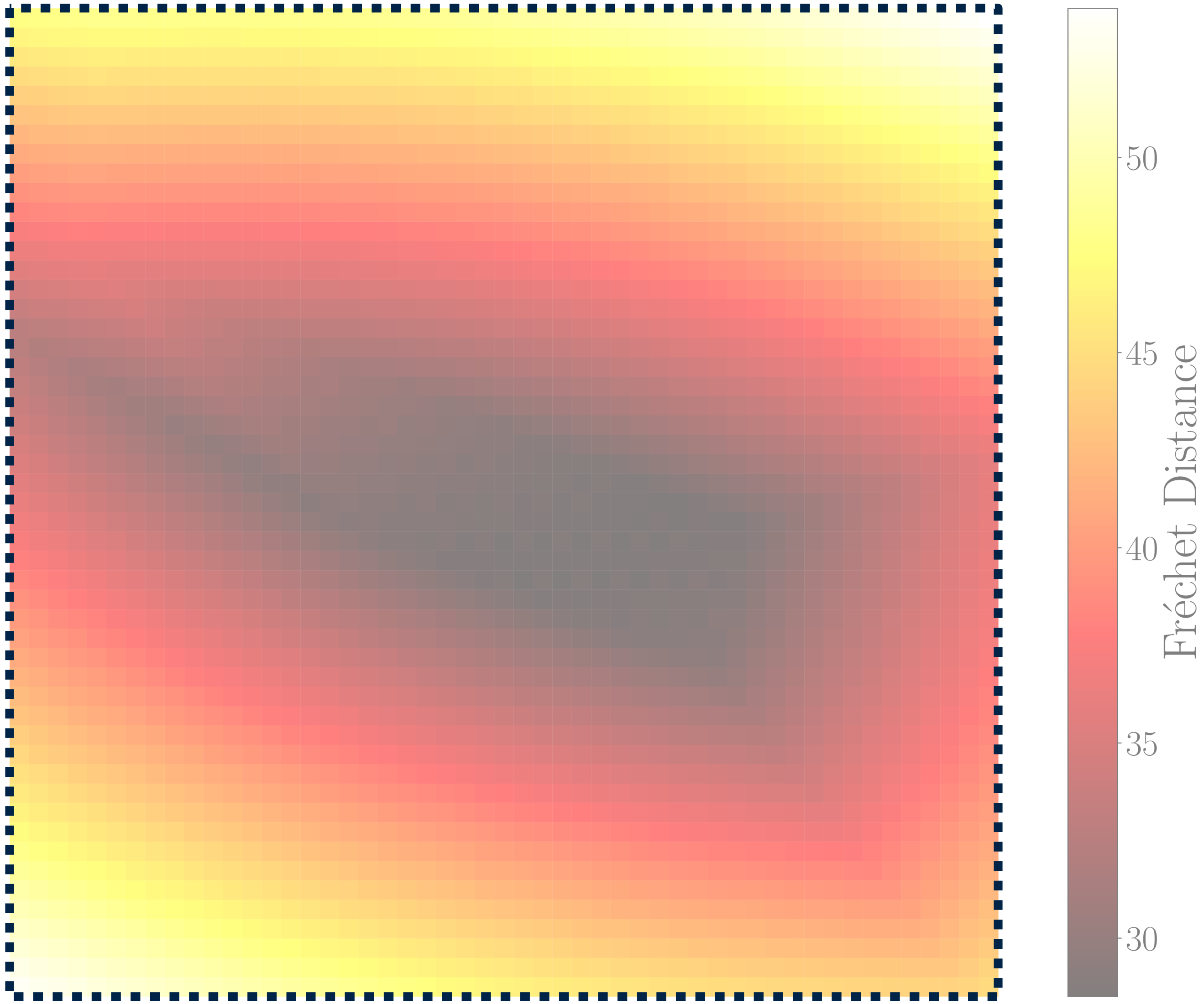
Approach:

1. Maintain local lower bound
2. Maintain global upper bound
3. Arrangement size estimation

adapt!

For each box:

Lipschitz-meets-Fréchet



Contribution II: From Decider to Value Computation

Lipschitz-meets-Fréchet

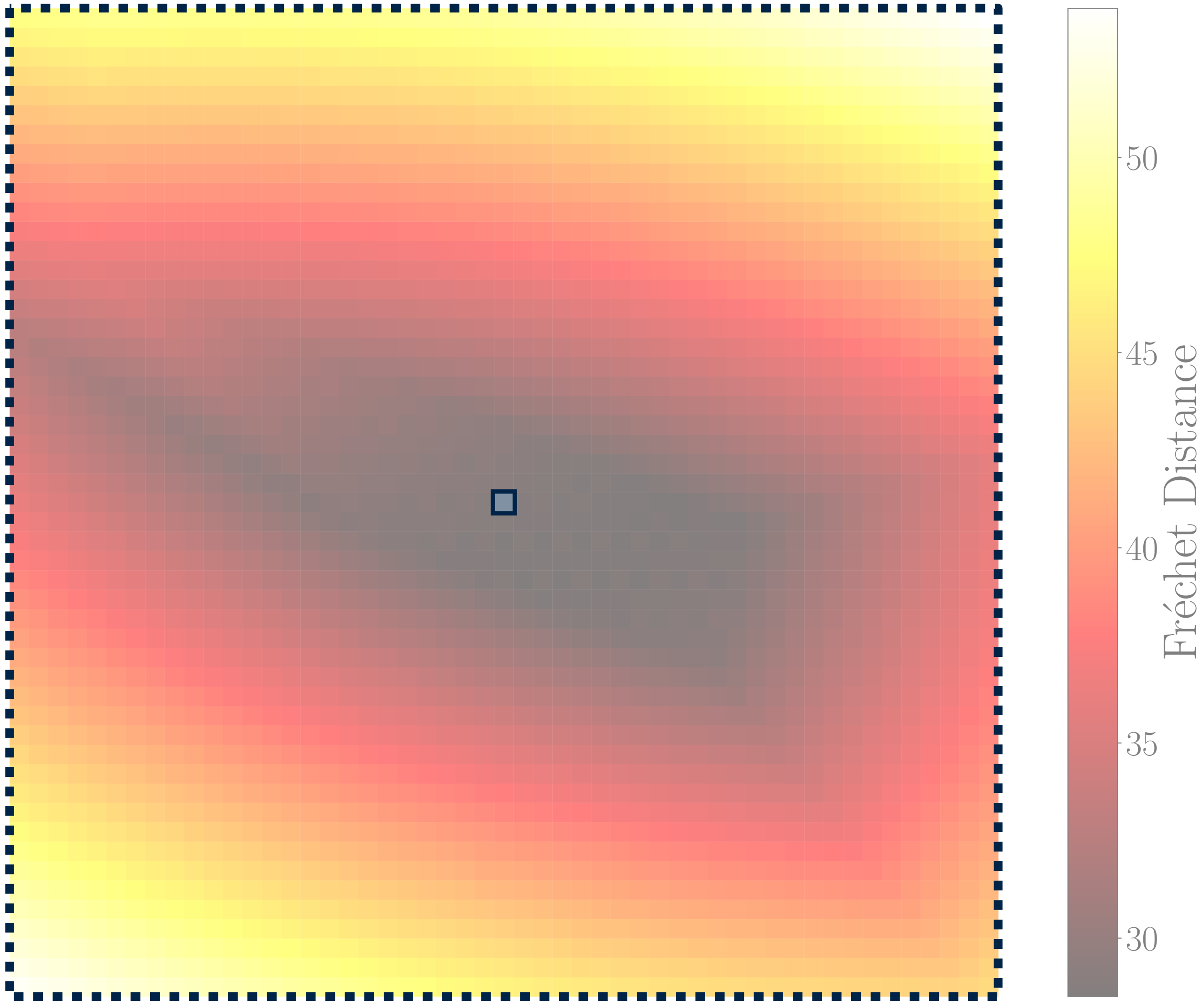
Approach:

1. Maintain local lower bound
2. Maintain global upper bound
3. Arrangement size estimation

adapt!

