Exploration of scenes with obstacles:

Lower bound (Albers, Kursawe, Schuierer '02):

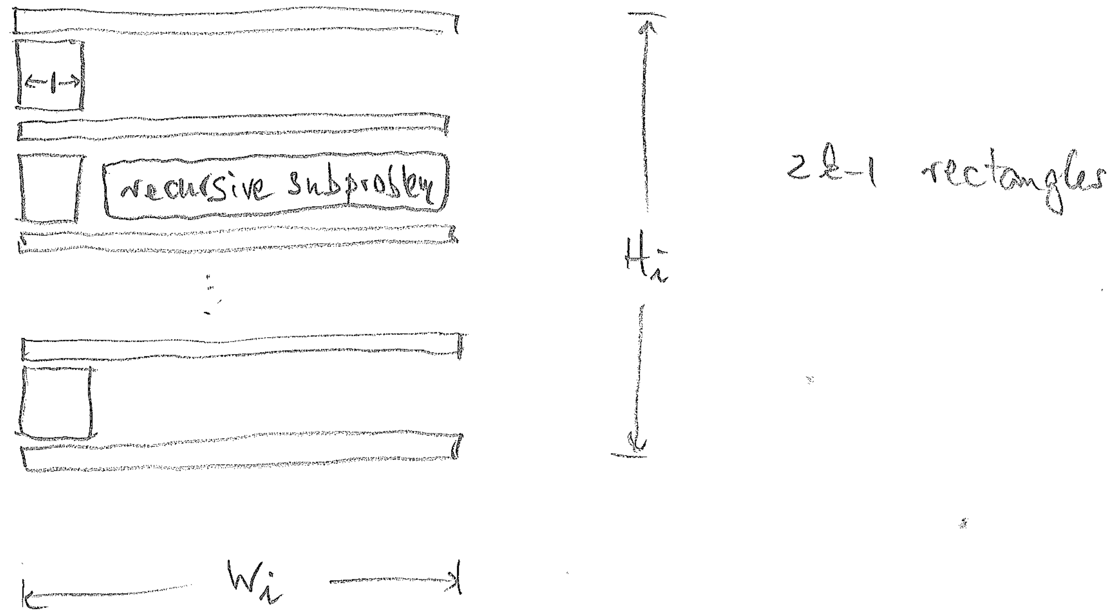
Theorem In exploring n rectangular obstacles, each strategy can be forced to produce a path π of length

$$|\pi| \geq c\sqrt{n}\,|W_{opt}|,$$

for some constant c.

Proof For $i = 1, 2, \ldots, k$ build a k-comb C_i

C_i:



where $H_1 = k$, $H_{i+1} = \frac{H_i}{2k}$

$W_1 = 2k$, $W_{i+1} = W_i - 1$, $i = 1, 2, \ldots, k$

at exactly one position, a recursive subproblem is hidden, robot needs to squeeze through to find out.

base rectangle

b

Let S be an exploration strategy, starting from s at the lower left corner. (8)

If S moves to the right side of C_i:

$$|\pi_i| \geq W_i = 2k-(i-1) \geq k$$

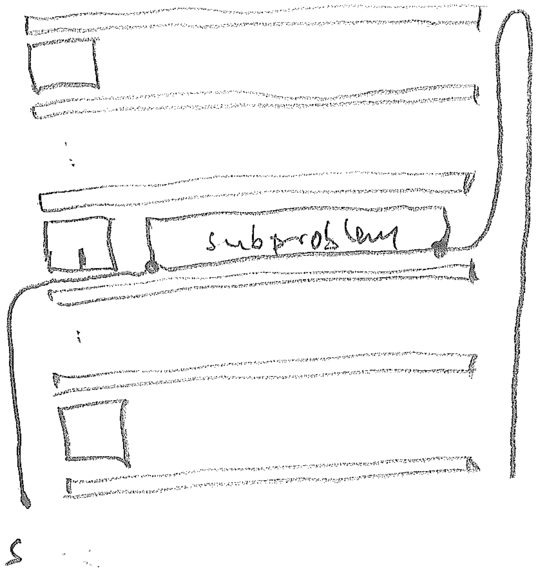
if S passes every base rectangle from the left (to find recursive subproblem in the last slab visited) $\Rightarrow$

$$|\pi_i| \geq k \cdot 1$$

$\Rightarrow$ over all levels, $|\pi| \geq k^2$. (*)

