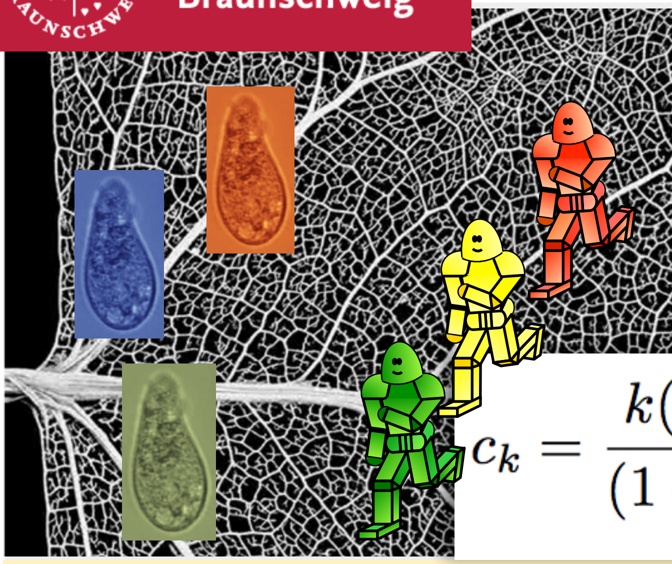
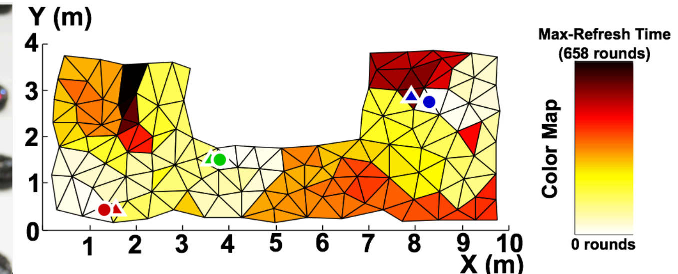
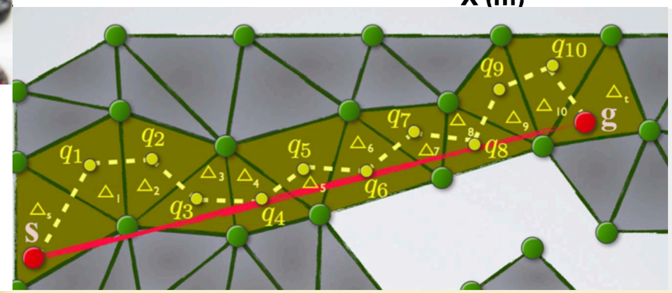


$$c_k = \frac{k(1+\frac{z_k}{k}) - (k-1)}{(1+\frac{z_k}{k}) - \frac{1}{(1+\frac{z_k}{k})^{k-1}}}$$

