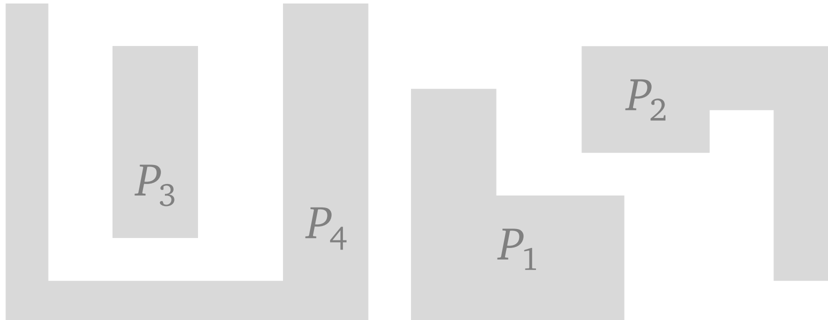
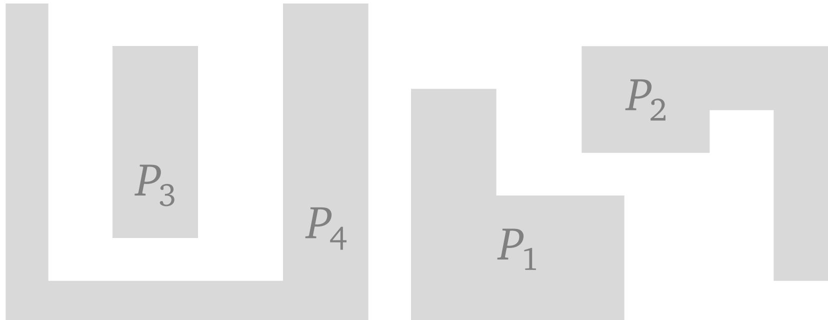


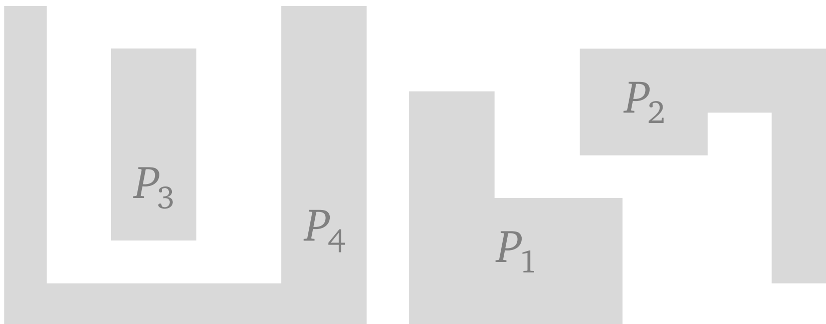
Touring a Sequence of Orthogonal Polygons

Katrin Casel Sándor Kisfaludi-Bak Linda Kleist
Jeroen S.K. Lamme Eunjin Oh Yanheng Wang



Disjoint orthogonal polygons $P_1, \ldots, P_k$ (order is given)

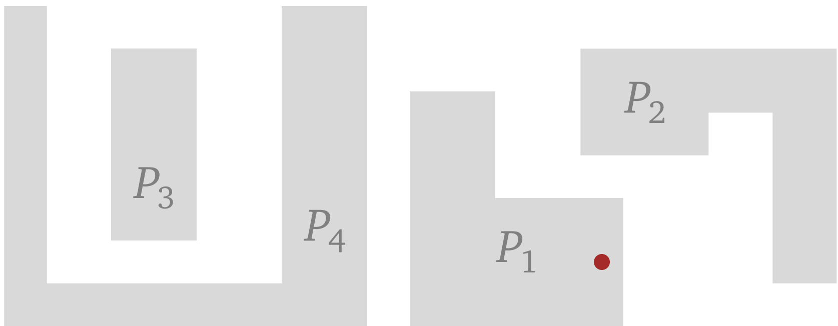
Total number of vertices $= n$



Disjoint orthogonal polygons $P_1, \ldots, P_k$ (order is given)

Total number of vertices $= n$

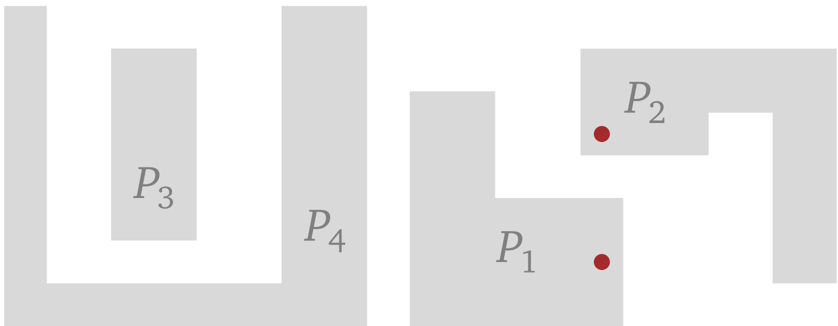
Pick $p_i \in P_i$ to minimize $\|p_1 - p_2\| + \|p_2 - p_3\| + \cdots + \|p_{k-1} - p_k\|$



Disjoint orthogonal polygons $P_1, \ldots, P_k$ (order is given)

Total number of vertices $= n$

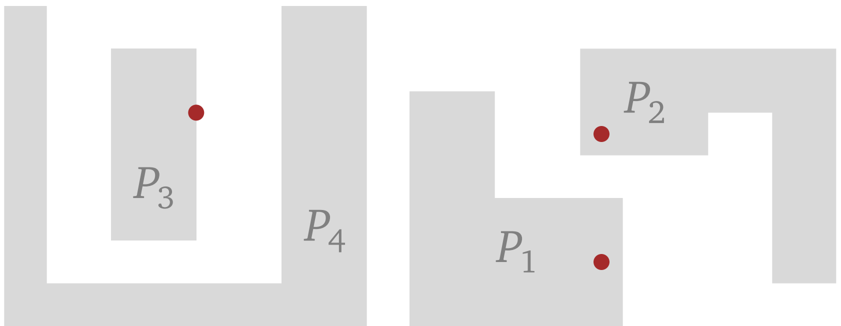
Pick $p_i \in P_i$ to minimize $\|p_1 - p_2\| + \|p_2 - p_3\| + \cdots + \|p_{k-1} - p_k\|$



Disjoint orthogonal polygons $P_1, \ldots, P_k$ (order is given)

Total number of vertices $= n$

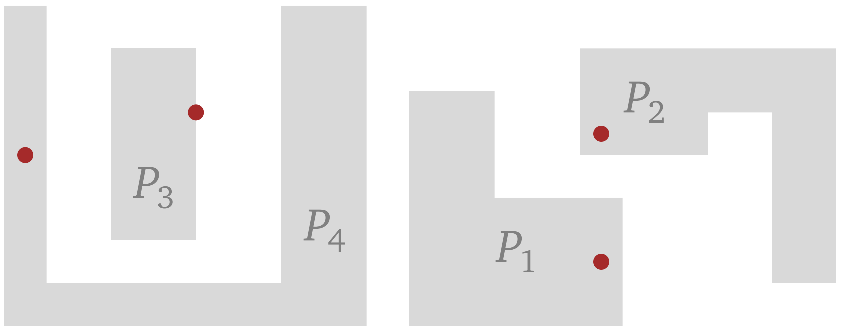
Pick $p_i \in P_i$ to minimize $\|p_1 - p_2\| + \|p_2 - p_3\| + \cdots + \|p_{k-1} - p_k\|$