For each box:

- update *global* upper bound:
  - $\min\{\mathrm{ub}, d_F(\pi, \sigma + \square)\}$



Contribution II: From Decider to Value Computation

Approach:

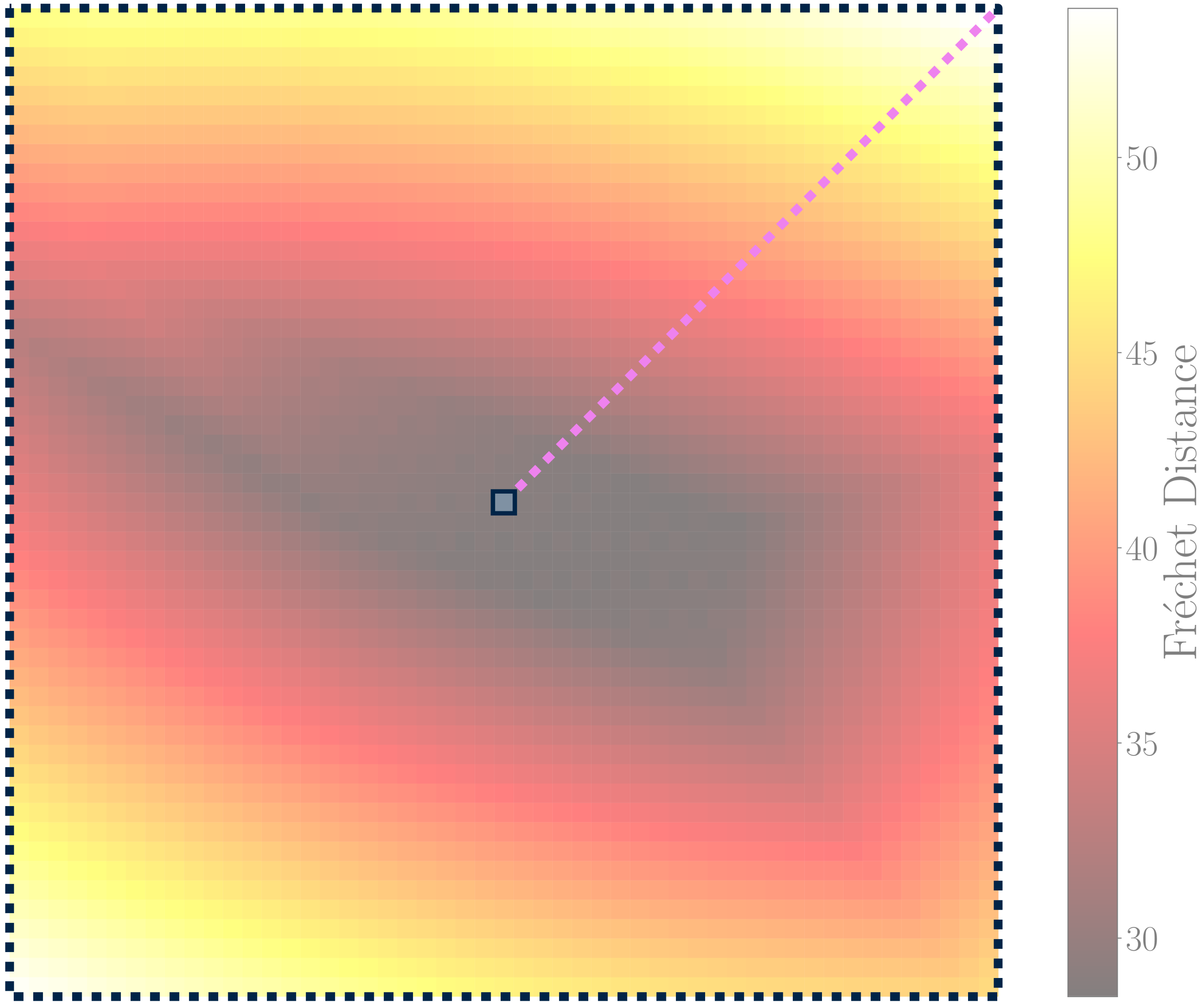
1. Maintain local lower bound
2. Maintain global upper bound
3. Arrangement size estimation

adapt!

For each box:

- update *global* upper bound:
  - $\min\{\text{ub}, d_F(\pi, \sigma + \blacksquare)\}$
- update *local* lower bound:
  - $\max\{\text{lb}, d_F(\pi, \sigma + \blacksquare) - \}$

Lipschitz-meets-Fréchet



Contribution II: From Decider to Value Computation

Approach:

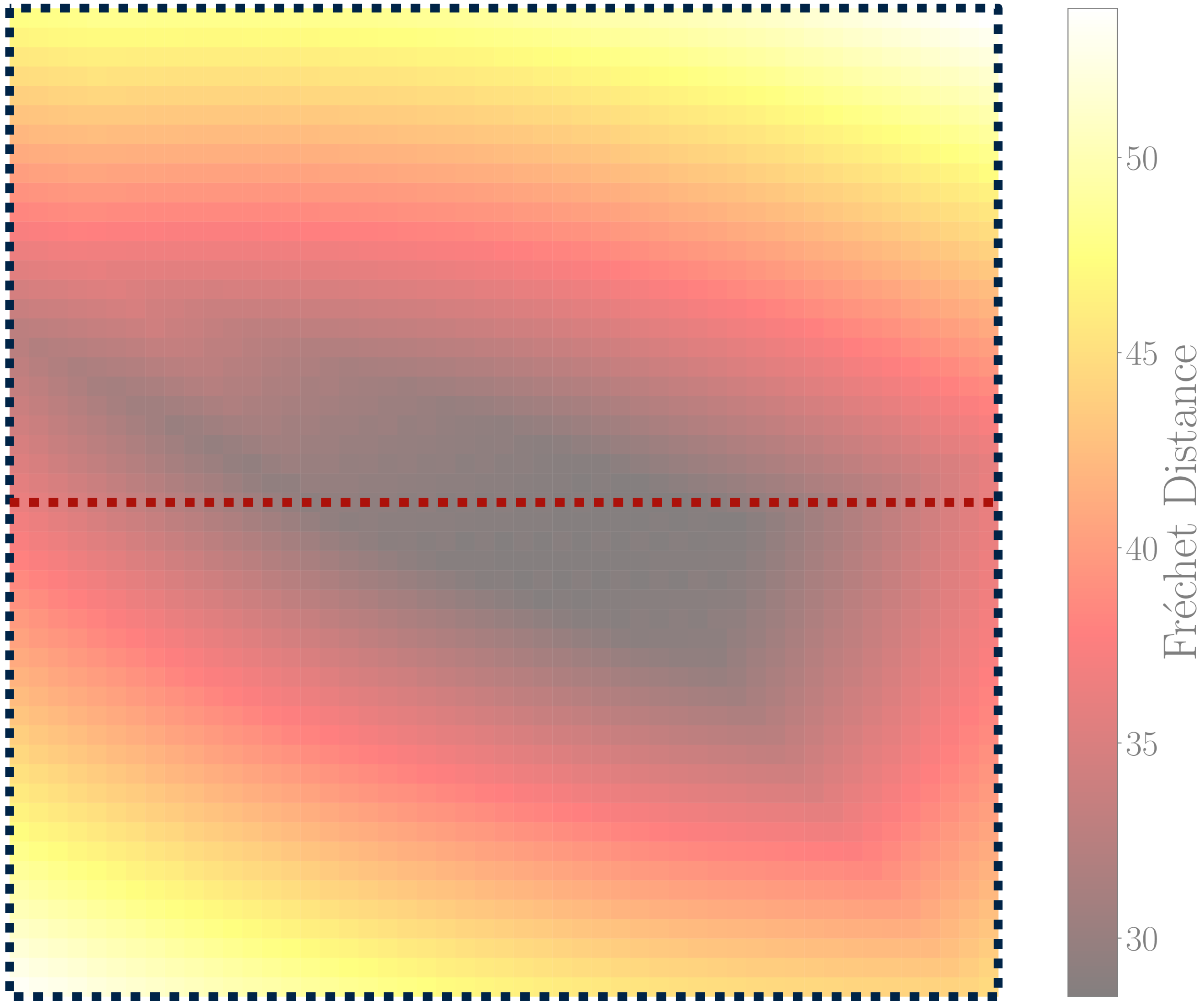
1. Maintain local lower bound
2. Maintain global upper bound
3. Arrangement size estimation

