Online Motion Planning MA-INF 1314
Application Search Path Approx.!

Elmar Langetepe
University of Bonn

Rep.: Search ratio approximation

- *Competitive ratio* : $C := \sup_{\mathcal{E}} \sup_{p \in \mathcal{G}} \frac{|\mathcal{A}(s,p)|}{|\mathsf{sp}(s,p)|}$

- *Search ratio:* $\mathsf{SR}(\mathcal{A},\mathcal{E}) := \sup_{p \in \mathcal{G}} \frac{|\mathcal{A}(s,p)|}{|\mathsf{sp}(s,p)|}$

- *Optimal search ratio*: $\mathsf{SR}_{\mathsf{OPT}}(\mathcal{E}) := \inf_{\mathcal{A}} \mathsf{SR}(\mathcal{A},\mathcal{E})$

- Approximation: $\mathcal{A}$ *search-competitiv*

$$C_s := \sup_{\mathcal{E}} \frac{\mathsf{SR}(\mathcal{A},\mathcal{E})}{\mathsf{SR}_{\mathsf{OPT}}(\mathcal{E})}$$

- Comparison not against SP, but against best possible SR

Rep.: Non approximation results: Theorem

No constant approximation of the search ratio! Graphs, no vision!

1. Planar graph $G = (V, E)$ multiple edges, goal set V.
2. General graph $G = (V, E)$ goal set V.
3. Directed graph $G = (V, E)$ goal set E and V. (Exercise!)

Counter examples, lower bound! Blackboard!

Rep.: Searching with vision!

Problem: Return path from $\mathrm{last}(d)$ to s has length $\leq d$, might be false! But: sp$(\mathrm{last}(d), s) \leq |\pi_{\mathrm{OPT}s}^{\mathrm{last}(d)}|$

Theorem:

- Roboter with vision
- Environment $\mathcal{E}$
- $\mathrm{Expl}:$ C_β-competitive, β-depth restrictable, Online Explorationstrategy for $\mathcal{E}$ (i.e. $|\mathrm{Expl}\,(d)| \leq C_\beta \cdot |\mathrm{Expl}_{\mathrm{OPT}}(\beta \cdot d)|$)

$\Rightarrow$ Algorithm gives $8\beta C_\beta$-Approximation of optimal search ratio.

Rep.: Proof of the Theorem

$$\mathsf{SR}(\Pi_{\text{opt}}) \geq \frac{\left|\pi_{\text{OPT}s}^{\text{last}(d)}\right|}{d} \geq \frac{|\Pi_{\text{Expl}_{\text{opt}}(d)}|}{2d} \Leftrightarrow |\Pi_{\text{Expl}_{\text{opt}}(d)}| \leq 2d \cdot \mathsf{SR}(\Pi_{\text{opt}})$$

Ratio against search path:

$$\begin{aligned} \frac{\sum_{i=1}^{j+1} |\Pi_{\text{Expl}_{\text{onl}}(2^i)}|}{2^j} &\leq C_\beta \cdot \frac{\sum_{i=1}^{j+1} |\Pi_{\text{Expl}_{\text{opt}}(\beta 2^i)}|}{2^j} \leq 2C_\beta \cdot \frac{\sum_{i=1}^{j+1} \beta 2^i\, \mathsf{SR}(\Pi_{\text{opt}})}{2^j} \\ &\leq 8\beta C_\beta \cdot \mathsf{SR}(\Pi_{\text{opt}})\,. \end{aligned}$$

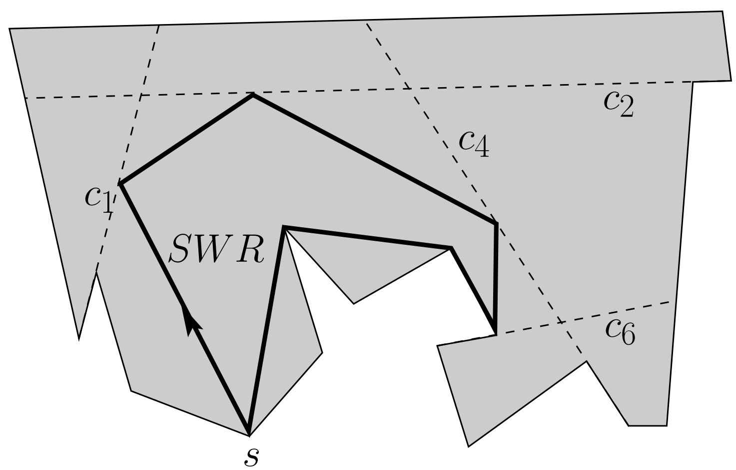
Rep.: Applications!

- Simple polygon, Offline: SWR ($C_\beta = 1 = \beta$)
  $\Rightarrow$ 8-Approximation
- Rectilinear Polygons, Online: Greedy-Online ($C_\beta = \sqrt{2}, \beta = 1$)
  $\Rightarrow 8\sqrt{2}$-Approximation
- Simple Polygons, Online: PolyExplore ($C_\beta = 26, \beta = 1$)
  $\Rightarrow$ 212-Approximation

Consider exploration task! Full and depth restricted!

Rep.: Simple Polygon Offline

- Optimal exploration tour
- Agent with vision, start point s, boundary
- Polygon is fully known
- Depth restriction
- First: General approach. Then: Depth restriction!
- Monotone, rectlilinear, general!



Rep.: Visit essential cuts! Def.

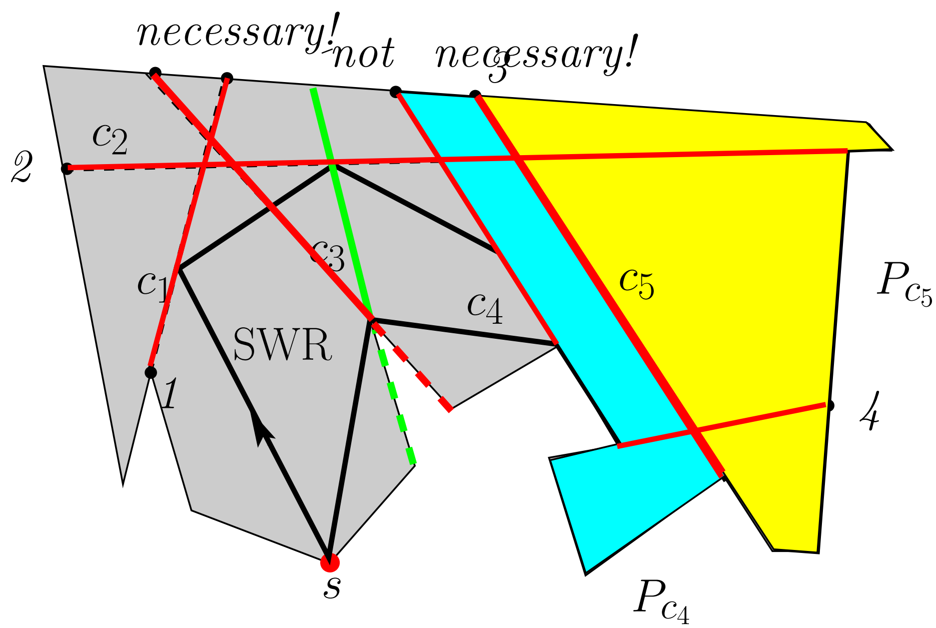
a) (Cuts) Extension of reflex vertex

b) Necessary cuts (w.r.t. s)