W_{opt} can move as follows in C_i:

C_i



$\leftarrow$ recursion inside subproblem

path in C_i (without recursive parts):

$i<k$: $2H_i + 1$

$i=k$: $2H_k + W_k$ (no recursion; robot moves up, to the right, and down)

plus W_1 for returning to s.

$$\sum_{j=1}^{k-1} (2\underbrace{H_j}_{\frac{k}{(2k)^{j-1}}}+1) + \underbrace{2H_k+W_k}_{k+1} + \underbrace{W_1}_{2k} \leq 2k \sum_{j=1}^{k} \frac{1}{(2k)^{j-1}} +$$

since $$\sum_{j=1}^{k} \frac{1}{(2k)^{j-1}} < \sum_{j=0}^{\infty} \frac{1}{(2k)^{j}} = \frac{1}{1-\frac{1}{2k}} = \frac{2k}{2k-1} < 2$$

we get

$|W_{opt}| < 8k$, hence

$$\frac{|\Pi_s|}{|W_{opt}|} > \frac{k^2}{8k} = \frac{1}{8} k \geq c \cdot \sqrt{\underbrace{k(2k-1)}_{\substack{=: n \\ = \#\text{ rectangles used}}}}$$

$\boxed{\text{Theorem}}$

Not much known about upper bounds for exploring simple polygons with holes.
Room and obstacles rectilinear: $O(n)$ competitive factor

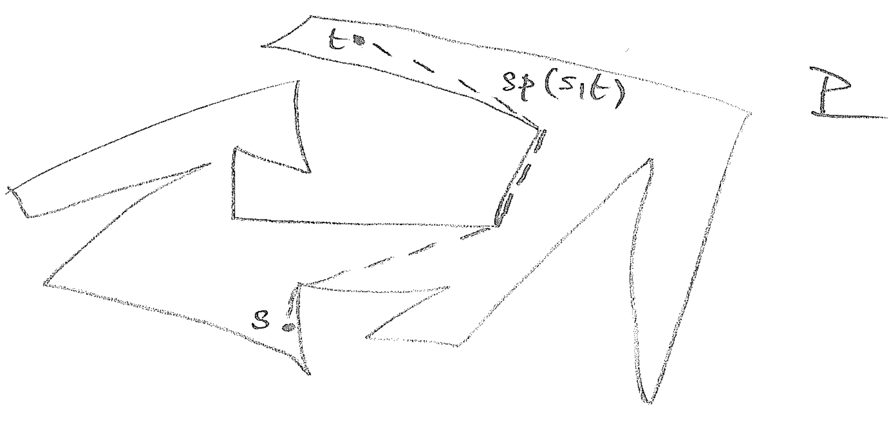
Searching

a) in a simple polygon

Given: simple polygon P, $s, t \in P$, robot with vision and memory, but no knowledge of P

Task: starting from s, find t on a path as short as possible.

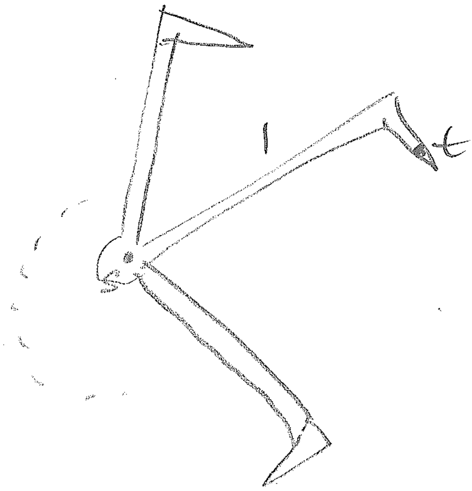
Optimum solution: Shortest path $sp(s,t)$ from s to t in P



d

Clearly, in arbitrary simple polygons no constant competitive factor can be achieved.