adapt!

For each box:

- update *global* upper bound:
  - $\min\{\mathrm{ub}, d_F(\pi, \sigma + \square)\}$
- update *local* lower bound:
  - $\max\{\mathrm{lb}, d_F(\pi, \sigma + \square) - \}$
- if $\mathrm{ub} > \mathrm{lb} + \epsilon$: split

Lipschitz-meets-Fréchet



Contribution II: From Decider to Value Computation

Lipschitz-meets-Fréchet

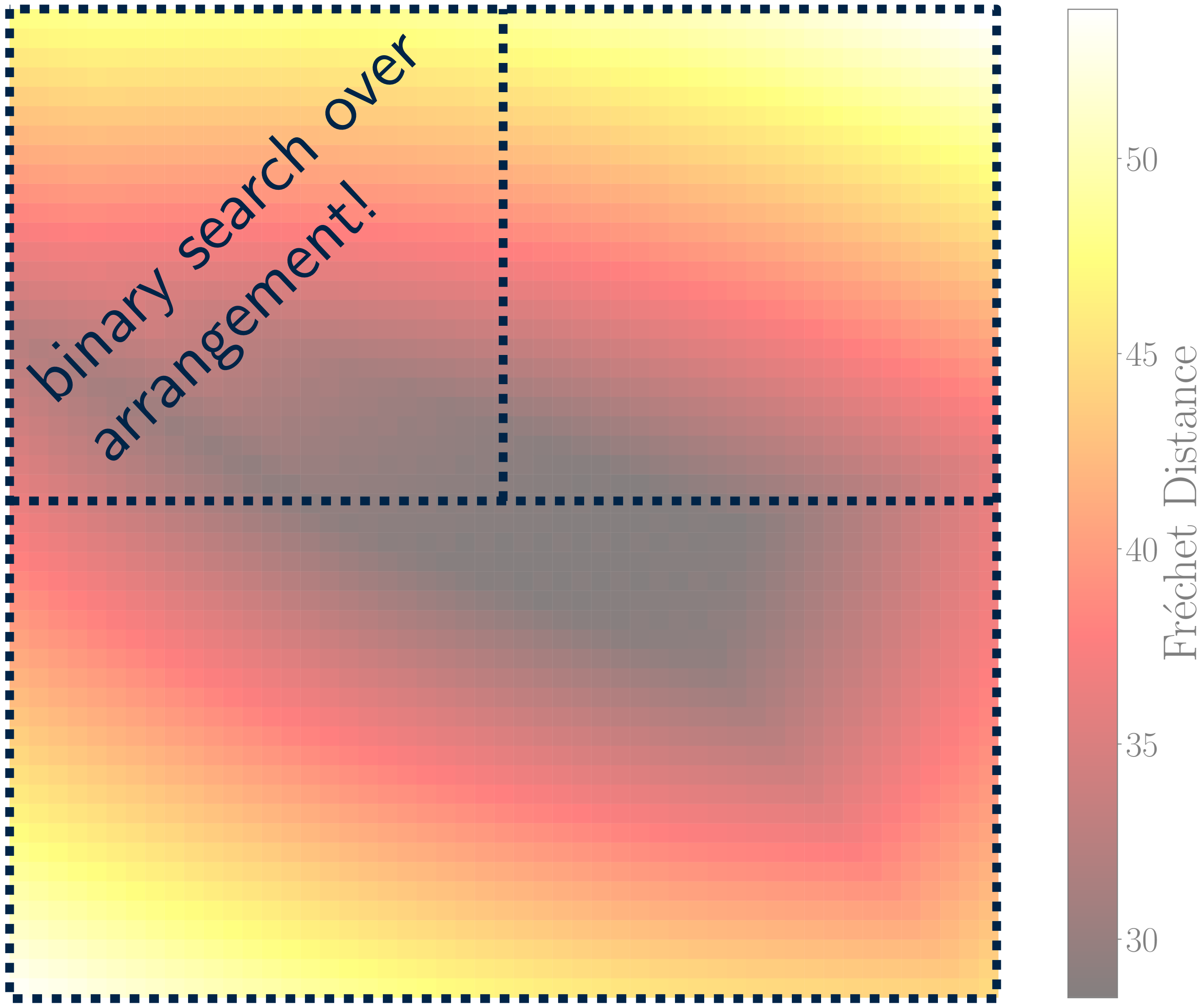
Approach:

1. Maintain local lower bound
2. Maintain global upper bound
3. Arrangement size estimation

adapt!

For each box:

- update *global* upper bound:
  - $\min\{\text{ub}, d_F(\pi, \sigma + \square)\}$
- update *local* lower bound:
  - $\max\{\text{lb}, d_F(\pi, \sigma + \square) - \}$
- if ub $>$ lb $+ \epsilon$: split



Contribution II: From Decider to Value Computation

Lipschitz-meets-Fréchet: Implementation Details

- Arrangement size estimation
- Initial estimates
- Priority queue on lower bound $\rightarrow$ no regret strategy!