c) Dominance-Relationsship $P_{c_i} \subseteq P_{c_j}$

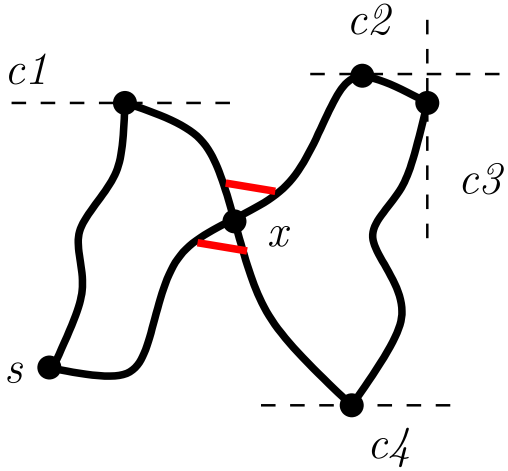
d) Essential cuts

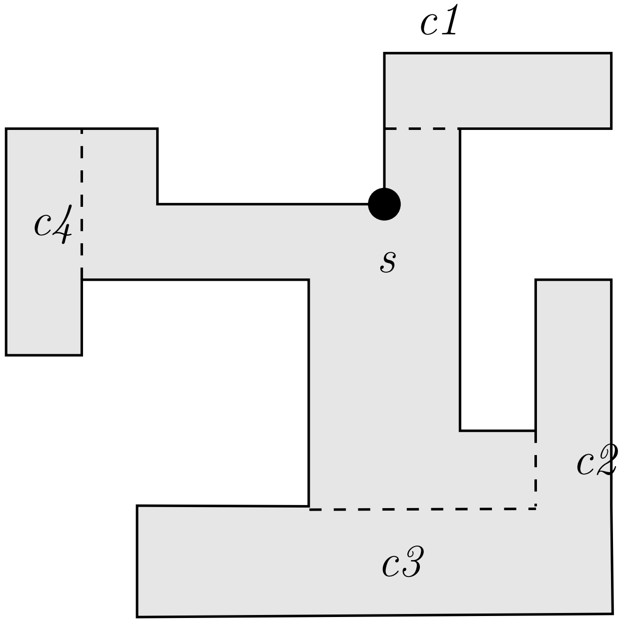
e) Order of the essential cuts



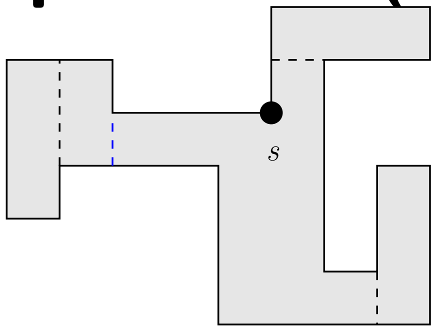
Rep.: Order along the boundary Lem.

- Rectilinear polygon
- Essential cuts intersect at most once
- SWR visits cuts by order around boundary
- Contradiction! Shortcut!
- $O(n)$ Algorithm!!



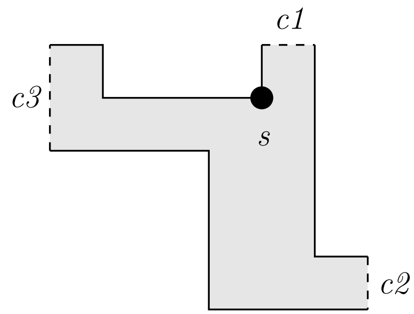


Rep.: SWR (RW Polygon) $O(n)$ Theo.



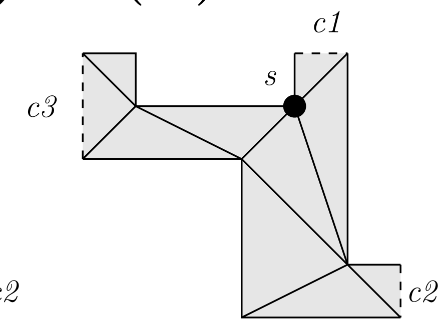
(i) Wesentliche Cuts

O(n)



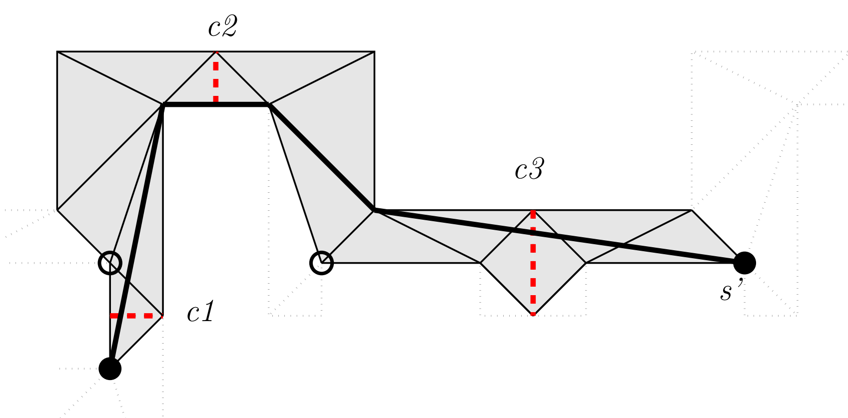
(ii) Abschneiden!