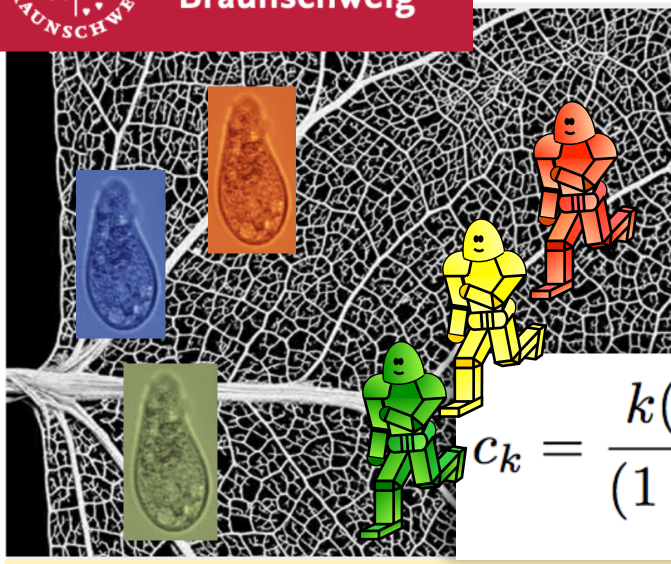


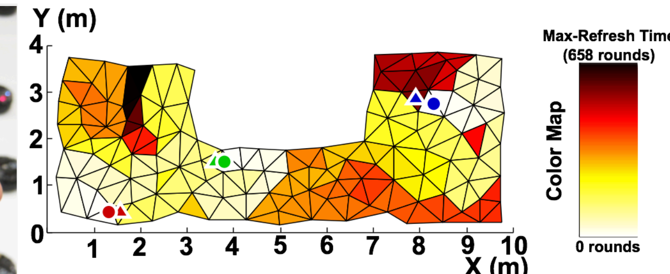


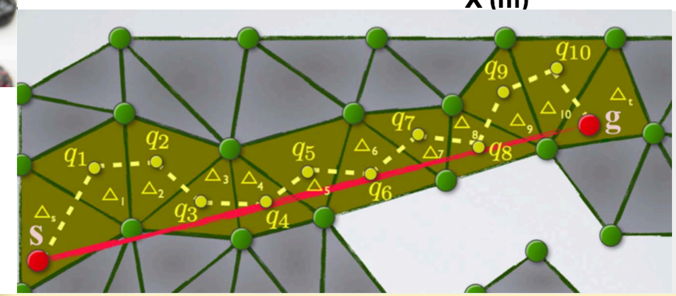
Online Algorithms 2024

Sándor P. Fekete

Technische Universität Braunschweig







$$c_k = \frac{k(1+\frac{z_k}{k}) - (k-1)}{(1+\frac{z_k}{k}) - \frac{1}{(1+\frac{z_k}{k})^{k-1}}}$$