Disjoint orthogonal polygons $P_1, \ldots, P_k$ (order is given)

Total number of vertices $= n$

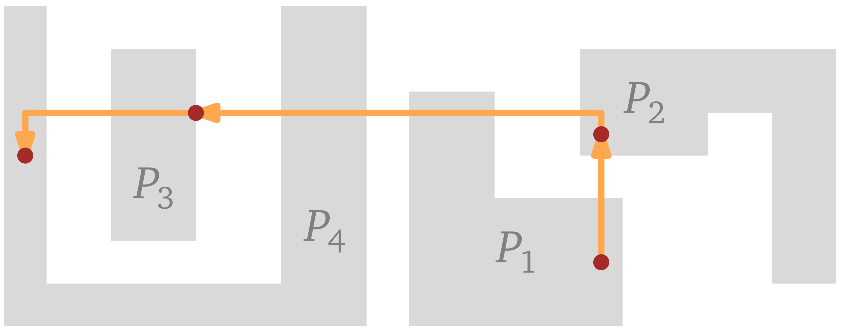
Pick $p_i \in P_i$ to minimize $\|p_1 - p_2\| + \|p_2 - p_3\| + \cdots + \|p_{k-1} - p_k\|$



Disjoint orthogonal polygons $P_1, \ldots, P_k$ (order is given)

Total number of vertices $= n$

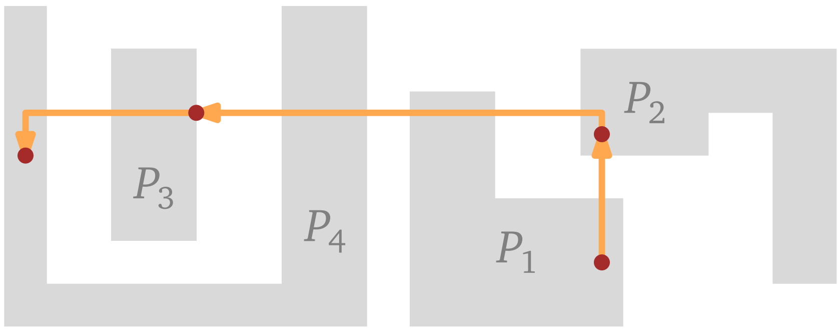
Pick $p_i \in P_i$ to minimize $\|p_1 - p_2\| + \|p_2 - p_3\| + \cdots + \|p_{k-1} - p_k\|$



Disjoint orthogonal polygons $P_1, \ldots, P_k$ (order is given)

Total number of vertices $= n$

Pick $p_i \in P_i$ to minimize $\|p_1 - p_2\| + \|p_2 - p_3\| + \cdots + \|p_{k-1} - p_k\|$

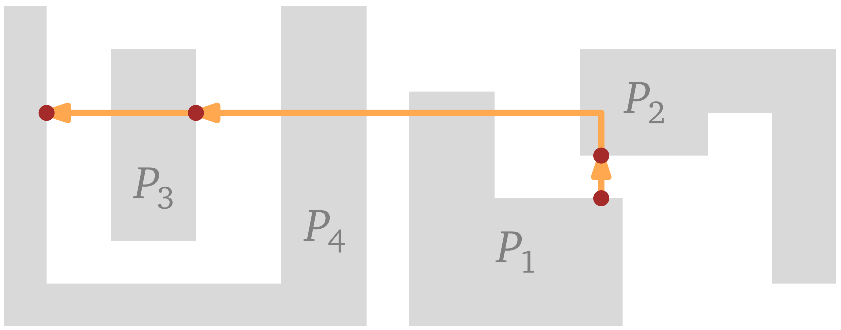


Disjoint orthogonal polygons $P_1, \ldots, P_k$ (order is given)

Total number of vertices $= n$

Pick $p_i \in P_i$ to minimize $\|p_1 - p_2\| + \|p_2 - p_3\| + \cdots + \|p_{k-1} - p_k\|$

may assume $p_i \in \partial P_i$

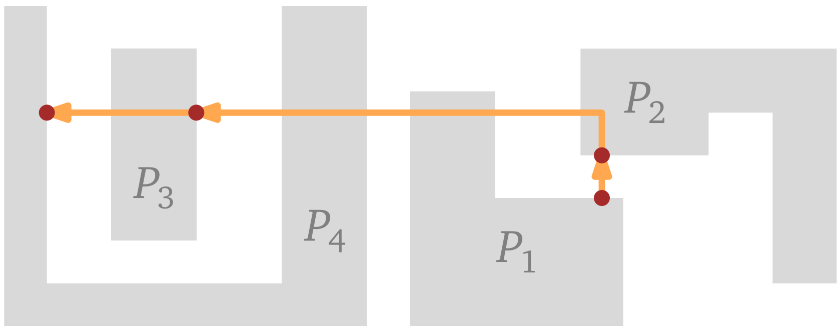


Disjoint orthogonal polygons $P_1, \ldots, P_k$ (order is given)

Total number of vertices $= n$

Pick $p_i \in P_i$ to minimize $\|p_1 - p_2\| + \|p_2 - p_3\| + \cdots + \|p_{k-1} - p_k\|$

may assume $p_i \in \partial P_i$



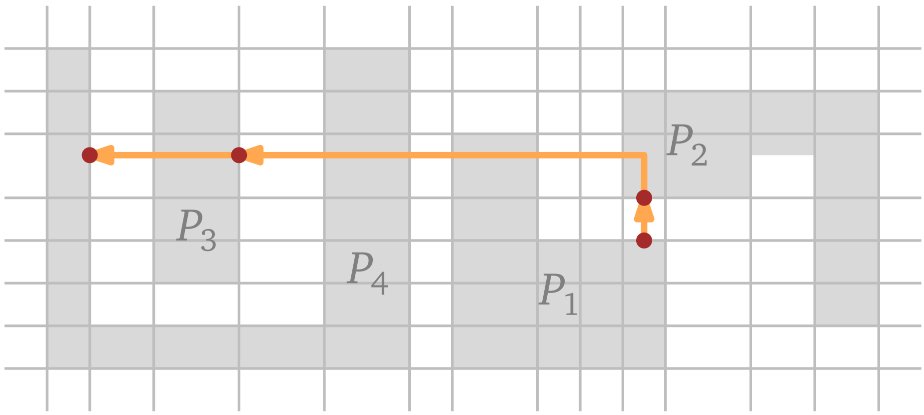
Disjoint orthogonal polygons $P_1, \ldots, P_k$ (order is given)