O(n)



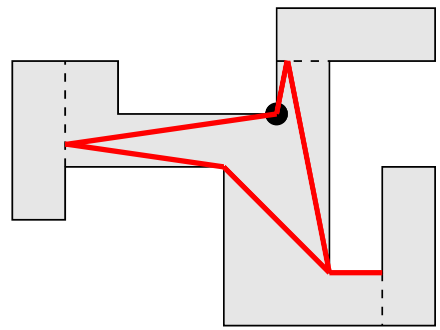
(iii) Triangulation

O(n)



(iv) Spiegeln und Ausrollen!!
(v) Weg berechnen

O(n)

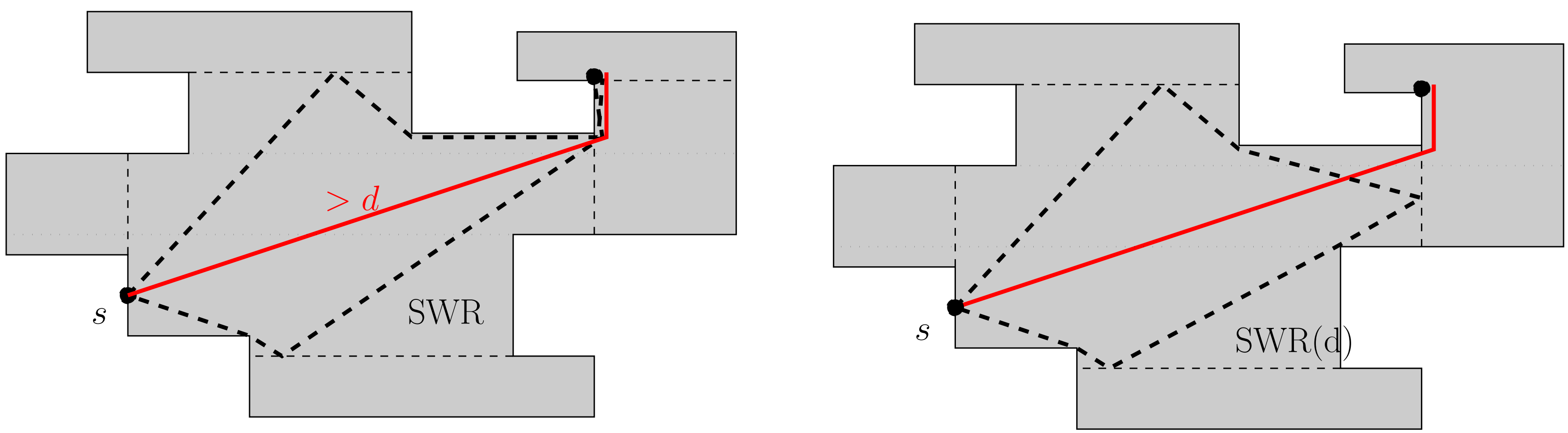


(vi) Zurckklappen!

O(n)

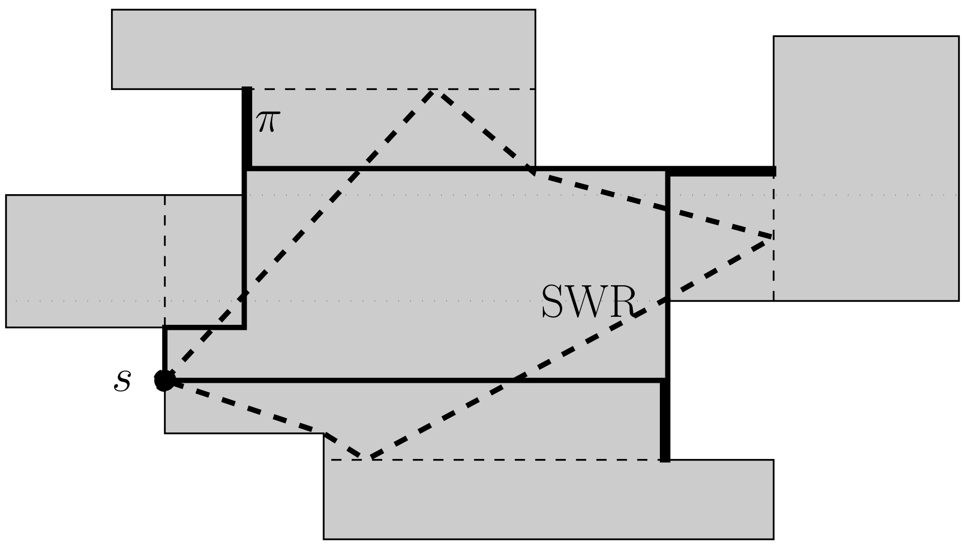
Rep.: SWR (Rect. polygon) depth restriction?

- Ignore cuts with distance $> d$, Shortest path to cut
- Ignore a cut here, Algorithm as before
- $\text{Expl}\,(d) = \text{Expl}_{\text{OPT}}(d)$
- **Theorem**: 8 Approximation of optimal search path!



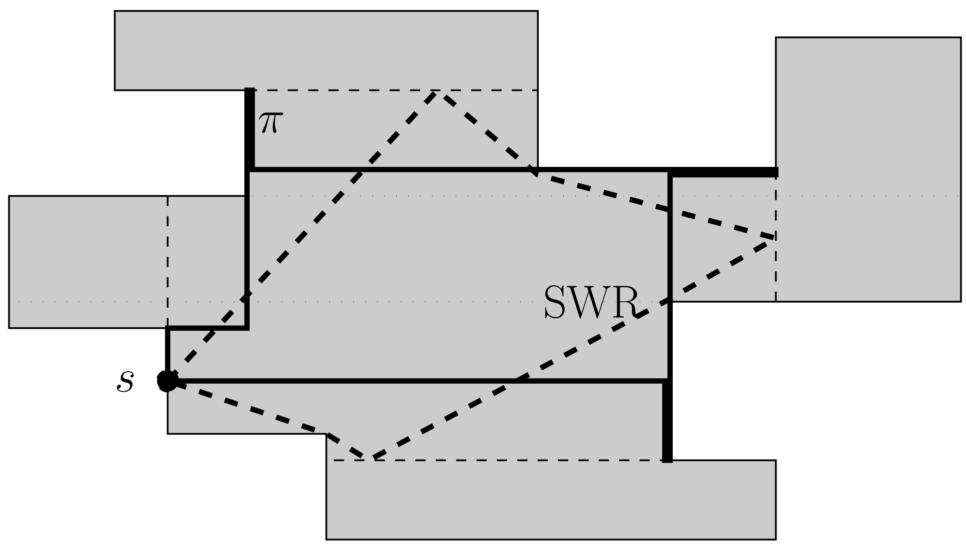
Rectilinear polygons Online

- Agent with vision, start point s
- Szene is not known!
- Depth restriction?
- First: General approach. Then: Depth restriction!