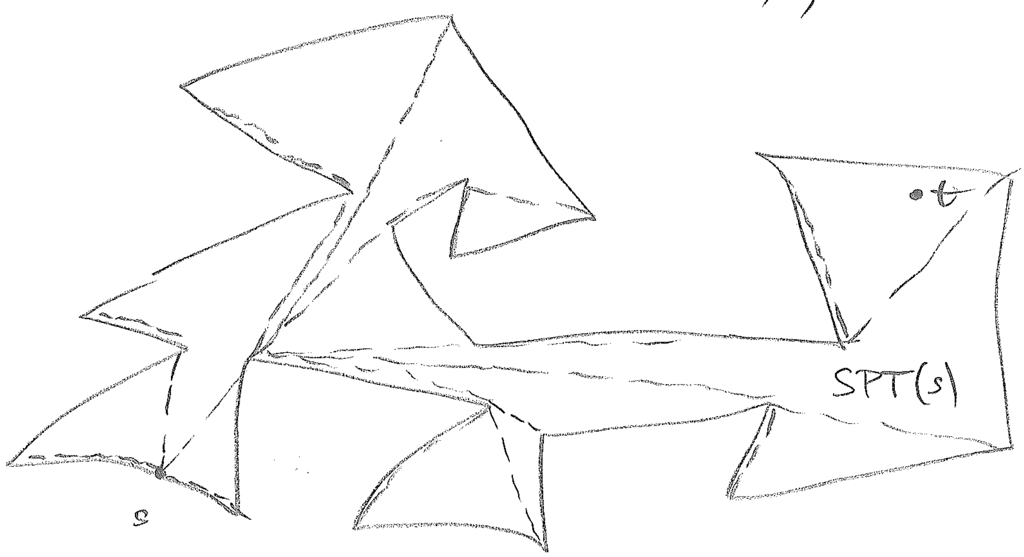
target t can be located in last leg visited



n "legs"

$\Rightarrow$ ratio $= \frac{2n-1}{1}$

Reasonable practical idea: apply optimum m-way search ("round-robin doubling") on shortest path tree of s (ignoring that paths can share edges), m = # leaves of SPT(s)



Each t visible from some vertex of SPT(s) (why?)

Problems: m, SPT(s) not known in advance!

Solutions: * Instead of using cyclic exploration depths $\left(\frac{m}{m-1}\right)^i$, apply doubling after each round

* construct partial SPT

Lemma A simple polygon with n vertices can be searched with a competitive factor of 8n. (Ex.)

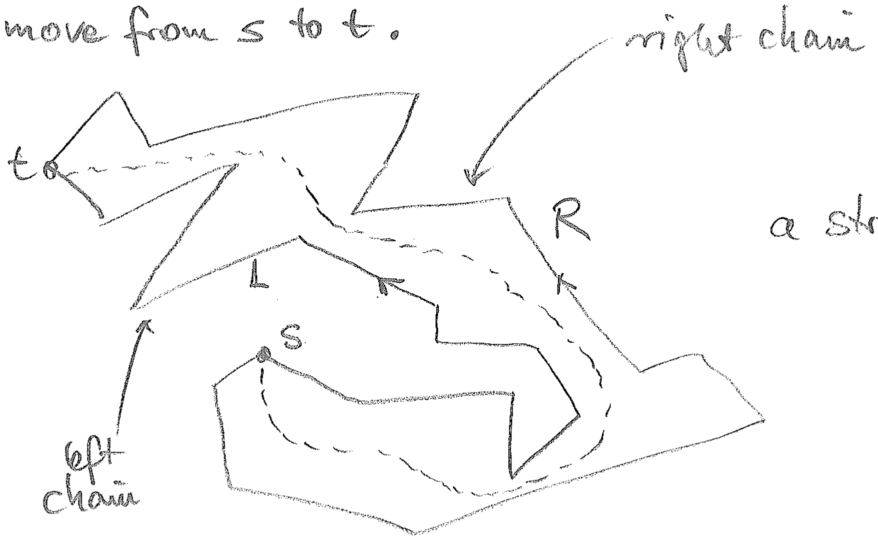
[Lemma]