Total number of vertices $= n$

Pick $p_i \in P_i$ to minimize $\|p_1 - p_2\| + \|p_2 - p_3\| + \cdots + \|p_{k-1} - p_k\|$

may assume $p_i \in \partial P_i$

may snap to the grid



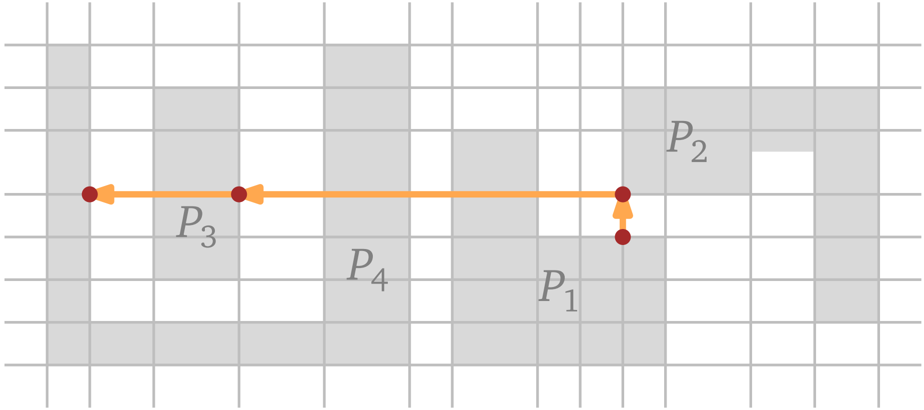
Disjoint orthogonal polygons $P_1, \ldots, P_k$ (order is given)

Total number of vertices $= n$

Pick $p_i \in P_i$ to minimize $\|p_1 - p_2\| + \|p_2 - p_3\| + \cdots + \|p_{k-1} - p_k\|$

may assume $p_i \in \partial P_i$

may snap to the grid



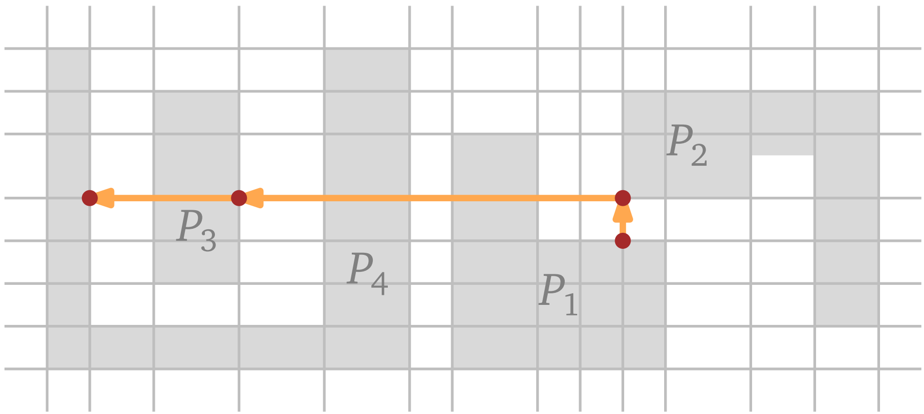
Disjoint orthogonal polygons $P_1, \ldots, P_k$ (order is given)

Total number of vertices $= n$

Pick $p_i \in P_i$ to minimize $\|p_1 - p_2\| + \|p_2 - p_3\| + \cdots + \|p_{k-1} - p_k\|$

may assume $p_i \in \partial P_i$

may snap to the grid



Disjoint orthogonal polygons $P_1, \dots, P_k$ (order is given)

Total number of vertices $= n$

Pick $p_i \in P_i$ to minimize $\|p_1 - p_2\| + \|p_2 - p_3\| + \cdots + \|p_{k-1} - p_k\|$

$$\left.\begin{array}{l}\text{may assume } p_i \in \partial P_i \\ \text{may snap to the grid}\end{array}\right\} \text{portal}_i := \partial P_i \cap \text{grid}$$

Dynamic Programming

$k :=$ #polygons
$n :=$ #vertices in total
$\text{portal}_i := \partial P_i \cap \text{grid}$

Dynamic Programming

$k :=$ #polygons
$n :=$ #vertices in total
$\text{portal}_i := \partial P_i \cap \text{grid}$

Let $f_i(p)$ be the minimum length to visit $P_1, \dots, P_i, p$.

Dynamic Programming

$k :=$ #polygons
$n :=$ #vertices in total
$\text{portal}_i := \partial P_i \cap \text{grid}$

Let $f_i(p)$ be the minimum length to visit $P_1, \ldots, P_i, p$.

$$f_i(p) = \min \left\{ f_{i-1}(q) + \|p - q\| : q \in \text{portal}_{i-1} \right\}$$

Dynamic Programming

$k :=$ #polygons
$n :=$ #vertices in total
$\text{portal}_i := \partial P_i \cap \text{grid}$

Let $f_i(p)$ be the minimum length to visit $P_1, \ldots, P_i, p$.

$$f_i(p) = \min\left\{f_{i-1}(q) + \|p - q\| : q \in \text{portal}_{i-1}\right\}$$

$\widetilde{O}\left(\sum_{i=1}^{k} |\text{portal}_i|\right) = \widetilde{O}(n^2)$ time via Voronoi diagram

Dynamic Programming

$k :=$ #polygons
$n :=$ #vertices in total
$\text{portal}_i := \partial P_i \cap \text{grid}$

Let $f_i(p)$ be the minimum length to visit $P_1, \ldots, P_i, p$.

$$f_i(p) = \min\left\{f_{i-1}(q) + \|p - q\| : q \in \text{portal}_{i-1}\right\}$$

$\widetilde{O}\left(\sum_{i=1}^{k} |\text{portal}_i|\right) = \widetilde{O}(n^2)$ time via Voronoi diagram

tight

Dynamic Programming

$k :=$ #polygons
$n :=$ #vertices in total
$\text{portal}_i := \partial P_i \cap \text{grid}$

Let $f_i(p)$ be the minimum length to visit $P_1, \ldots, P_i, p$.

$$f_i(p) = \min\left\{f_{i-1}(q) + \|p - q\| : q \in \text{portal}_{i-1}\right\}$$

$$\widetilde{O}\left(\sum_{i=1}^{k} |\text{portal}_i|\right) \underset{\text{tight}}{=} \widetilde{O}(n^2) \text{ time via Voronoi diagram}$$

Can we break the quadratic barrier?
(assume each P_i has exactly s vertices)

Dynamic Programming

$k :=$ #polygons
$n :=$ #vertices in total
$\text{portal}_i := \partial P_i \cap \text{grid}$

Let $f_i(p)$ be the minimum length to visit $P_1, \ldots, P_i, p$.

$$f_i(p) = \min\left\{f_{i-1}(q) + \|p - q\| : q \in \text{portal}_{i-1}\right\}$$

$\widetilde{O}\left(\sum_{i=1}^{k} |\text{portal}_i|\right) = \widetilde{O}(n^2)$ time via Voronoi diagram

tight

Can we break the quadratic barrier?
(assume each P_i has exactly s vertices)