Rectilinear polygons Online

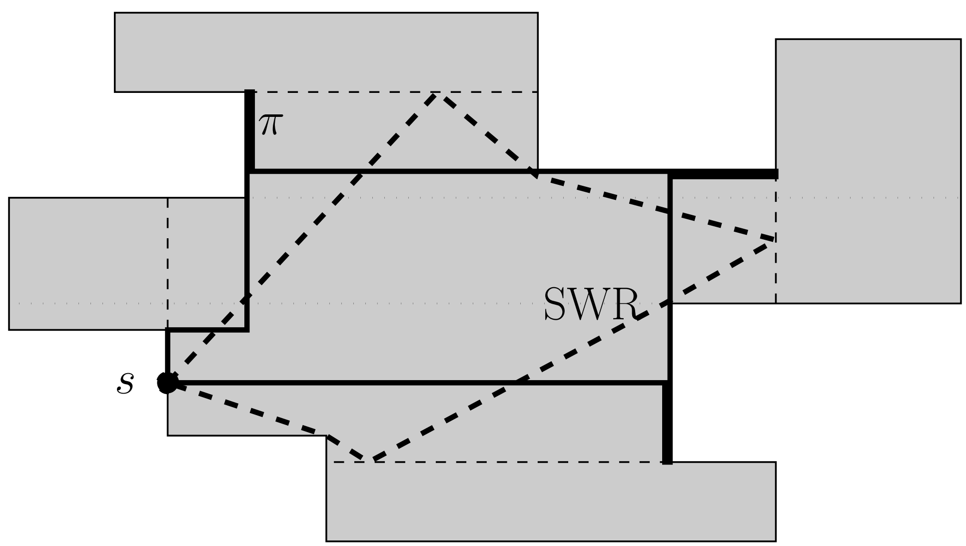
- Assume, s boundary point
- Greedy! Scan the boundary up to the first invisible point. Move to the cut on the shortest path!
- Shortest L_1-path to the cut, online!
- **Algorithmus** Always approach the next reflex vertex along the boundary that blocks the visibility.



Online variant for rectilinear polygons

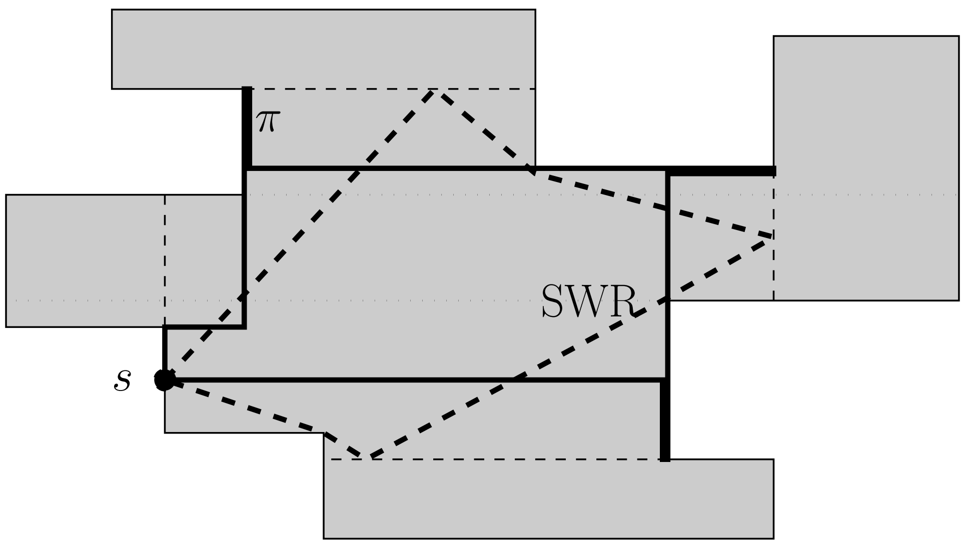
Exploration rectilinear polygons DKP

WHILE Polygon not fully explored
Do Move orthogonally toward the cut of next reflex vertex in cw-order along the boundary
END



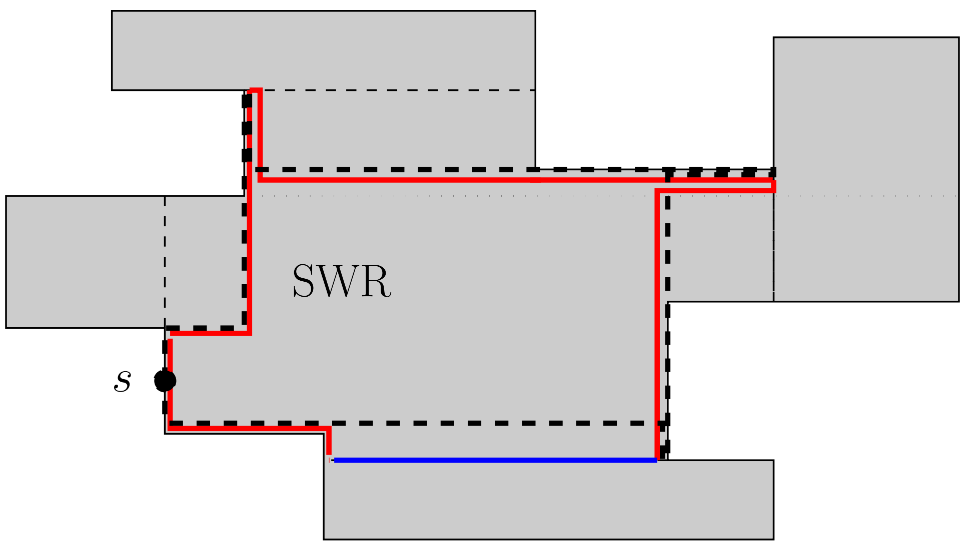
L_1-opt./$\sqrt{2}$-competitive! Theorem

- Analysis: 1) Show L_1-optimal path to essential cuts
- Inductively! Number of steps! First step, trivial!
- Ass.: Along opt. L_1-path to an essential cut!
- Next step, visit cut, ok! Otherwise, vertex on the track! Next step also optimal!