Special class of simple polygons: Streets

<u>Definition</u> P a simple polygon, s, t two vertices.
P is called a street iff the two s–t boundary chains are mutually weakly visible.
<u>Robot's task</u>: To move from s to t.
<u>Example</u>

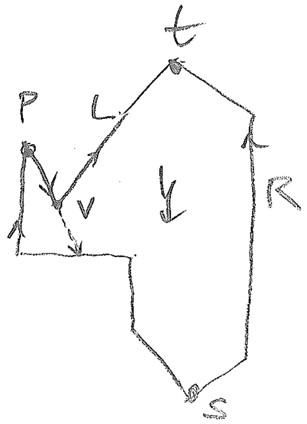
each point of L can see point of R, and vice versa

mutually weakly visible



a street

not a street: p cannot see a point of R



each vertex v ≠ s, t has an incoming and an outgoing edge

extension of outgoing edge cannot hit earlier point on the same chain.

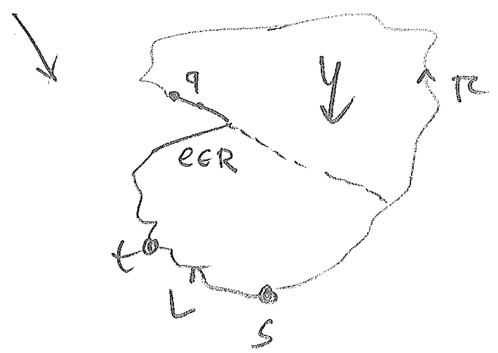
<u>Lemma 1</u> (P, s, t) is a street $\iff$ each s-to-t path in P sees all of P

<u>Proof</u> „$\Rightarrow$" Let π be an s-to-t path in a street (P, s, t), and let $p \in L$. By definition, p sees some point $q \in R$.
Segment $\overline{pq}$ connects points of L and R
$\Rightarrow$ path π must cross $\overline{pq}$
$\Rightarrow$ p visible from π.

f „⇐" Let $p \in L$. By assumption, s-to-t path R can see every point of P, in particular p. $\boxed{\text{Lemma 1}}$

<u>Structural property</u>: <u>Lemma 2</u> In clockwise orientation around s, vis(s) consists of a sequence of left caves, followed by a sequence of right caves (either one can be empty).
If t is not visible from s, it can only be contained in the last left cave, or in the first right cave.

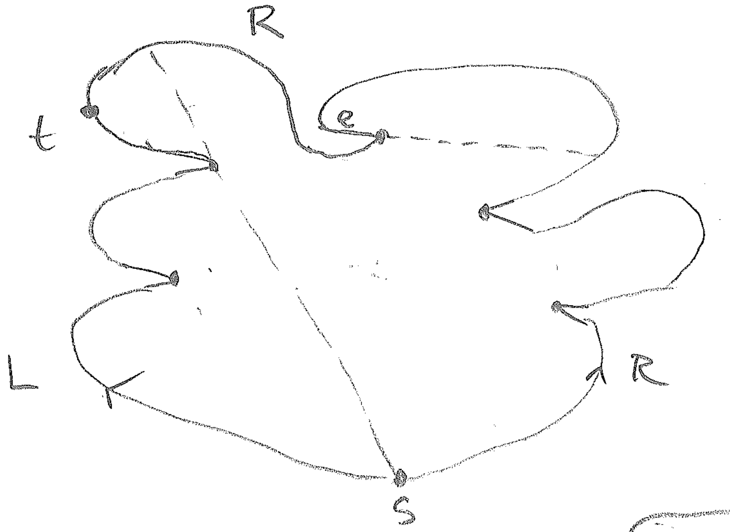
<u>Proof</u> The visible edge of a left cave must belong to L because here