Online Navigation for Robots

Pravesh Agrawal, Aaron Becker, Erik D. Demaine

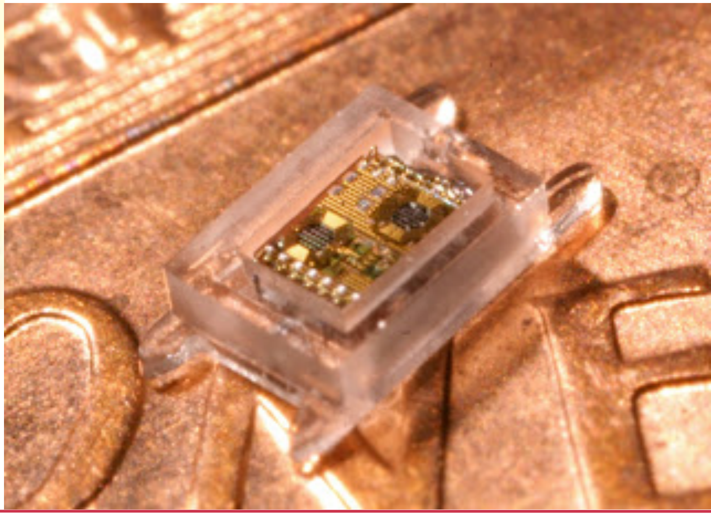
Sándor P. Fekete

Golnaz Habibi, Rolf Klein, Alexander Kröller, Andreas Nüchter

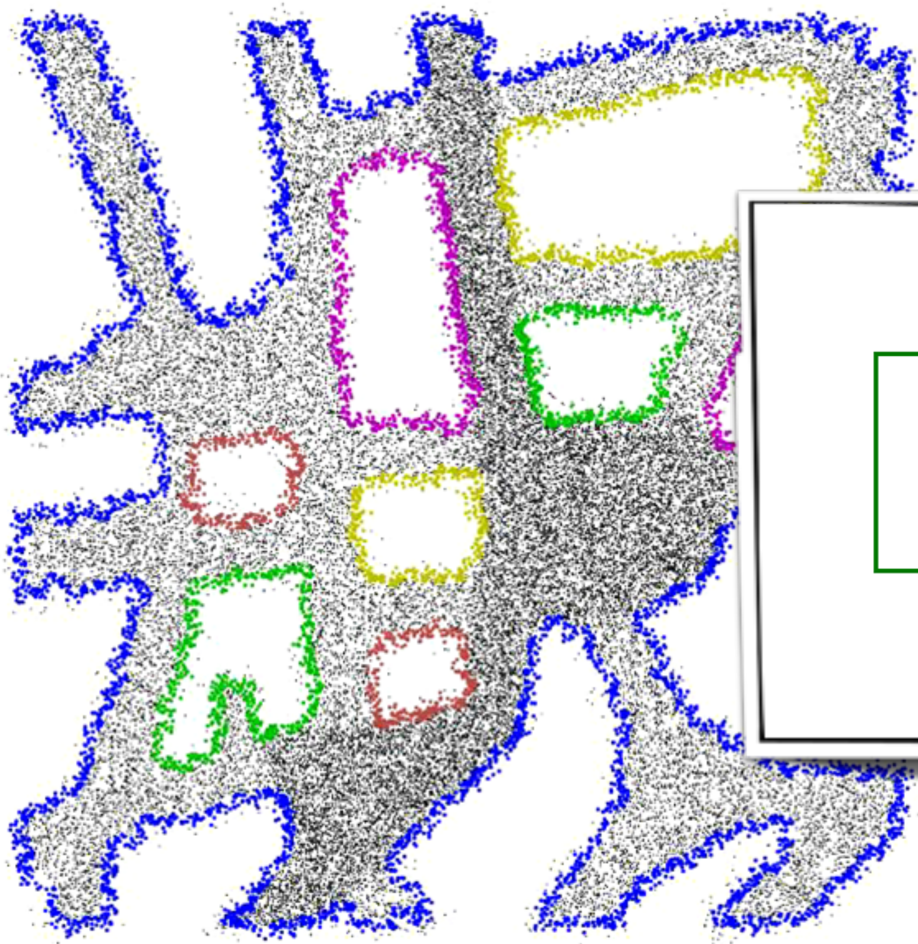
Seoung Kyou Lee, James McLurkin, Christiane Schmidt