– Case $s \leq n^{0.8}$
– Case $s > n^{0.8}$ (and thus $k < n^{0.2}$)

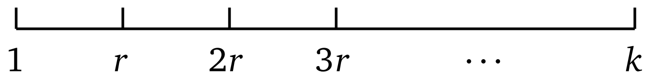
Case $s \leq n^{0.8}$

$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
$\text{portal}_i := \partial P_i \cap \text{grid}$
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

Case $s \le n^{0.8}$

$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
$\text{portal}_i := \partial P_i \cap \text{grid}$
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

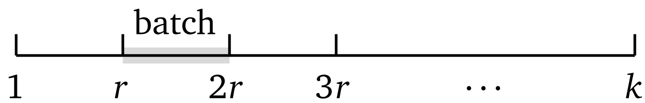
Idea: time localization



Case $s \le n^{0.8}$

$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
$\text{portal}_i := \partial P_i \cap \text{grid}$
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

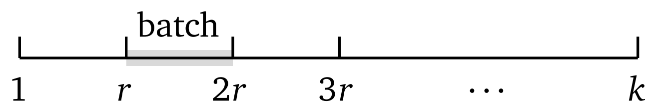
Idea: time localization



Case $s \leq n^{0.8}$

$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
$\text{portal}_i := \partial P_i \cap \text{grid}$
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

Idea: time localization

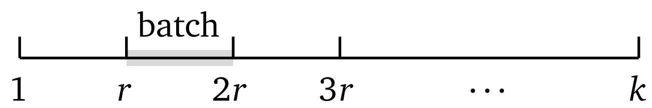


$\text{terminal}_i := \partial P_i \cap \text{grid of the batch}$

Case $s \le n^{0.8}$

$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
$\text{portal}_i := \partial P_i \cap \text{grid}$
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

Idea: time localization



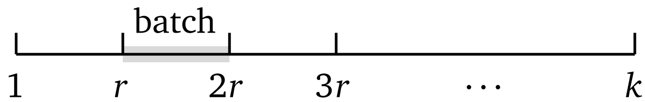
$\text{terminal}_i := \partial P_i \cap$ grid of the batch

$$\sum_{i \in \text{batch}} |\text{terminal}_i| \le (rs)^2$$

Case $s \le n^{0.8}$

$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
$\text{portal}_i := \partial P_i \cap \text{grid}$
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

Idea: time localization



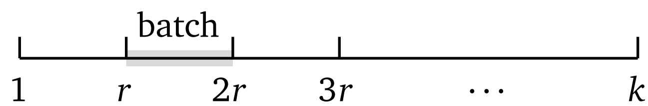
$\text{terminal}_i := \partial P_i \cap \text{grid of the batch}$ $\qquad \sum_{i \in \text{batch}} |\text{terminal}_i| \le (rs)^2$

If shortest path visits terminals only, *then* can DP in time $\widetilde{O}((rs)^2)$.

Case $s \leq n^{0.8}$

$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
$\text{portal}_i := \partial P_i \cap \text{grid}$
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

Idea: time localization

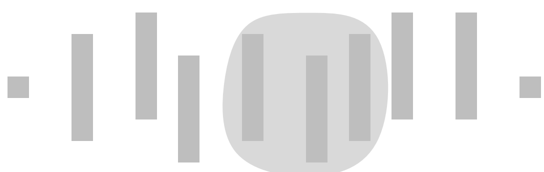


$\text{terminal}_i := \partial P_i \cap$ grid of the batch

$\sum_{i \in \text{batch}} |\text{terminal}_i| \leq (rs)^2$

If shortest path visits terminals only, *then* can DP in time $\widetilde{O}((rs)^2)$.

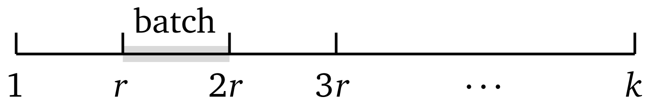
But...



Case $s \le n^{0.8}$

$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
$\text{portal}_i := \partial P_i \cap \text{grid}$
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

Idea: time localization

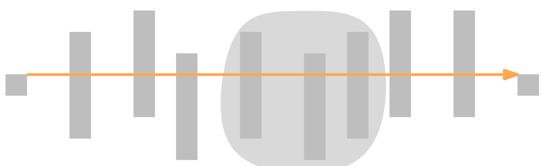


$\text{terminal}_i := \partial P_i \cap \text{grid of the batch}$

$\sum_{i \in \text{batch}} |\text{terminal}_i| \le (rs)^2$

If shortest path visits terminals only, *then* can DP in time $\widetilde{O}((rs)^2)$.

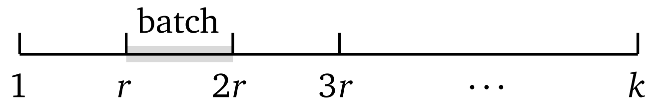
But...



Case $s \leq n^{0.8}$

$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
$\text{portal}_i := \partial P_i \cap \text{grid}$
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

Idea: time localization



$\text{terminal}_i := \partial P_i \cap \text{grid of the batch}$ $\quad \sum_{i \in \text{batch}} |\text{terminal}_i| \leq (rs)^2$

If shortest path visits terminals only, *then* can DP in time $\widetilde{O}((rs)^2)$.

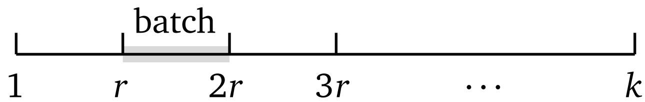
Lemma. Fix $a = jr$ and $p \in \text{portal}_{a+r}$. There is a shortest path visiting $P_1, \ldots, P_{a+r}, p$ with one of the following properties:

- it visits $\text{portal}_1, \ldots, \text{portal}_{a-1}, \text{terminal}_a, \ldots, \text{terminal}_{a+r}, p$;
- it visits $\text{portal}_1, \ldots, \text{portal}_{a-1}$ and is horizontal afterwards;
- it visits $\text{portal}_1, \ldots, \text{portal}_{a-1}$ and is vertical afterwards.

Case $s \le n^{0.8}$

$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
$\text{portal}_i := \partial P_i \cap \text{grid}$
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

Idea: time localization



$\text{terminal}_i := \partial P_i \cap$ grid of the batch

$\sum_{i \in \text{batch}} |\text{terminal}_i| \le (rs)^2$

If shortest path visits terminals only, *then* can DP in time $\widetilde{O}((rs)^2)$.

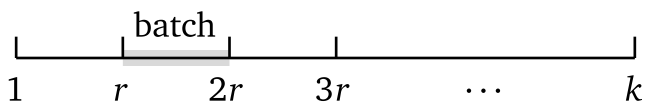
Lemma. Fix $a = jr$ and $p \in \text{portal}_{a+r}$. There is a shortest path visiting $P_1, \ldots, P_{a+r}, p$ with one of the following properties:

- ✓ it visits $\text{portal}_1, \ldots, \text{portal}_{a-1}, \text{terminal}_a, \ldots, \text{terminal}_{a+r}, p$;
- it visits $\text{portal}_1, \ldots, \text{portal}_{a-1}$ and is horizontal afterwards;
- it visits $\text{portal}_1, \ldots, \text{portal}_{a-1}$ and is vertical afterwards.