point $q \in R$ would not see a point of L

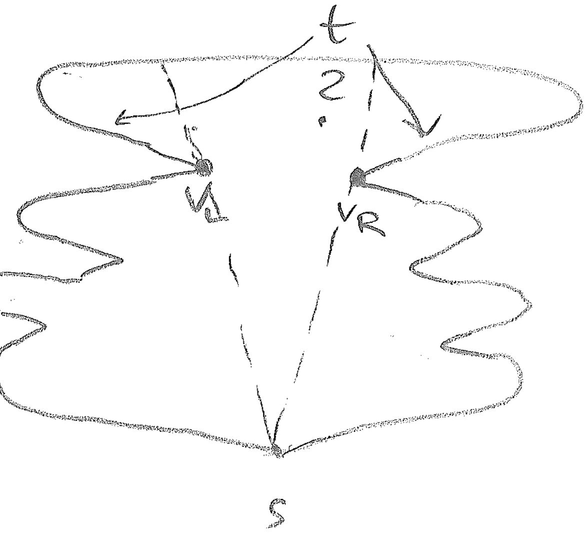
Analogously for right caves.

Clearly, pieces of L and R appear consecutively in vis(s)



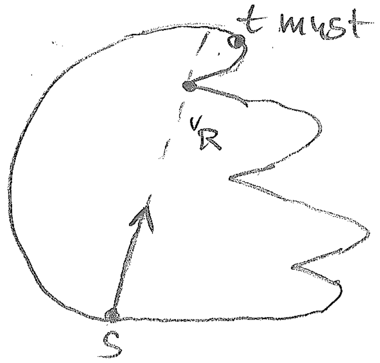
if t were situated in a left cave before the last one, points on outgoing edge in last left cave would not see a point on

⇒

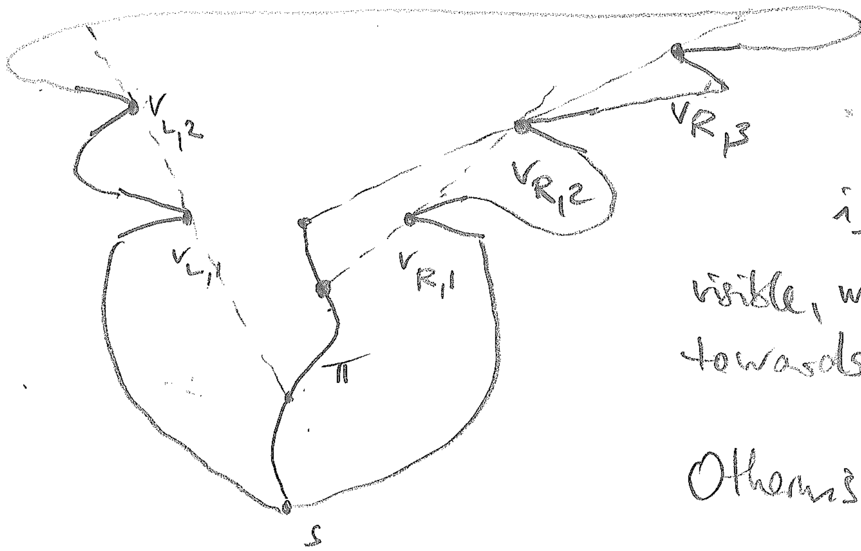


$\boxed{\text{Lemma 2}}$

if only left (or right) caves exist, robot should walk straight to reflex vertex of first right cave:



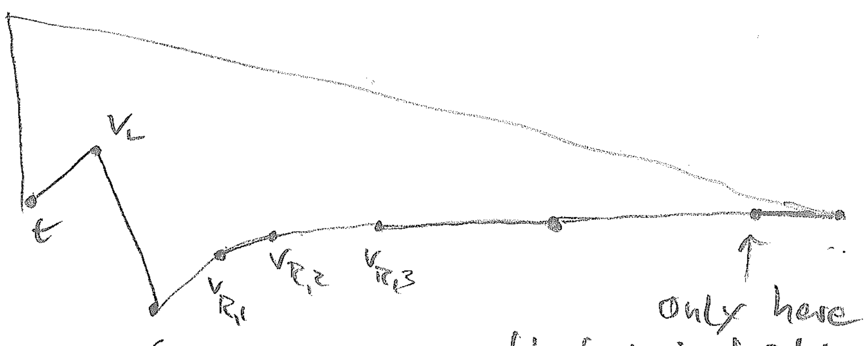
if both v_L and v_R exist, robot must approach them both in some way.
As it does, vis(CP) may change because reflex vertices get discovered, or fully explored.
Of interest: changes in v_L, v_R



if t becomes visible, move straight towards t.

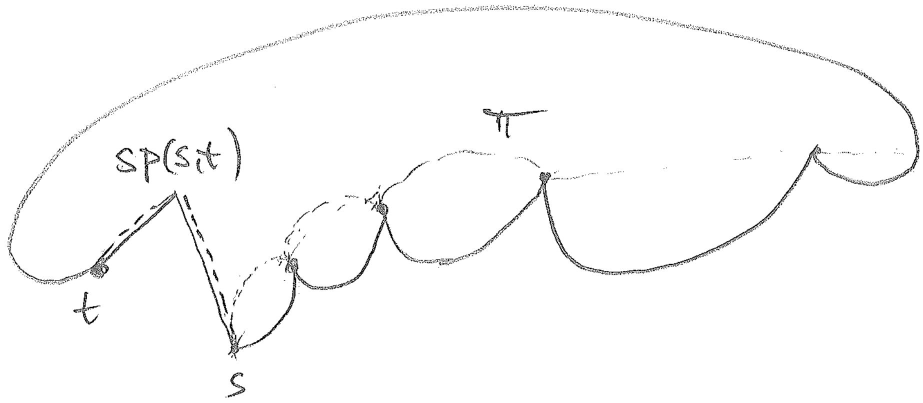
Otherwise?

<u>Idea 1</u> Walk straight to the nearest of v_L, v_R

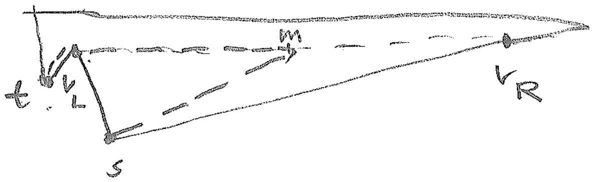


only here robot learns that t is hidden in left cave

In last example, approaching vertices on circles would help. But not here:



Idea 2 Walk straight towards the middle point on $\overline{v_L v_R}$

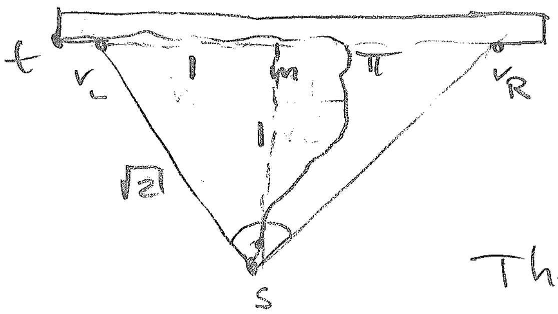


does not work, either.

But we can turn it into lower bound construction!

Lemma 3 No street-searching strategy can have a competitive factor $< \sqrt{2}$.

Proof



Any strategy must reach segment $\overline{v_L v_R}$. Suppose this happens to the right of m. Then place target t in left.

$\Rightarrow$ $|\pi| \geq 1 + 1 = 2$, but $sp(s,t) = \sqrt{2}$

(ignoring mini-edges) $\boxed{\text{Lemma 3}}$