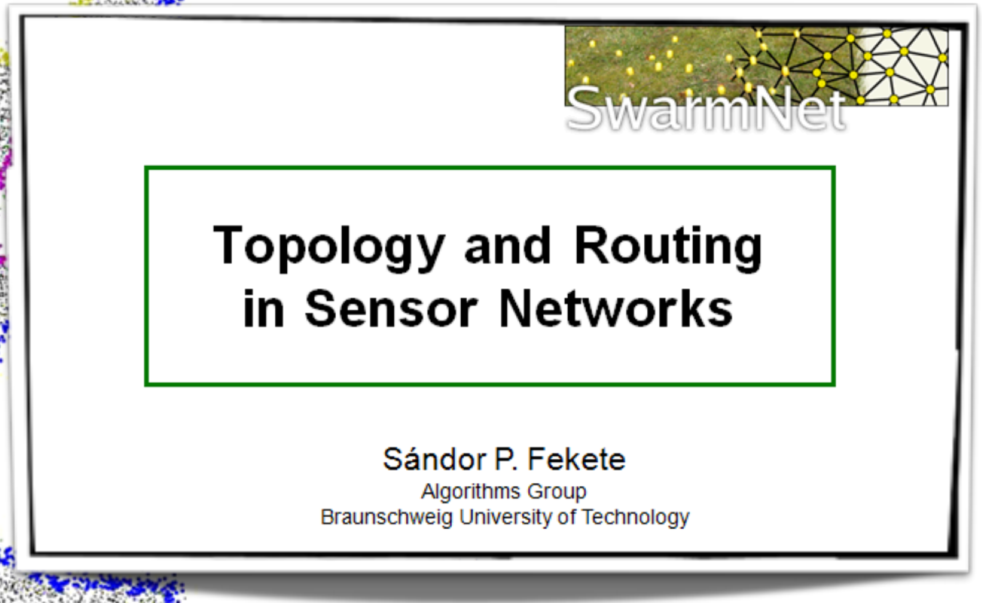
Preface: Processors and Mobile Objects

Processors

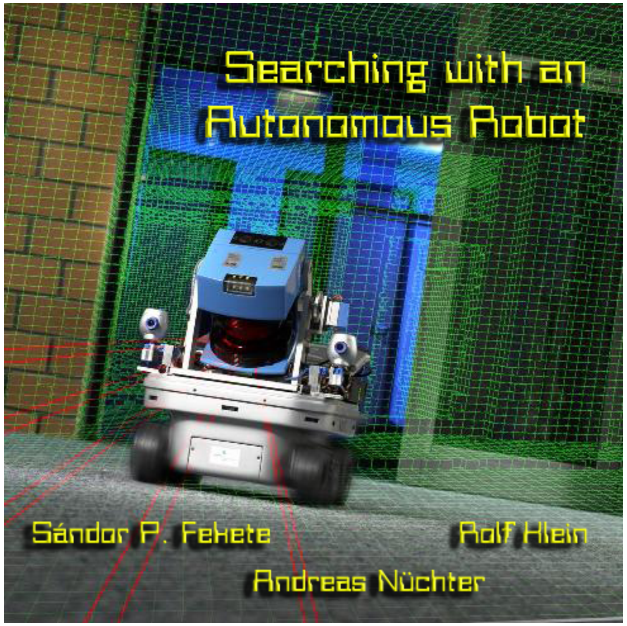


"Smart Dust"



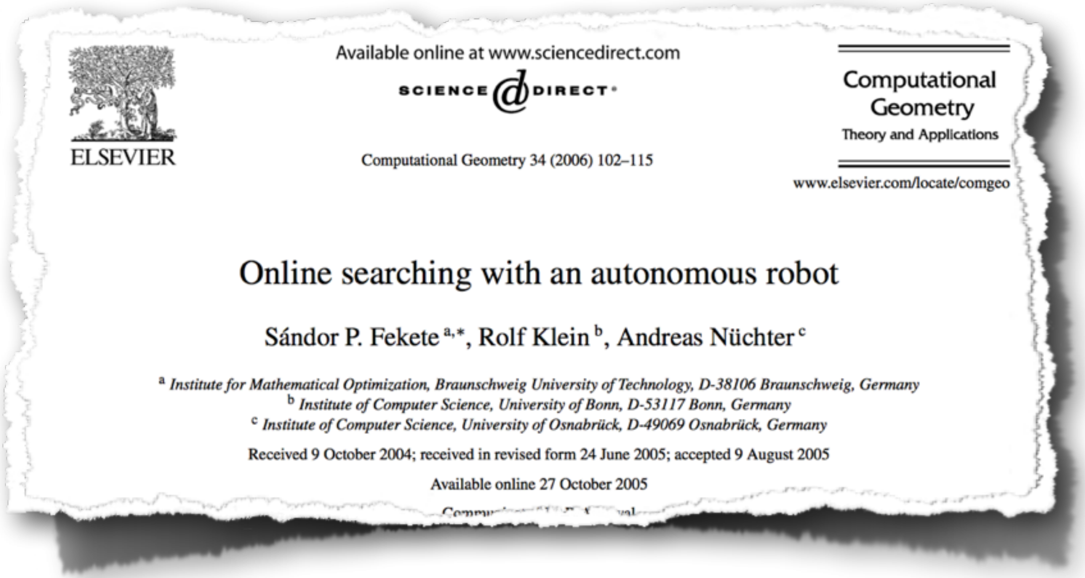

SwarmNet
Topology and Routing
in Sensor Networks
Sándor P. Fekete
Algorithms Group
Braunschweig University of Technology