Experiments

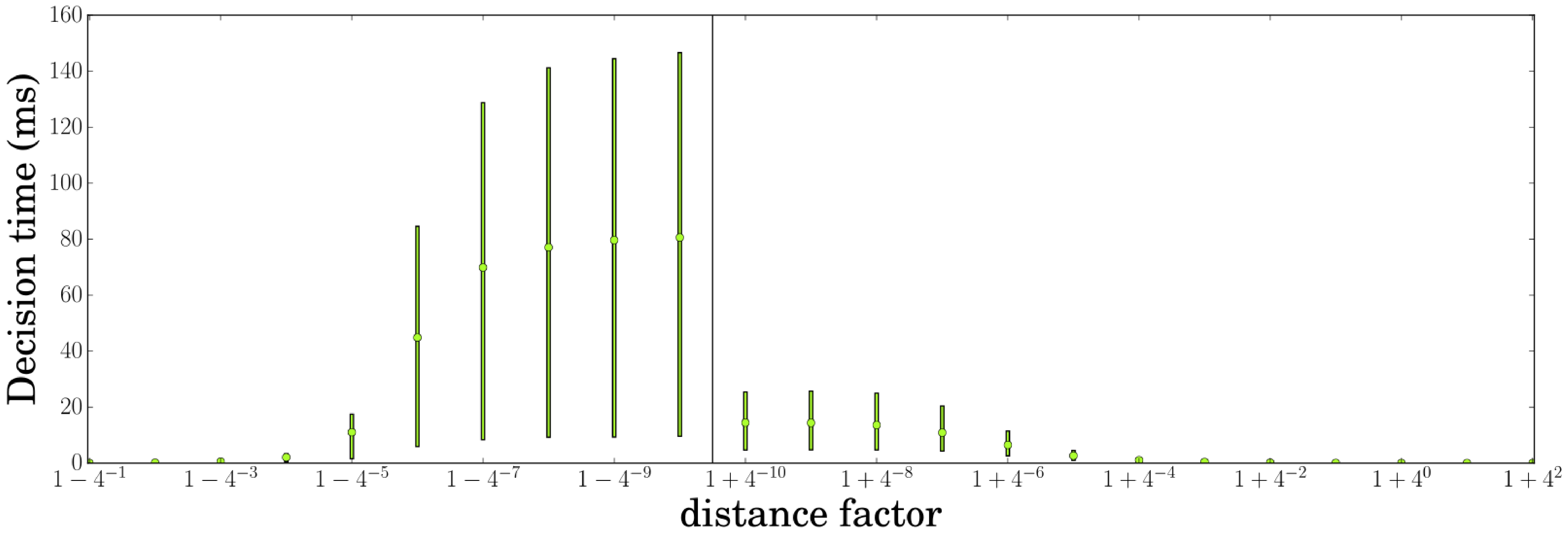
Data Sets

Data set	Type	#Curves	Mean #vertices
SIGSPATIAL	synthetic GPS-like	20199	247.8
CHARACTERS	20 handwritten chars	2858 (142.9 per character)	120.9

Experiments

Running Times

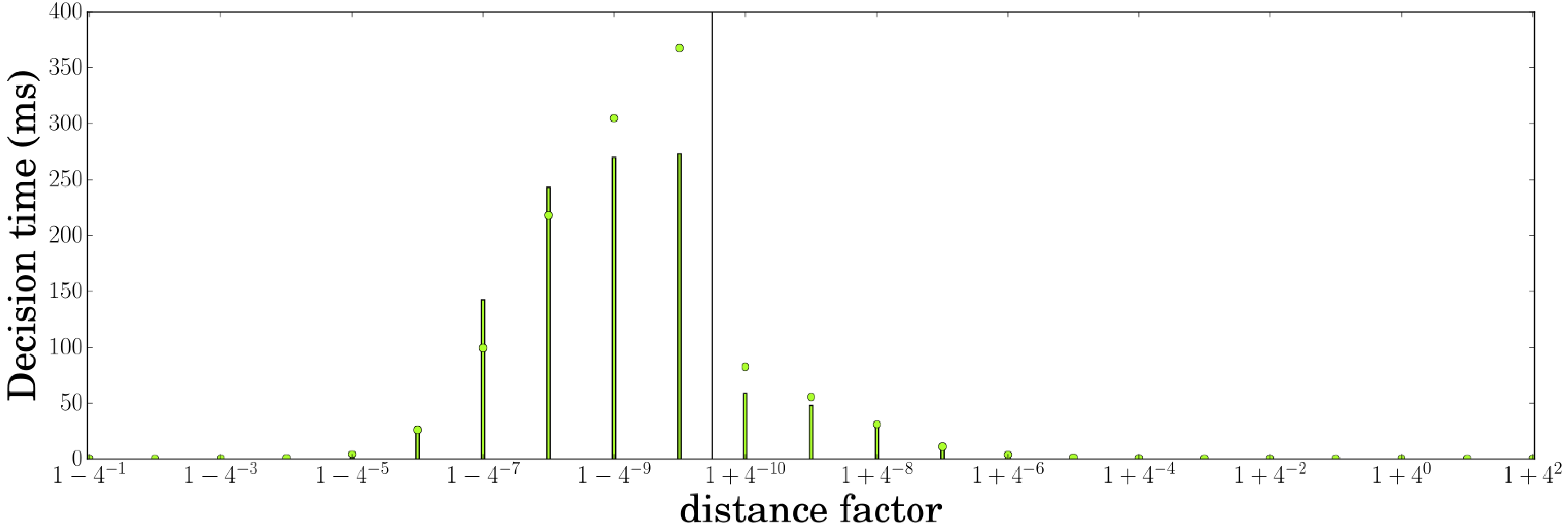
Same-characters:



Experiments

Running Times

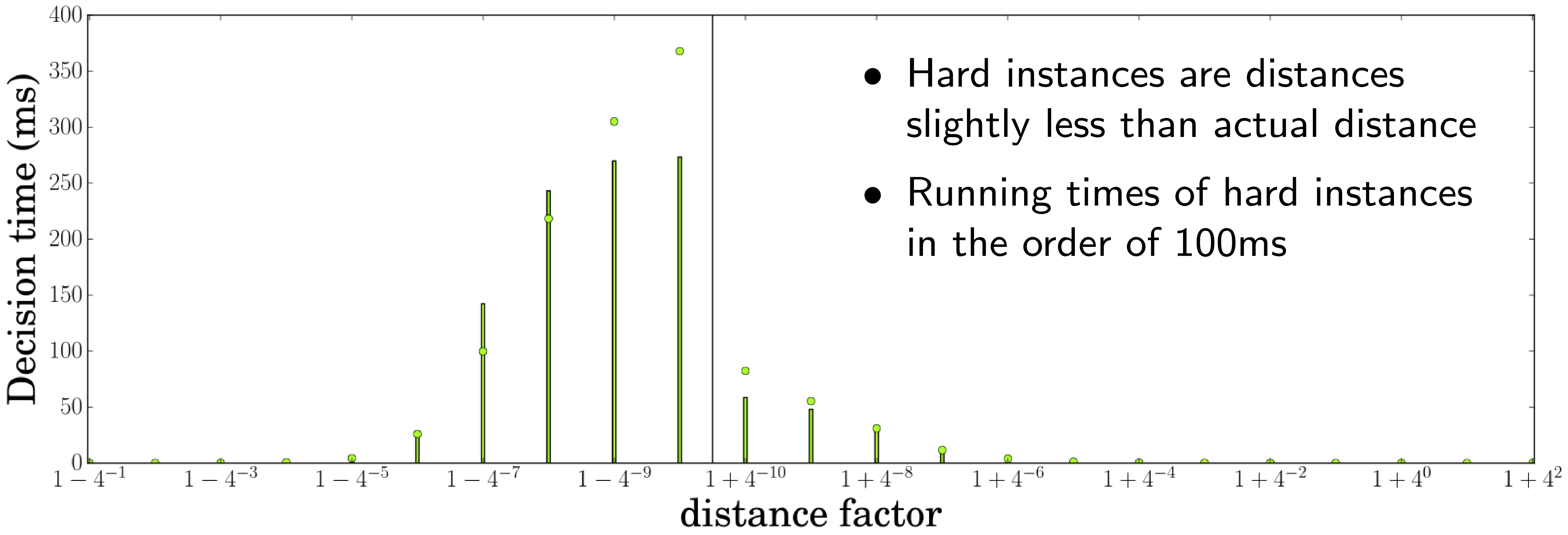
SIGSPATIAL:



Experiments

Running Times

SIGSPATIAL:

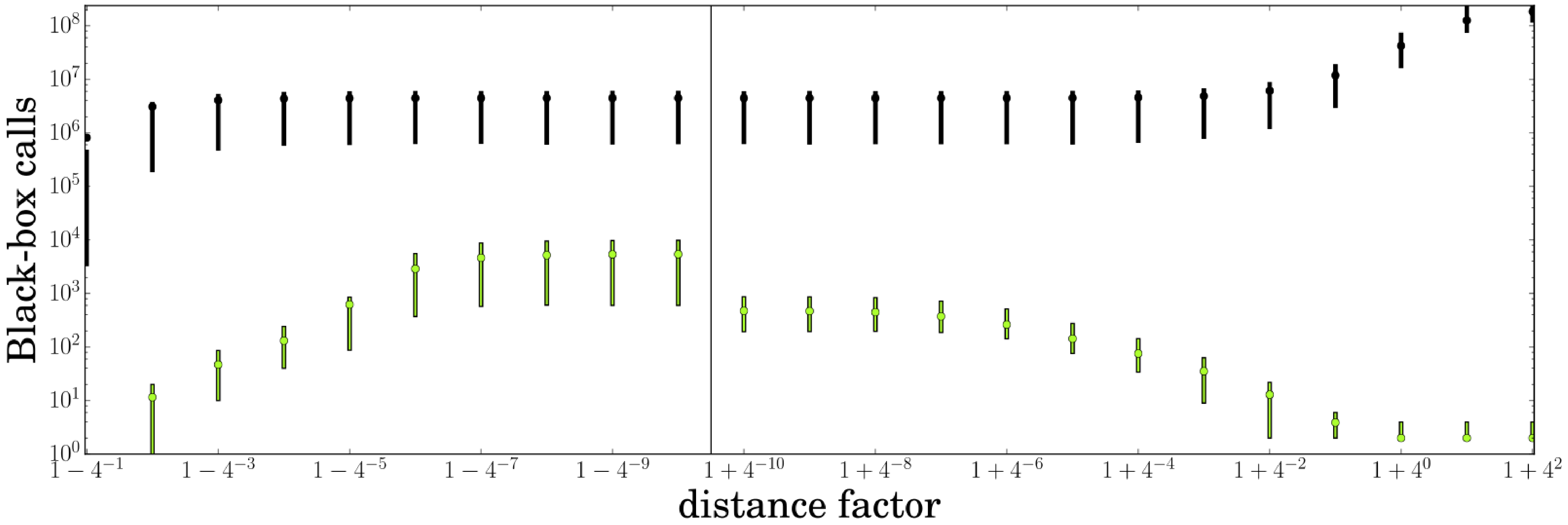


- Hard instances are distances slightly less than actual distance
- Running times of hard instances in the order of 100ms

Experiments

Black box calls vs. arrangement size

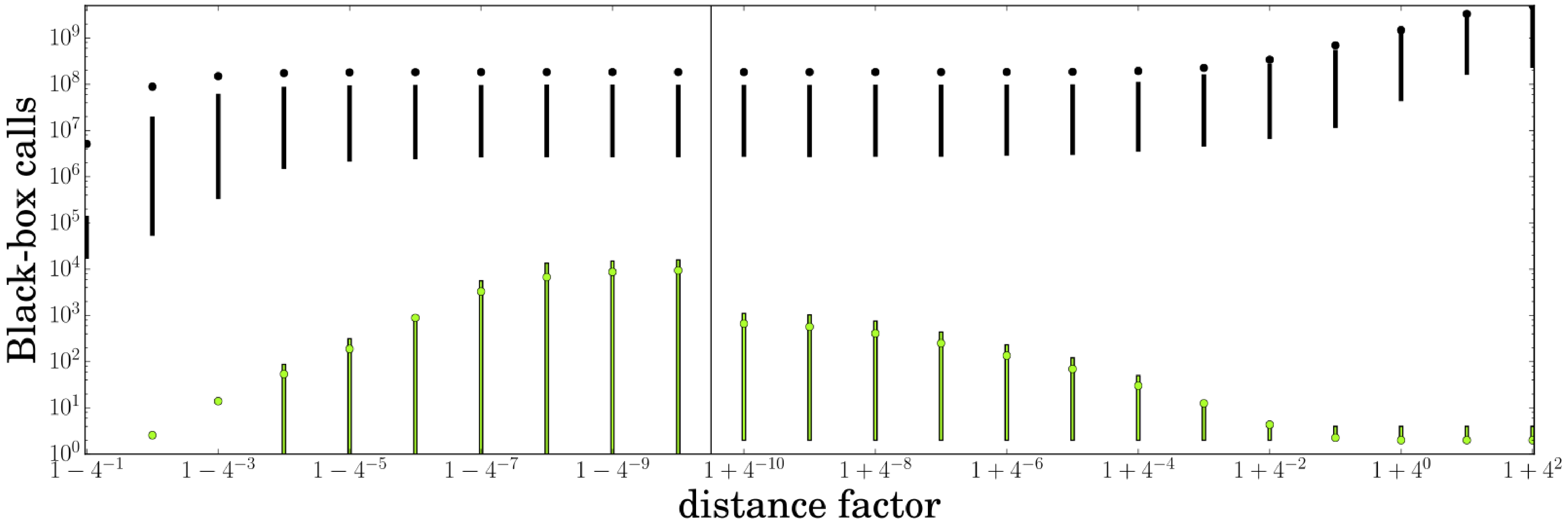
SAME-CHARACTERS:



Experiments

Black box calls vs. arrangement size

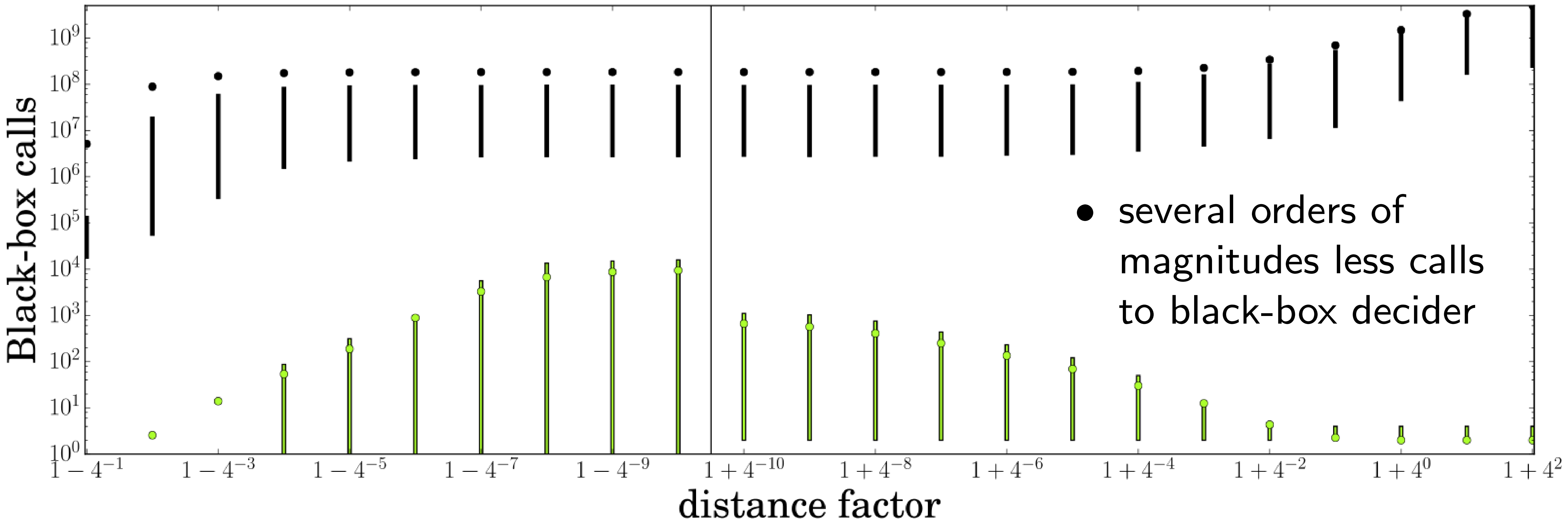
SIGSPATIAL:



Experiments

Black box calls vs. arrangement size

SIGSPATIAL:



- several orders of magnitudes less calls to black-box decider

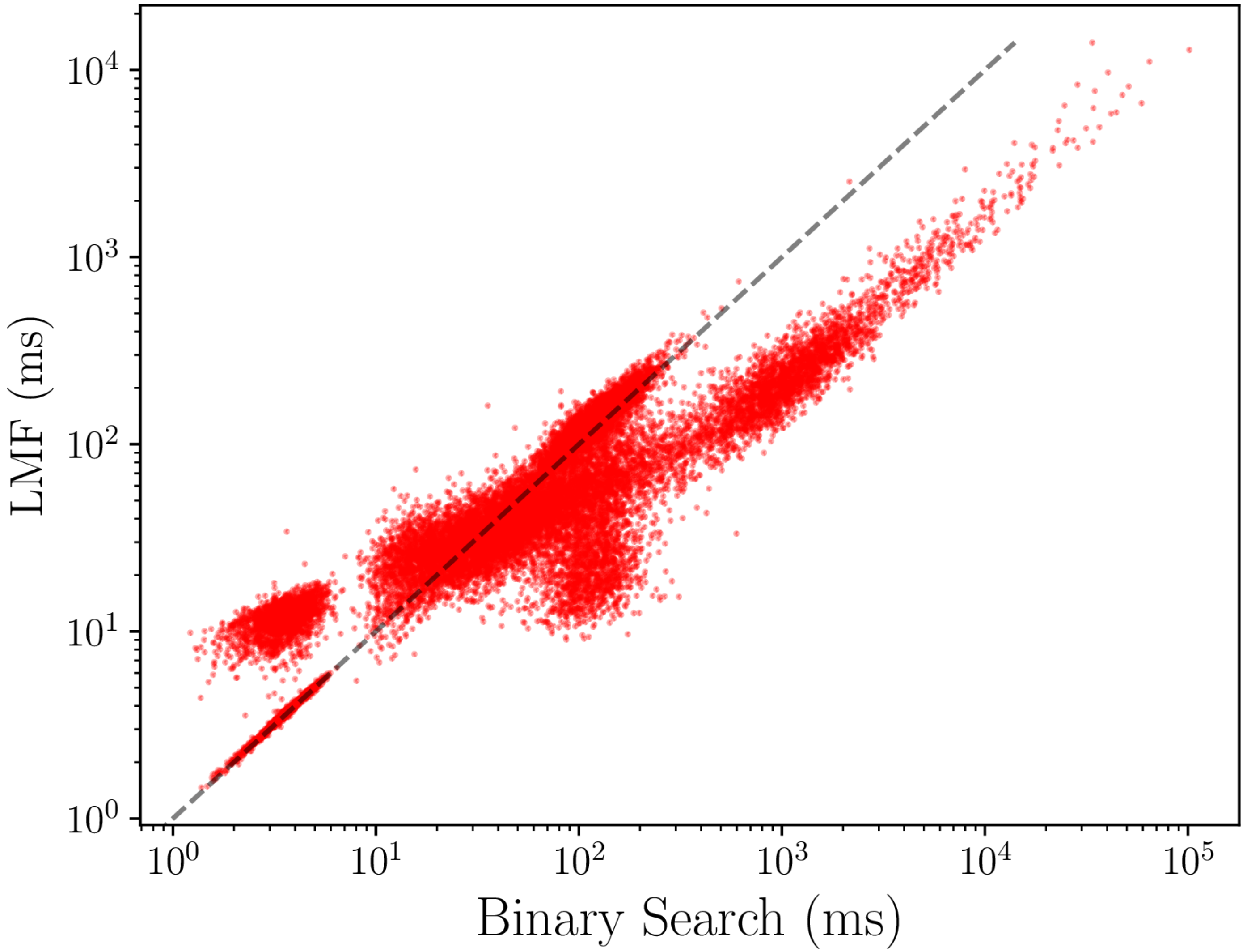
Experiments

Value Computation Times

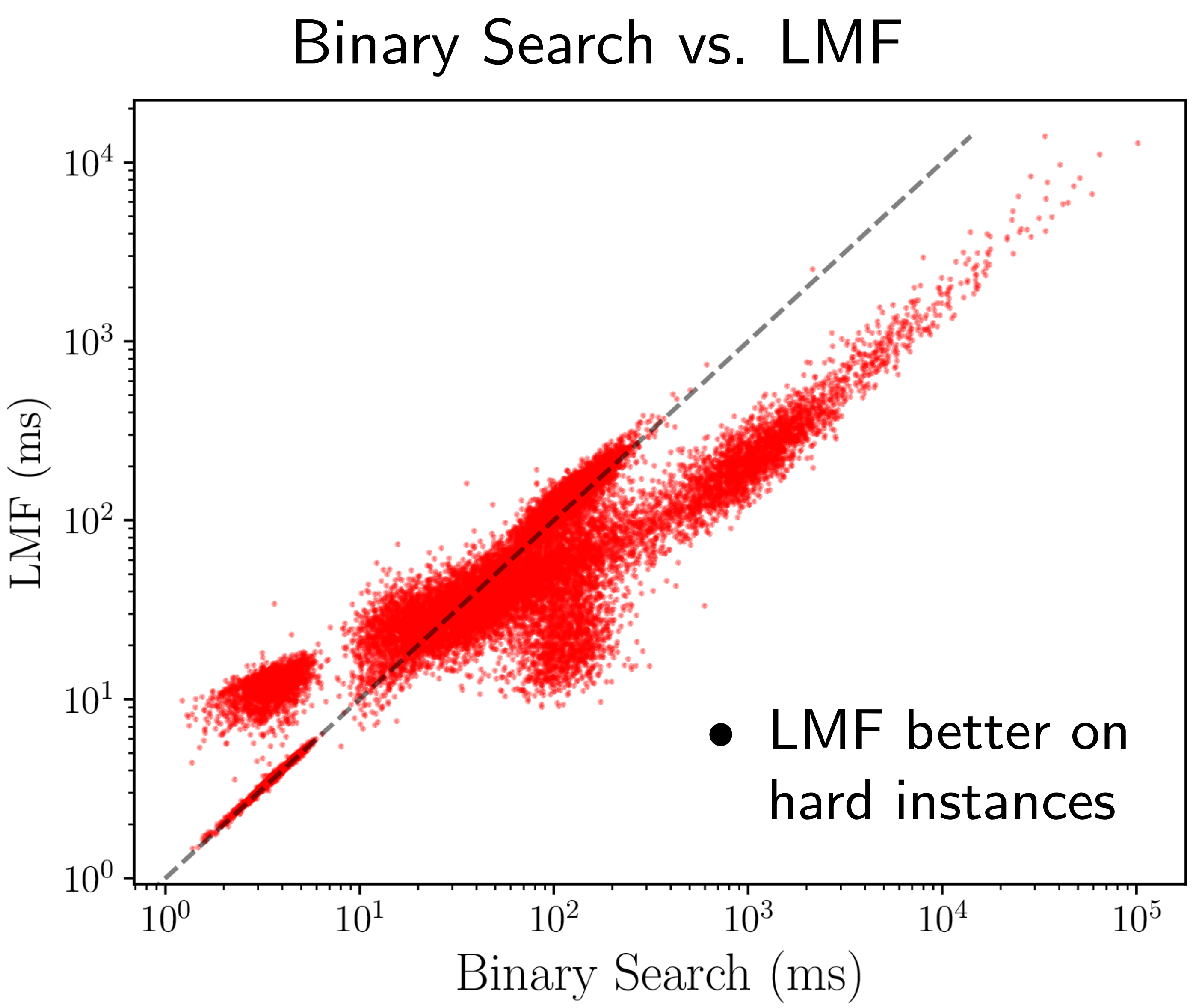
Approach	Time	Black-Box Calls
LMF	148,032 ms (141.0 ms per instance)	13,323,232 (12,688.8 per instance)
Binary Search	536,853 ms (511.3 ms per instance)	45,909,628 (43,723.5 per instance)
Lipschitz-only	4,204,521 ms (4,004.3 ms per instance)	820,468,224 (781,398.3 per instance)