Case $s \leq n^{0.8}$

$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
portal$_i := \partial P_i \cap$ grid
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

Idea: time localization



terminal$_i := \partial P_i \cap$ grid of the batch $\qquad \sum_{i \in \text{batch}} |\text{terminal}_i| \leq (rs)^2$

If shortest path visits terminals only, *then* can DP in time $\widetilde{O}((rs)^2)$.

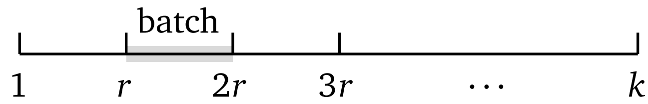
Lemma. Fix $a = jr$ and $p \in \text{portal}_{a+r}$. There is a shortest path visiting $P_1, \ldots, P_{a+r}, p$ with one of the following properties:

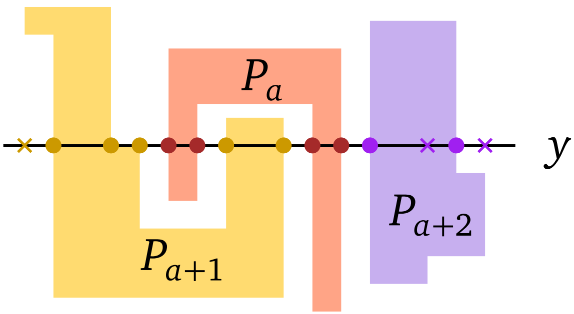
✓ – it visits portal$_1, \ldots,$ portal$_{a-1}$, terminal$_a, \ldots,$ terminal$_{a+r}$, p;
? – it visits portal$_1, \ldots,$ portal$_{a-1}$ and is horizontal afterwards;
? – it visits portal$_1, \ldots,$ portal$_{a-1}$ and is vertical afterwards.

Case $s \leq n^{0.8}$

$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
$\text{portal}_i := \partial P_i \cap \text{grid}$
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

Idea: time localization

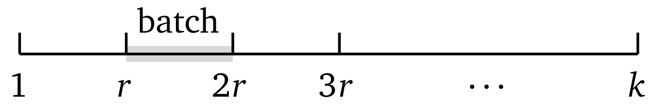


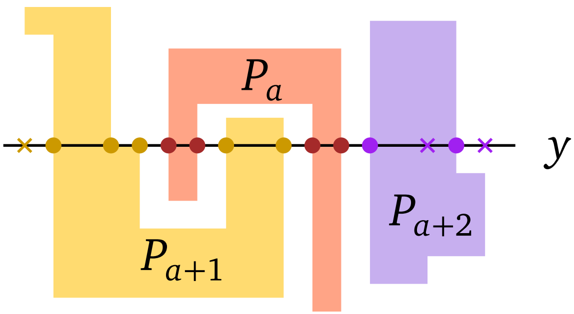


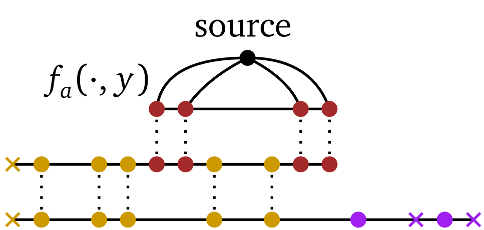
Case $s \leq n^{0.8}$

$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
$\text{portal}_i := \partial P_i \cap \text{grid}$
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

Idea: time localization



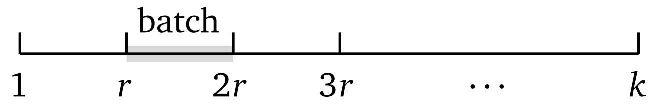


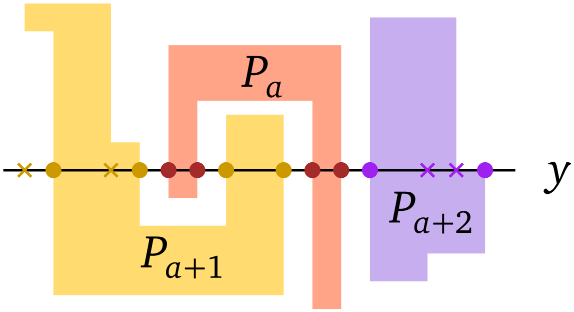


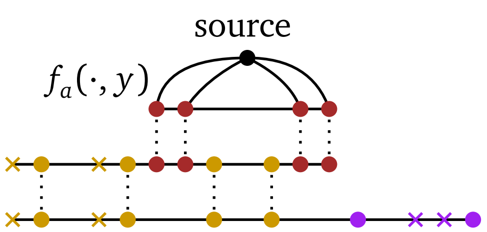
Case $s \leq n^{0.8}$

$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
$\text{portal}_i := \partial P_i \cap \text{grid}$
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

Idea: time localization



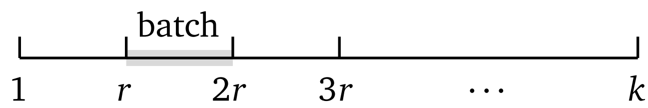


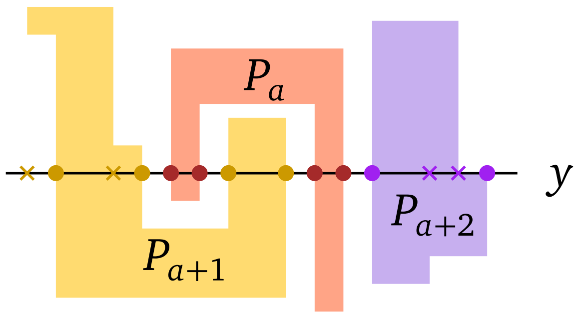


Case $s \leq n^{0.8}$

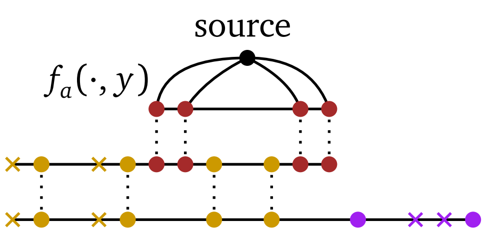
$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
$\text{portal}_i := \partial P_i \cap \text{grid}$
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

Idea: time localization





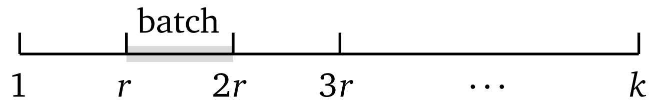
Sweep the line for n steps

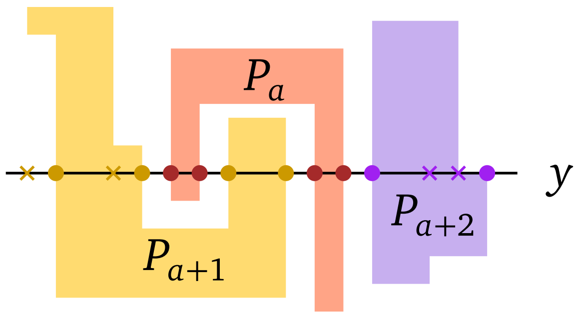


Case $s \leq n^{0.8}$

$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
$\text{portal}_i := \partial P_i \cap \text{grid}$
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