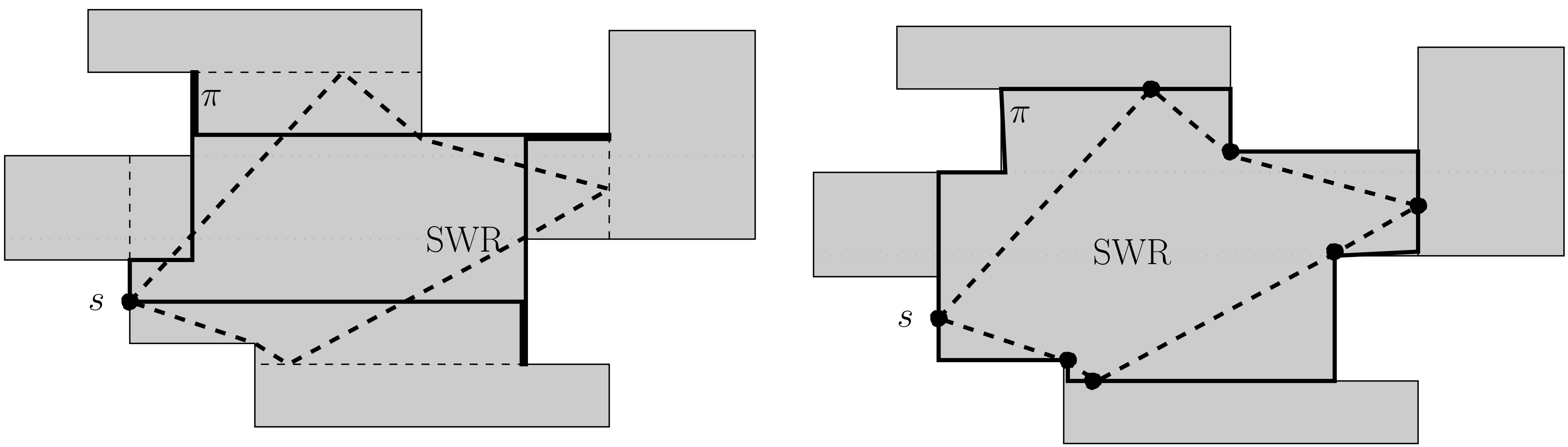
L_1-opt./$\sqrt{2}$-competitive! Theorem

- Sketch!
- Analysis: 2) Combine the optimal L_1-paths!
- L_1-paths, combination is also L_1-optimal!



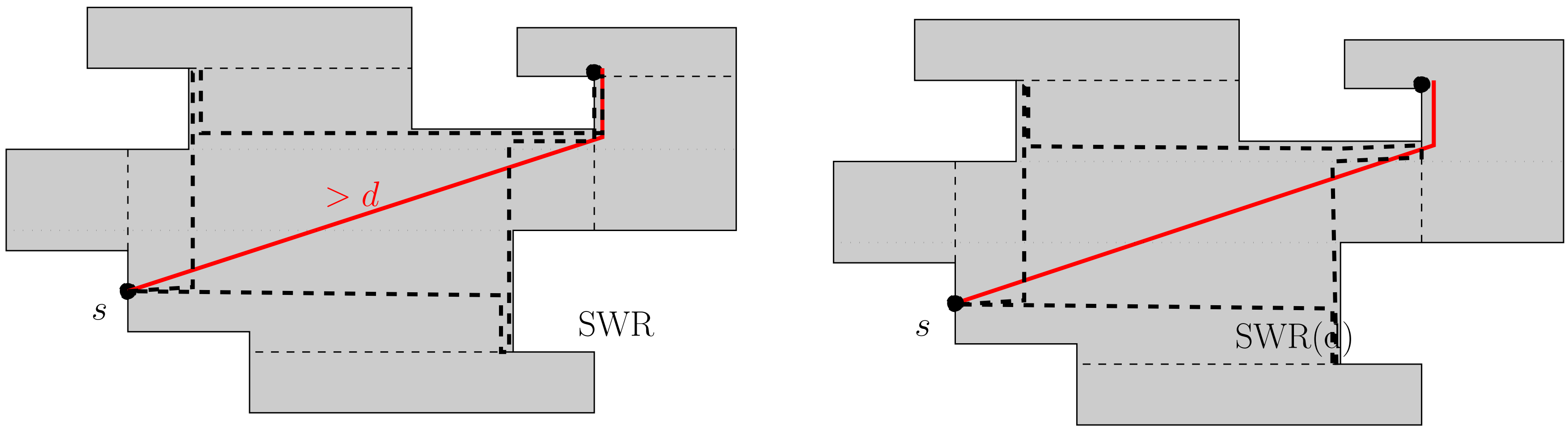
L_1-opt./$\sqrt{2}$-competitive! Theorem

- *Shift* paths toward the cuts, such that (Euclidean) SWR is included! Path has the same length!
- L_1-optimal path between any two points!
- Euclidean shortest path in between
- Triangle! Situation! Blackboard! $\sqrt{2}$-Umweg maximal!



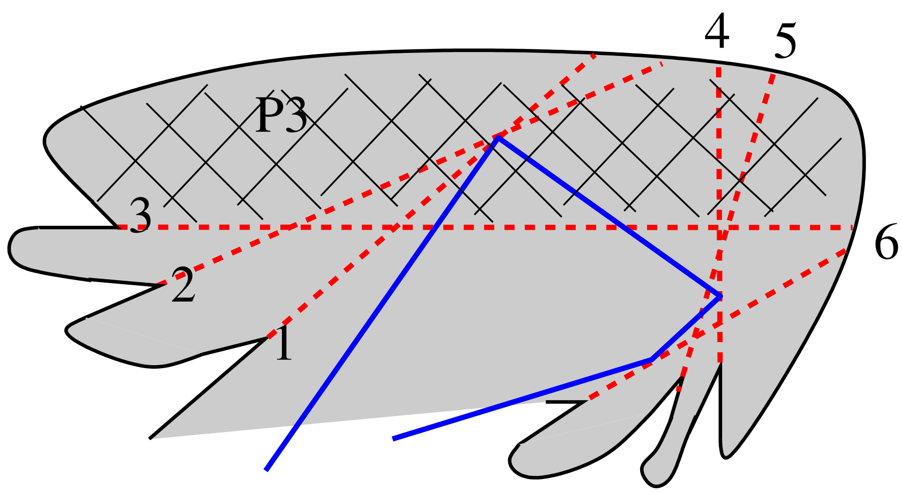
L_1-opt./$\sqrt{2}$-competitive! Theorem

- $\sqrt{2}$-competitive
- Depth restrictable!
- Online: Ignore Cuts with distance $> d$
- $\mathrm{Expl}_{\mathrm{ONL}}(d) \leq \sqrt{2}\ \mathrm{Expl}_{\mathrm{OPT}}(d)$
- **Theorem**: $8\sqrt{2}$-Approximation



SWR (General case): Offline!