Experiments

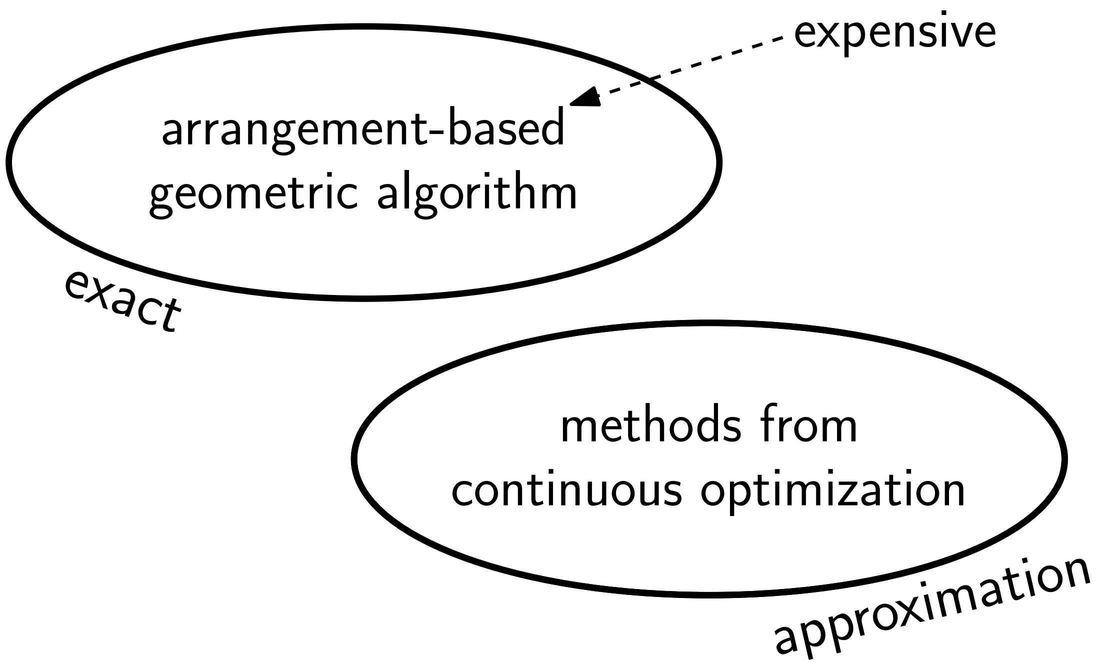
Binary Search vs. LMF



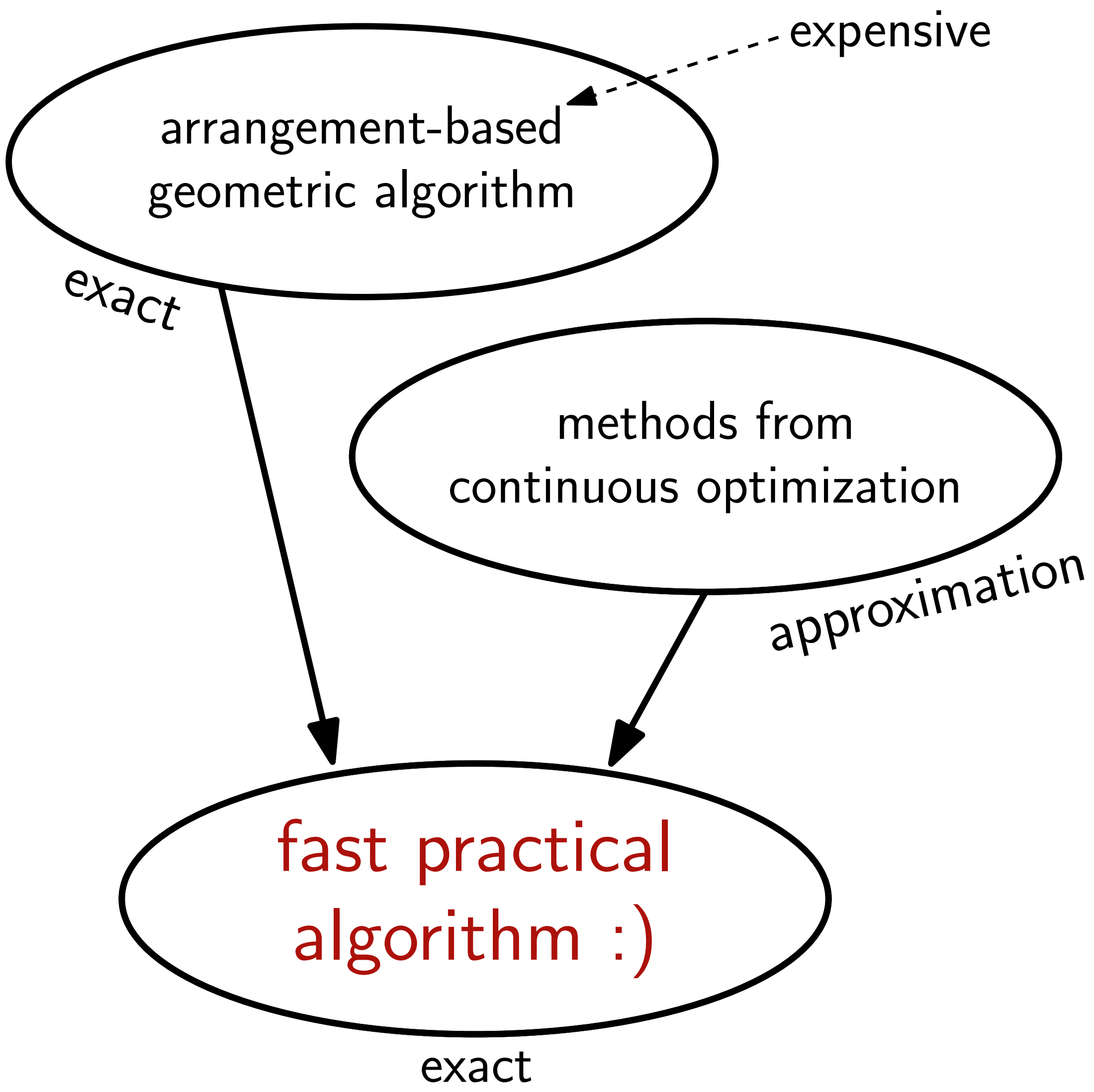
Experiments



Summary

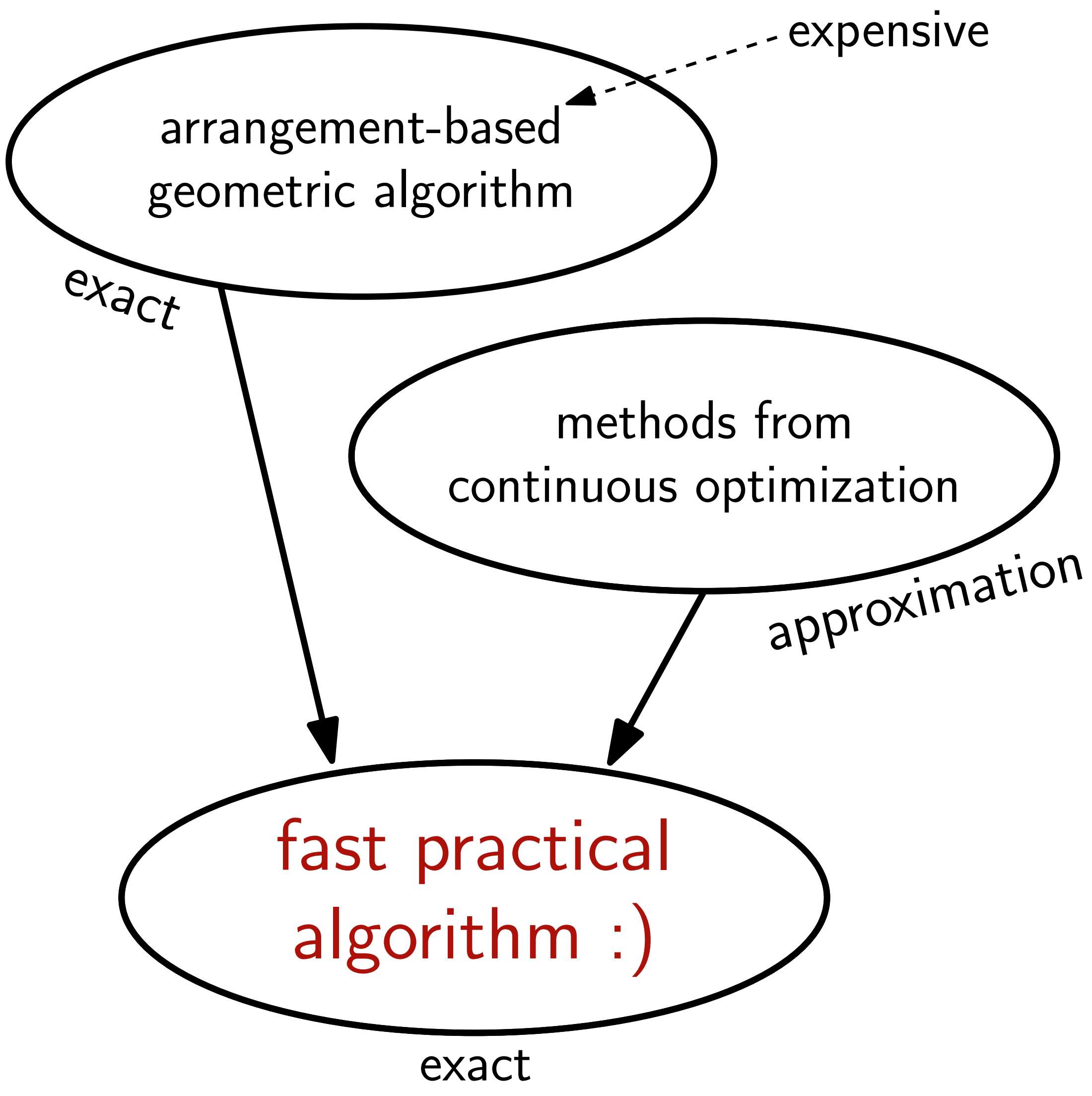

expensive
arrangement-based
geometric algorithm
exact
methods from
continuous optimization
approximation

Summary



expensive
arrangement-based geometric algorithm
exact
methods from continuous optimization