Idea: time localization



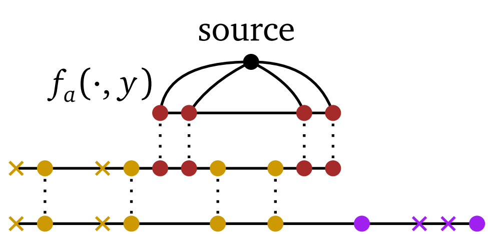


Sweep the line for n steps

#insertions/deletions $= O(rs)$

#weight updates $= O(ns)$

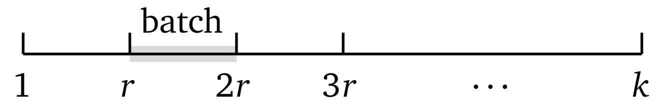
#queries $= O(ns)$

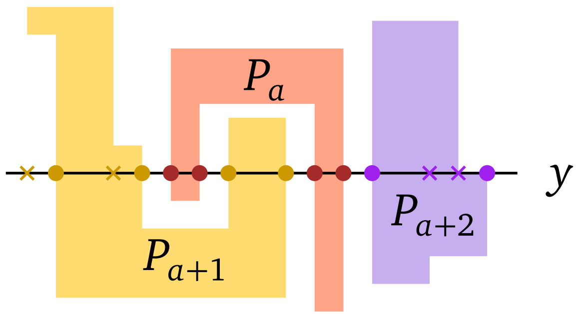


Case $s \leq n^{0.8}$

$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
$\text{portal}_i := \partial P_i \cap \text{grid}$
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

Idea: time localization



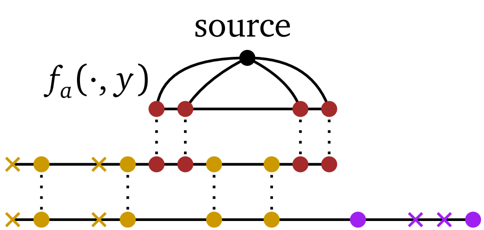


Sweep the line for n steps

#insertions/deletions $= O(rs)$

#weight updates $= O(ns)$

#queries $= O(ns)$



Dynamic planar SSSP

[Das, Probst Gutenberg, Wulff-Nilsen 2022]

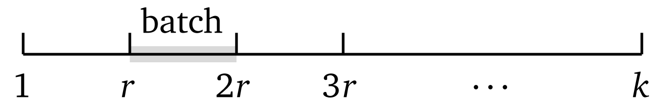
update time $\widetilde{O}(N^{1/2})$

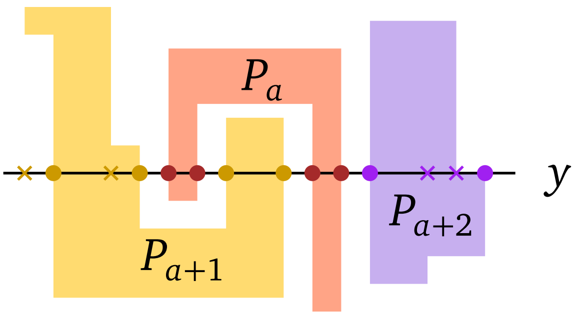
query time $\widetilde{O}(N^{1/2})$

Case $s \leq n^{0.8}$

$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
$\text{portal}_i := \partial P_i \cap \text{grid}$
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

Idea: time localization



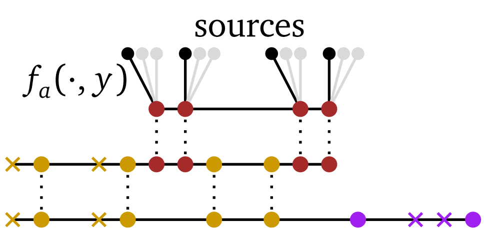


Sweep the line for n steps

#insertions/deletions $= O(rs)$

~~#weight updates $= O(ns)$~~

#queries $= O(ns)$



Dynamic planar MSSP

[Charalampopoulos, Karczmarz 2020]

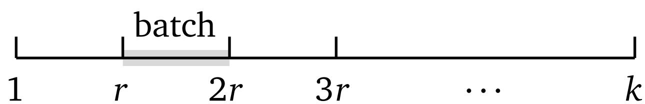
update time $\widetilde{O}(N^{4/5} + N^{3/4}s^{1/4})$

query time $\widetilde{O}(1)$

Case $s \leq n^{0.8}$

$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
$\text{portal}_i := \partial P_i \cap \text{grid}$
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

Idea: time localization



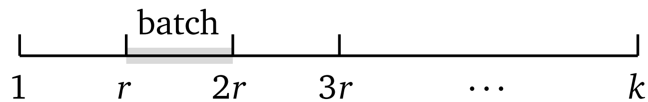
Running time:

$$\text{each batch } \widetilde{O}\Big(\underbrace{(rs)^2}_{\text{DP}} + \underbrace{n^{9/5} + n^{7/4}s^{1/4} + ns}_{\text{line sweep}}\Big)$$

Case $s \leq n^{0.8}$

$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
$\text{portal}_i := \partial P_i \cap \text{grid}$
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

Idea: time localization



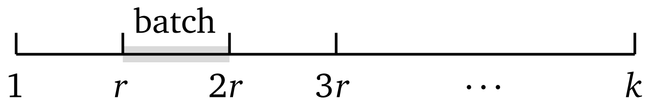
Running time:

$$\text{each batch } \widetilde{O}\Big(\underbrace{(rs)^2}_{\text{DP}} + \underbrace{n^{9/5} + n^{7/4}s^{1/4} + ns}_{\text{line sweep}}\Big) \;\; \leq \underset{\text{choose } r := n^{0.98}/s}{\widetilde{O}(n^{1.96})}$$

Case $s \leq n^{0.8}$

$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
$\text{portal}_i := \partial P_i \cap \text{grid}$
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

Idea: time localization



Running time:

each batch $\widetilde{O}\Big(\underbrace{(rs)^2}_{\text{DP}} + \underbrace{n^{9/5} + n^{7/4}s^{1/4} + ns}_{\text{line sweep}}\Big) \leq \widetilde{O}(n^{1.96})$

choose $r := n^{0.98}/s$

overall $\widetilde{O}(n^{1.96}) \cdot n^{0.02} = \widetilde{O}(n^{1.98})$

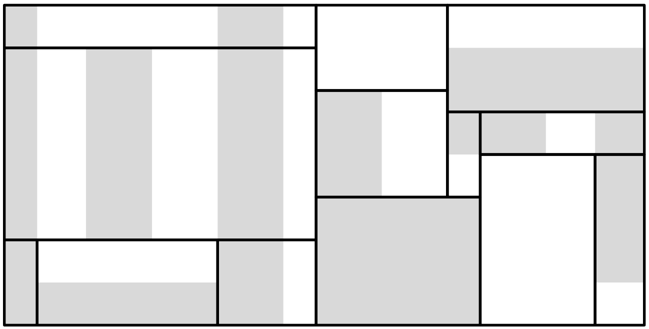
Case $s > n^{0.8}$

$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
$\text{portal}_i := \partial P_i \cap \text{grid}$
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