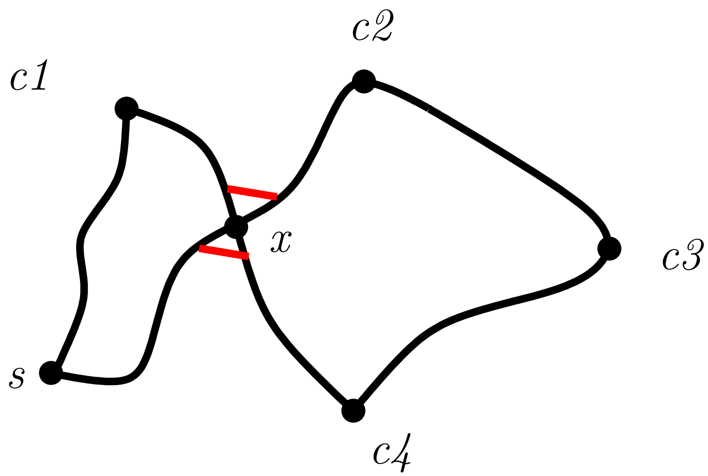
- Corner problem!!
- Sequence of essential cuts, successive cuts
- Not visited by order along boundary.
- But the corresponding P_{c_i}!!!



Visisting the corners!

The SWR visits the different corners by the order along the boundary.

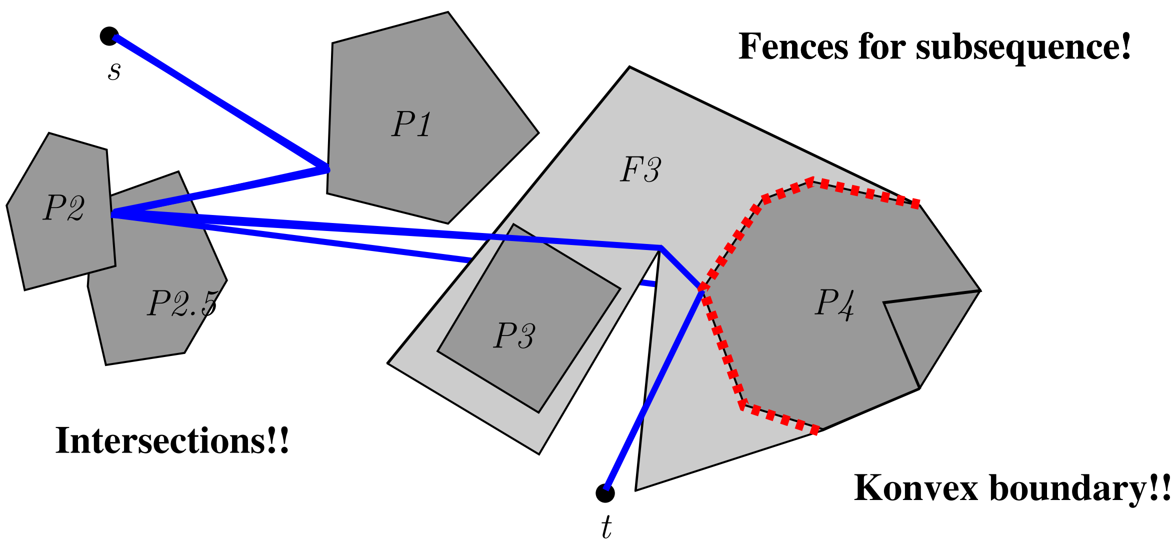
Proof: As before! Local shortcuts!



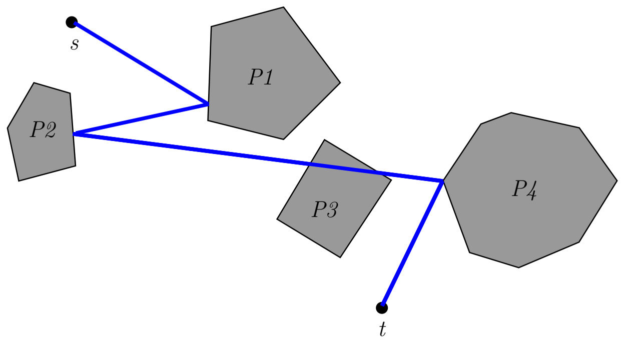
Adjustments inside the corners: Not easy to realize!

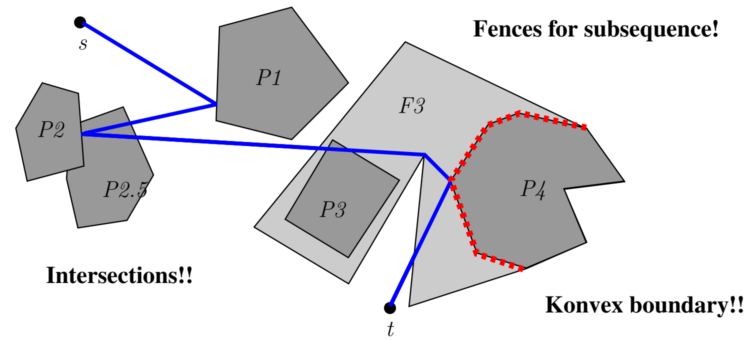
Touring a sequence of polygons (TPP)

- Sequence of convex polygons
- Start s, target t
- Visit polygons w.r.t. sequence, shortest path



TPP



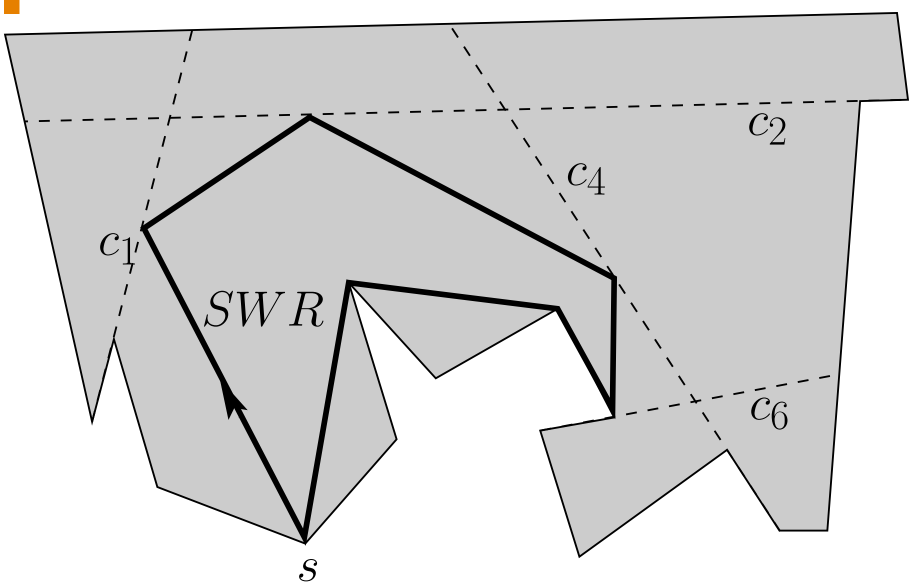


- Simple version:
- $O(nk \log \frac{n}{k})$
- Build(Query): $O(nk \log \frac{n}{k})$
- Compl.: $O(n)$
- Query (fixed s): $O(k \log \frac{n}{k})$

- General version:
- Fences, convex boundary, etc.
- $O(nk^2 \log n)$
- Build(Query): $O(nk^2 \log n)$
- Compl.: $O(nk)$
- Query (fixed s): $O(k \log n + m)$

Results from: Dror, Efrat, Lubiw, Mitchell 2003!!