approximation
fast practical algorithm :)
exact

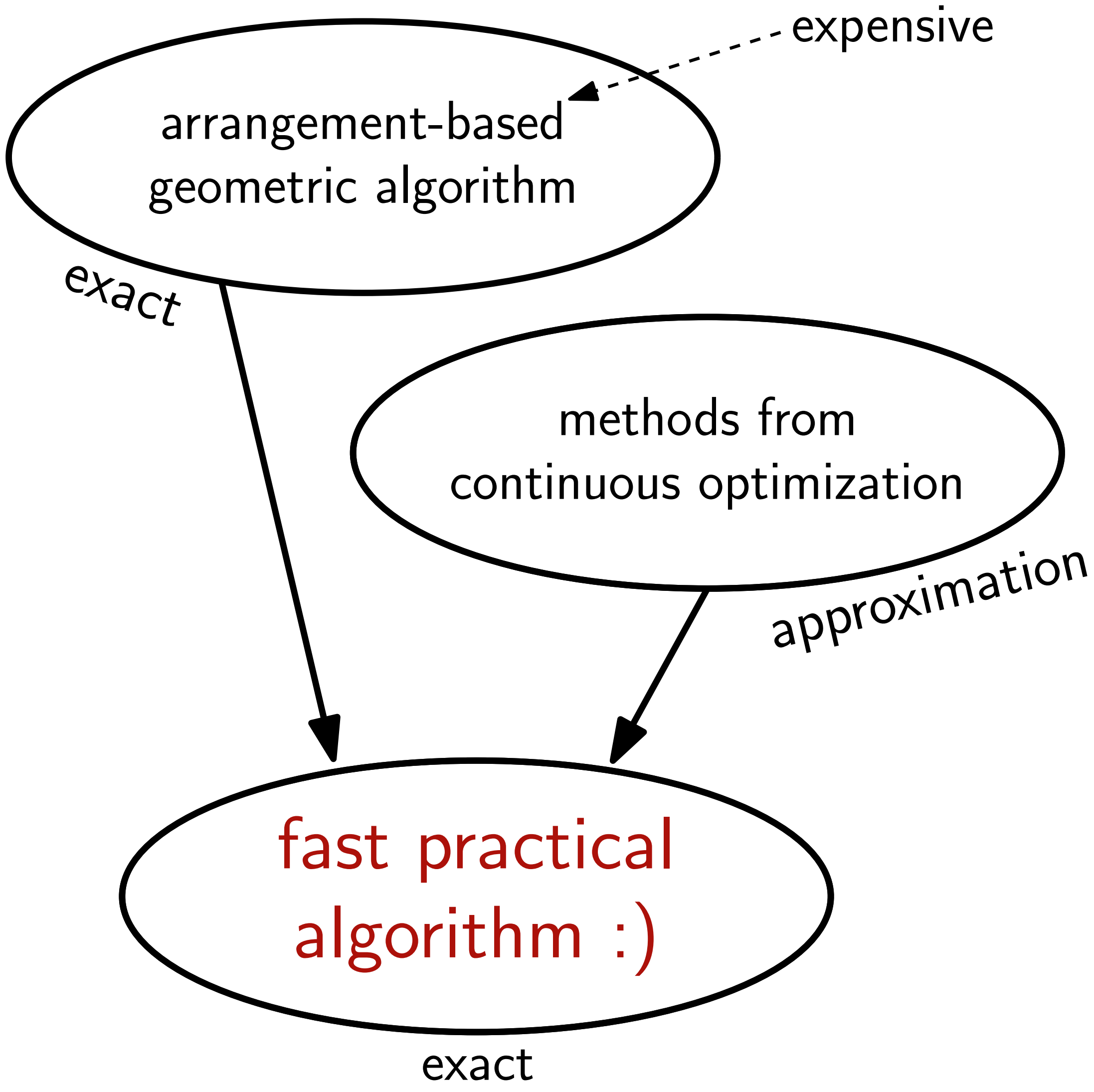
Summary



Future Directions:

- Apply approach to other problems
- Find optimal point of building the arrangement

Summary



Future Directions:

- Apply approach to other problems
- Find optimal point of building the arrangement

Code:
https://gitlab.com/anusser/frechet_distance_under_translation