Case $s > n^{0.8}$

$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
$\text{portal}_i := \partial P_i \cap \text{grid}$
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

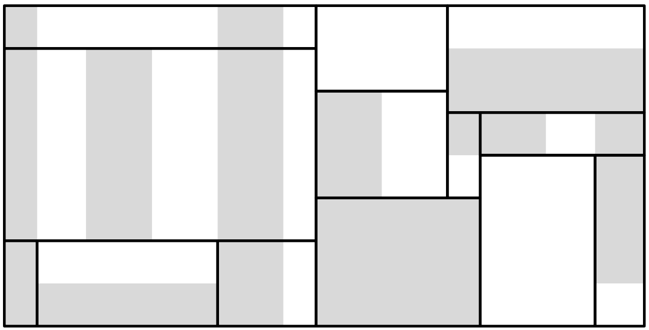
Idea: space localization



Case $s > n^{0.8}$

$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
$\text{portal}_i := \partial P_i \cap \text{grid}$
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

Idea: space localization



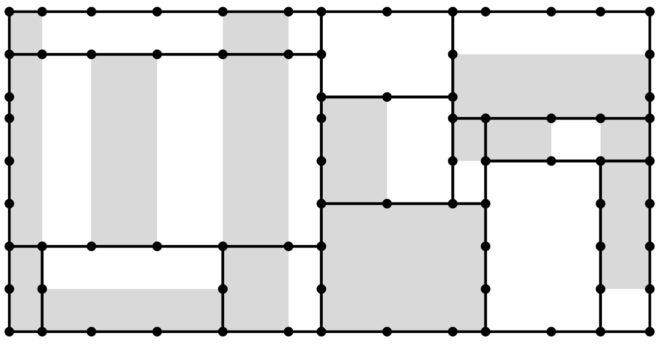
rectangle partition with small stabbing number

[De Berg, Van Krevald 1993]

Case $s > n^{0.8}$

$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
$\text{portal}_i := \partial P_i \cap \text{grid}$
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

Idea: space localization



rectangle partition with small stabbing number

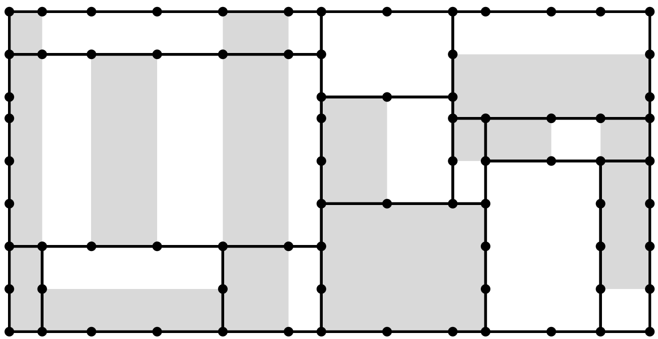
[De Berg, Van Krevald 1993]

"hubs"

Case $s > n^{0.8}$

$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
$\text{portal}_i := \partial P_i \cap \text{grid}$
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

Idea: space localization



rectangle partition with small stabbing number

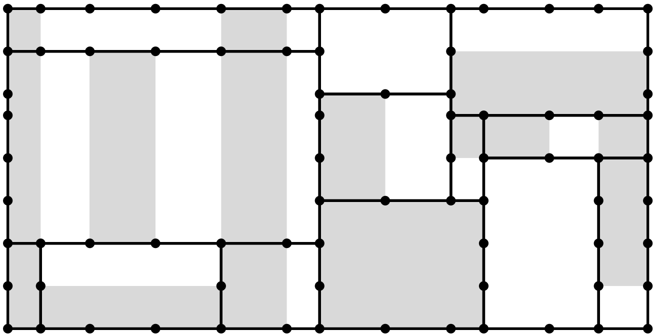
[De Berg, Van Krevald 1993]

#“hubs” $= O(n^{1.5})$

Case $s > n^{0.8}$

$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
$\text{portal}_i := \partial P_i \cap \text{grid}$
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

Idea: space localization



rectangle partition with small stabbing number

[De Berg, Van Krevald 1993]

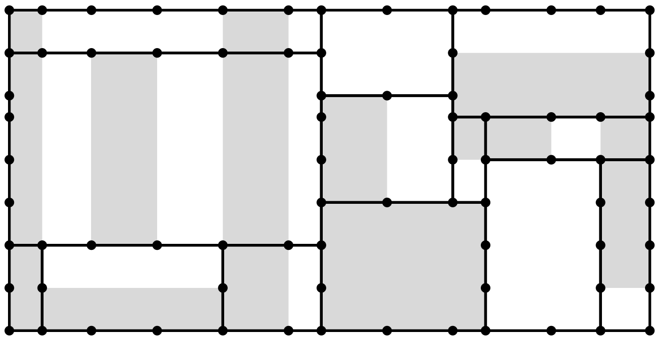
#"hubs" $= O(n^{1.5})$

precompute costs inside rectangles, then DP on hubs

Case $s > n^{0.8}$

$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
$\text{portal}_i := \partial P_i \cap \text{grid}$
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

Idea: space localization



rectangle partition with small stabbing number

[De Berg, Van Krevald 1993]

#"hubs" $= O(n^{1.5})$

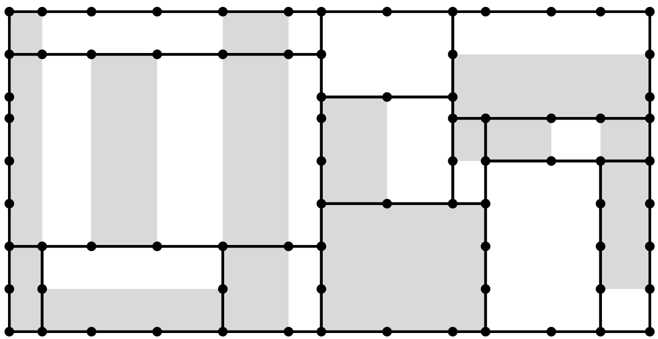
precompute costs inside rectangles, then DP on hubs

running time $= \widetilde{O}(n^{1.5}k^2)$

Case $s > n^{0.8}$

$k :=$ #polygons
$s :=$ #vertices of each polygon
$n := ks$
$\text{portal}_i := \partial P_i \cap \text{grid}$
$f_i(p)$: $P_1 \to \cdots \to P_i \to p$

Idea: space localization



rectangle partition with small stabbing number

[De Berg, Van Krevald 1993]

#"hubs" $= O(n^{1.5})$

precompute costs inside rectangles, then DP on hubs

running time $= \widetilde{O}(n^{1.5}k^2) \le \widetilde{O}(n^{1.9})$

What Have We Shown?