Application: SWR



Essntila *parts*!

Use the order along the boundary!

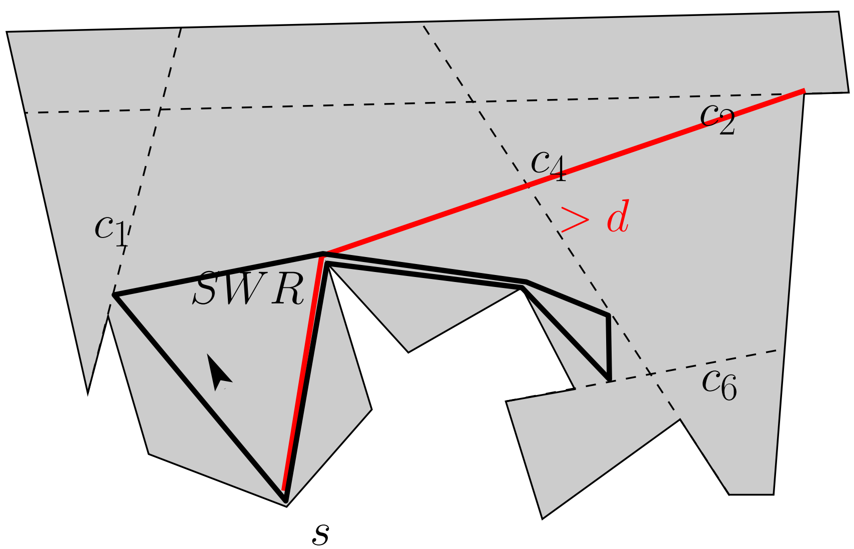
One common fence, intersections!

Start and target identical!

- $O(n^4)$ '91
- $O(n^4)$ Tan et al. '99
- $O(n^3 \log n)$ by this result!
- **Theorem**

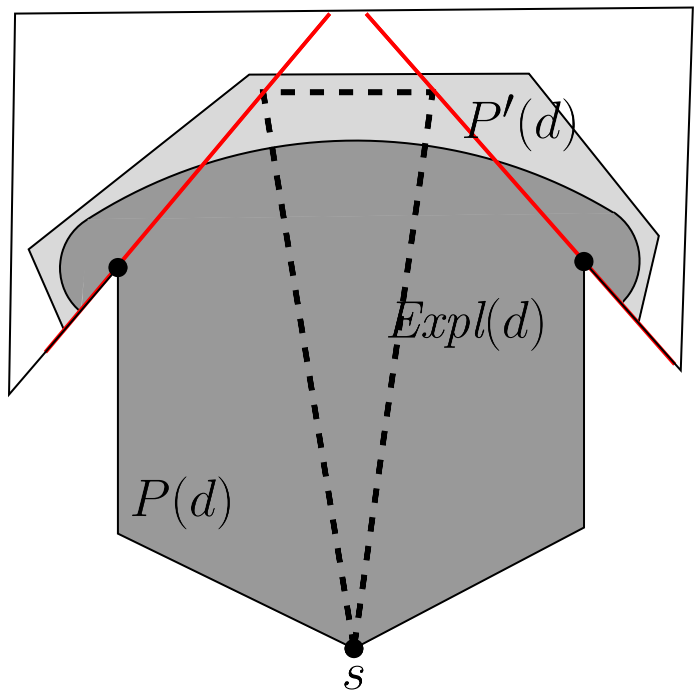
Application: General simple polygons Offline

- Compute optimal exploration tour
- Agent with vision, start s at the boundary
- Depth restriction: Ignore cuts with distance $> d$
- $\mathrm{Expl}\,(d) = \mathrm{Expl}_{\mathrm{OPT}}(d)$
- **Theorem**: 8 Approximation, Online??



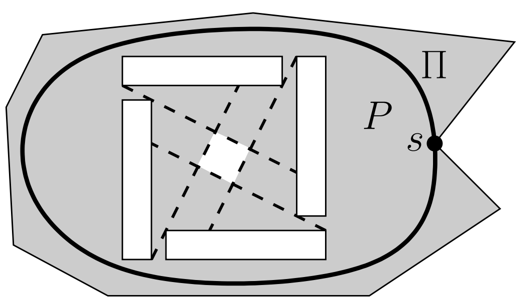
Remark: Depth restriction Offline

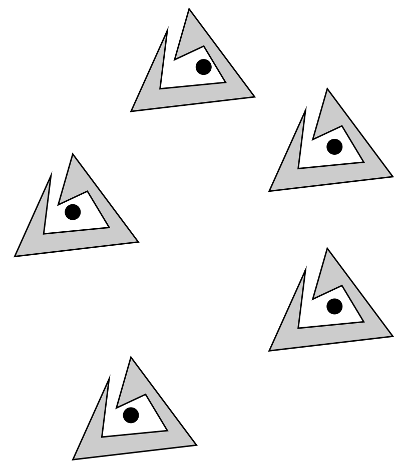
- $P(d)$ subset of P
- $\mathrm{Expl}\,(d) = \mathrm{Expl}_{\mathrm{OPT}}(d)$ can leave $P(d)$



Vision: Negative result, polygon with *holes*

- Much more difficult
- Example: See boundary $\Leftrightarrow$ see everything
- Not true for such scenes
- Offline: Computation SWR is NP-hart, reduction idea TSP





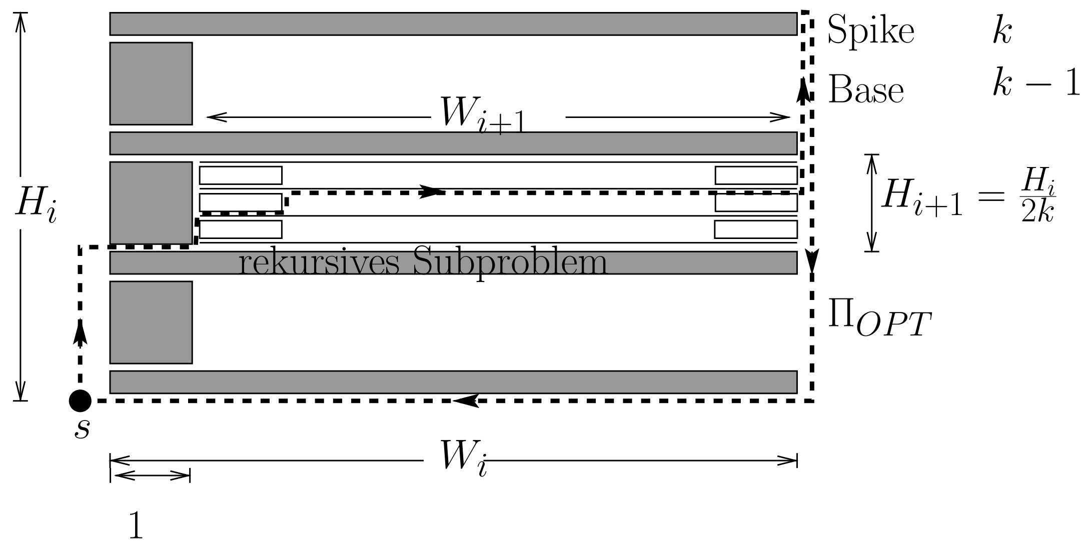
Polygons with holes

There is no constant online approximation of the optimal search ratio

Theorem Let A be an online strategy for the exploration of a polygon with n obstacles (holes), we have: $|\Pi_A| \geq \sqrt{n}|\Pi_{OPT}|$

Proof: LB by examples!

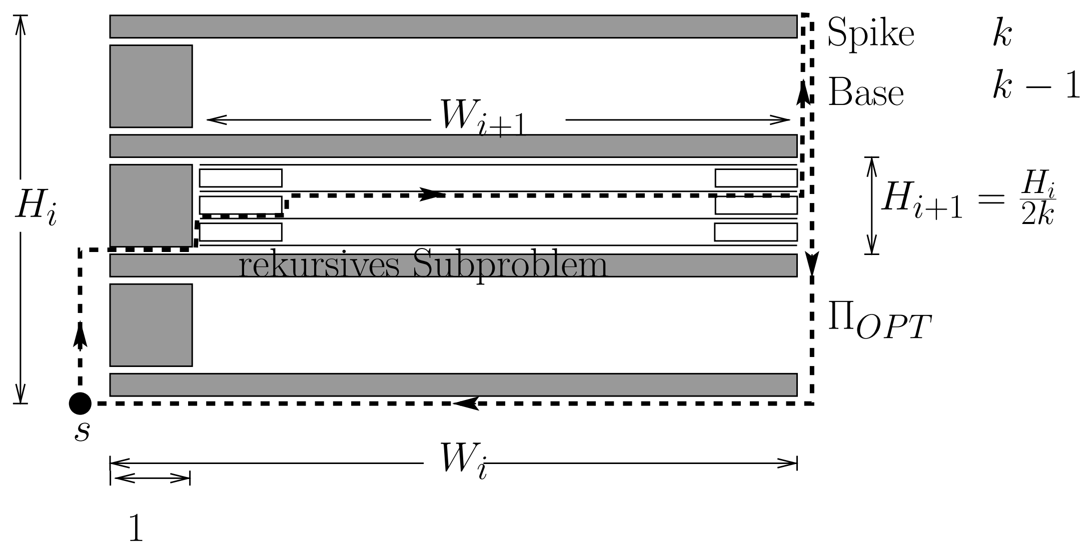
Polygon with holes: $|\Pi_A| \geq \sqrt{n}|\Pi_{OPT}|$



- $W_1 = 2k$, $H_1 = k$, k spikes, $(k-1)$ bases, $(2k-1)k$ rectangles
- $H_i = \frac{H_1}{(2k)^{i-1}}$, $W_i = 2k - i + 1 \geq k$, $i = 1, \ldots, k$
- Situation H_i: Online strategy does not know position of block H_{i+1}

- Left side: Look behind any block
- Right side: Move once upwards
- Adversary: Find block after $\Omega(k)$ steps
- Altogehter: $\Omega(k \times k)$ for any Strategy

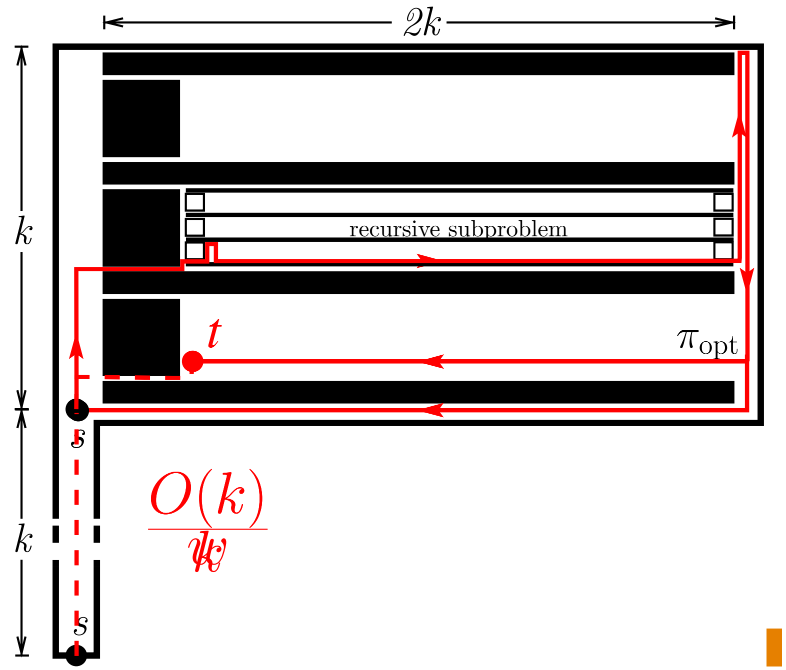
Polygons with holes: $|\Pi_A| \geq \sqrt{n}|\Pi_{OPT}|$



- Optimale strategy: Move directly to the block
- Go on recursively, at the end move along any block
- $[\Pi_{OPT}| = W_1 + 2\sum_{i=1}^{k} H_i \leq 6k$
- $k = \lfloor \sqrt{n} \rfloor$ gives the result

Polygons with holes Corollary

- No $O(1)$-competitive exploration for such environments ($\Omega(\sqrt{n})$)
- Optimal exploration has a bad Search Ratio
- Trick: Extension
- Then: Optimal exploration has Search Ratio $O(1)$
- Any online strategy has Search Ratio $\Omega(k)$



Summary

- Connection between exploration and search:
- $\exists$ constant-competitive, depth-restrictable exploration strategy
  $\Rightarrow \exists$ search strategy with mit competitiver Search Ratio
- $\nexists$ constant-competitive exploration strategy,
  but $\exists$ 'extendable' lower bound
  $\Rightarrow \nexists$ search strategy with competitive Search Ratio