In the sparse case $s \le n^{0.8}$, can solve in time $\widetilde{O}(n^{1.98})$
In the dense case $s > n^{0.8}$, can solve in time $\widetilde{O}(n^{1.9})$

What Have We Shown?

In the sparse case $s \leq n^{0.8}$, can solve in time $\widetilde{O}(n^{1.98})$
In the dense case $s > n^{0.8}$, can solve in time $\widetilde{O}(n^{1.9})$

Hence: an $\widetilde{O}(n^{1.98})$ algorithm for disjoint polygons with the same number of vertices

What Have We Shown?

In the sparse case $s \leq n^{0.8}$, can solve in time $\widetilde{O}(n^{1.98})$
In the dense case $s > n^{0.8}$, can solve in time $\widetilde{O}(n^{1.9})$

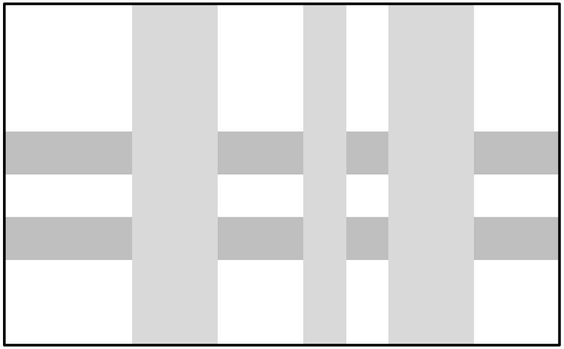
Hence: an $\widetilde{O}(n^{1.98})$ algorithm for disjoint polygons with the same number of vertices

step-disjoint

What Have We Shown?

In the sparse case $s \leq n^{0.8}$, can solve in time $\widetilde{O}(n^{1.98})$
In the dense case $s > n^{0.8}$, can solve in time $\widetilde{O}(n^{1.9})$

Hence: an $\widetilde{O}(n^{1.98})$ algorithm for disjoint polygons with the same number of vertices
step-disjoint



What Have We Shown?

In the sparse case $s \leq n^{0.8}$, can solve in time $\widetilde{O}(n^{1.98})$
In the dense case $s > n^{0.8}$, can solve in time $\widetilde{O}(n^{1.9})$

Hence: an $\widetilde{O}(n^{1.98})$ algorithm for disjoint polygons with the same number of vertices

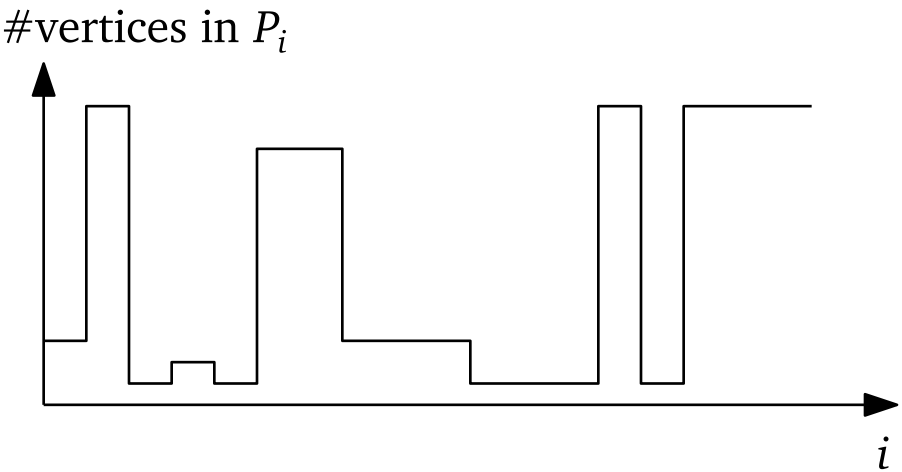
varying

What Have We Shown?

In the sparse case $s \le n^{0.8}$, can solve in time $\widetilde{O}(n^{1.98})$
In the dense case $s > n^{0.8}$, can solve in time $\widetilde{O}(n^{1.9})$

Hence: an $\widetilde{O}(n^{1.98})$ algorithm for disjoint polygons with the same number of vertices

varying

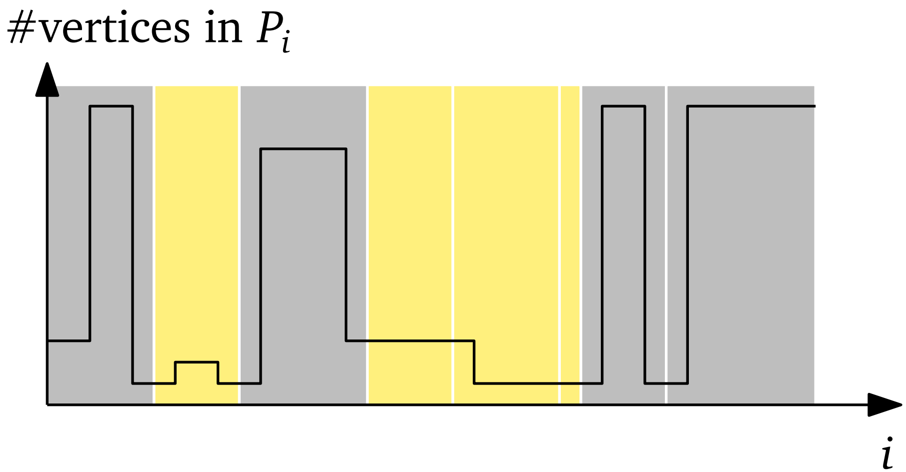


What Have We Shown?

In the sparse case $s \leq n^{0.8}$, can solve in time $\widetilde{O}(n^{1.98})$
In the dense case $s > n^{0.8}$, can solve in time $\widetilde{O}(n^{1.9})$

Hence: an $\widetilde{O}(n^{1.98})$ algorithm for disjoint polygons with the same
number of vertices

varying



What Have We Shown?

In the sparse case $s \leq n^{0.8}$, can solve in time $\widetilde{O}(n^{1.98})$
In the dense case $s > n^{0.8}$, can solve in time $\widetilde{O}(n^{1.9})$

Hence: an $\widetilde{O}(n^{1.98})$ algorithm for disjoint polygons with the same number of vertices

step-disjoint varying

What Have We Shown?

In the sparse case $s \le n^{0.8}$, can solve in time $\widetilde{O}(n^{1.98})$
In the dense case $s > n^{0.8}$, can solve in time $\widetilde{O}(n^{1.9})$

Hence: an $\widetilde{O}(n^{1.98})$ algorithm for disjoint polygons with the same number of vertices

step-disjoint varying

Open Problems

Intersecting polygons?

Faster dynamic planar MSSP?

Superlinear lower bound?

Euclidean distance?

$\widetilde{O}(n^2)$ time for disjoint convex polygons [Dror, Efrat, Lubiw, Mitchell 2003]