Mobile Objects and Robots



Part 1: One Robot

Part 1.1:
Looking around a corner



Available online at www.sciencedirect.com
SCIENCE DIRECT®

ELSEVIER

Computational Geometry 34 (2006) 102–115

Computational Geometry
Theory and Applications

www.elsevier.com/locate/comgeo

Online searching with an autonomous robot

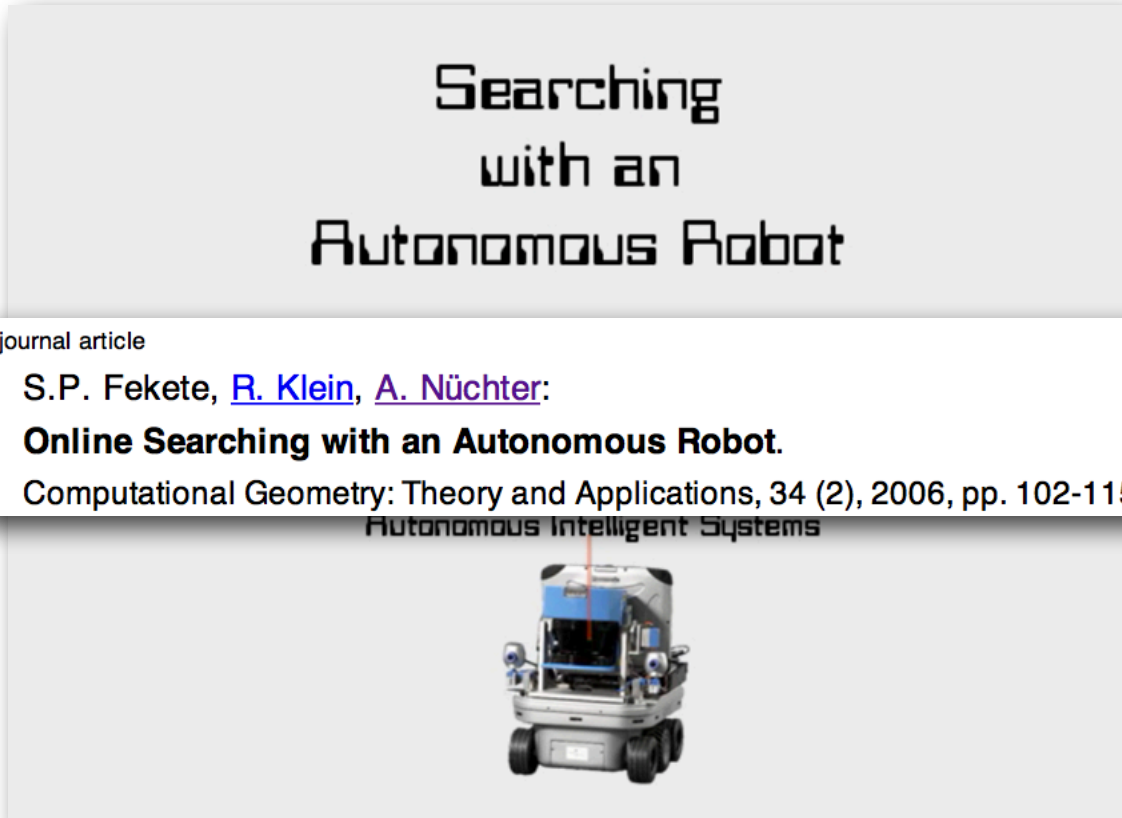
Sándor P. Fekete [a,*], Rolf Klein [b], Andreas Nüchter [c]

[a] *Institute for Mathematical Optimization, Braunschweig University of Technology, D-38106 Braunschweig, Germany*
[b] *Institute of Computer Science, University of Bonn, D-53117 Bonn, Germany*
[c] *Institute of Computer Science, University of Osnabrück, D-49069 Osnabrück, Germany*

Received 9 October 2004; received in revised form 24 June 2005; accepted 9 August 2005

Available online 27 October 2005

Video!



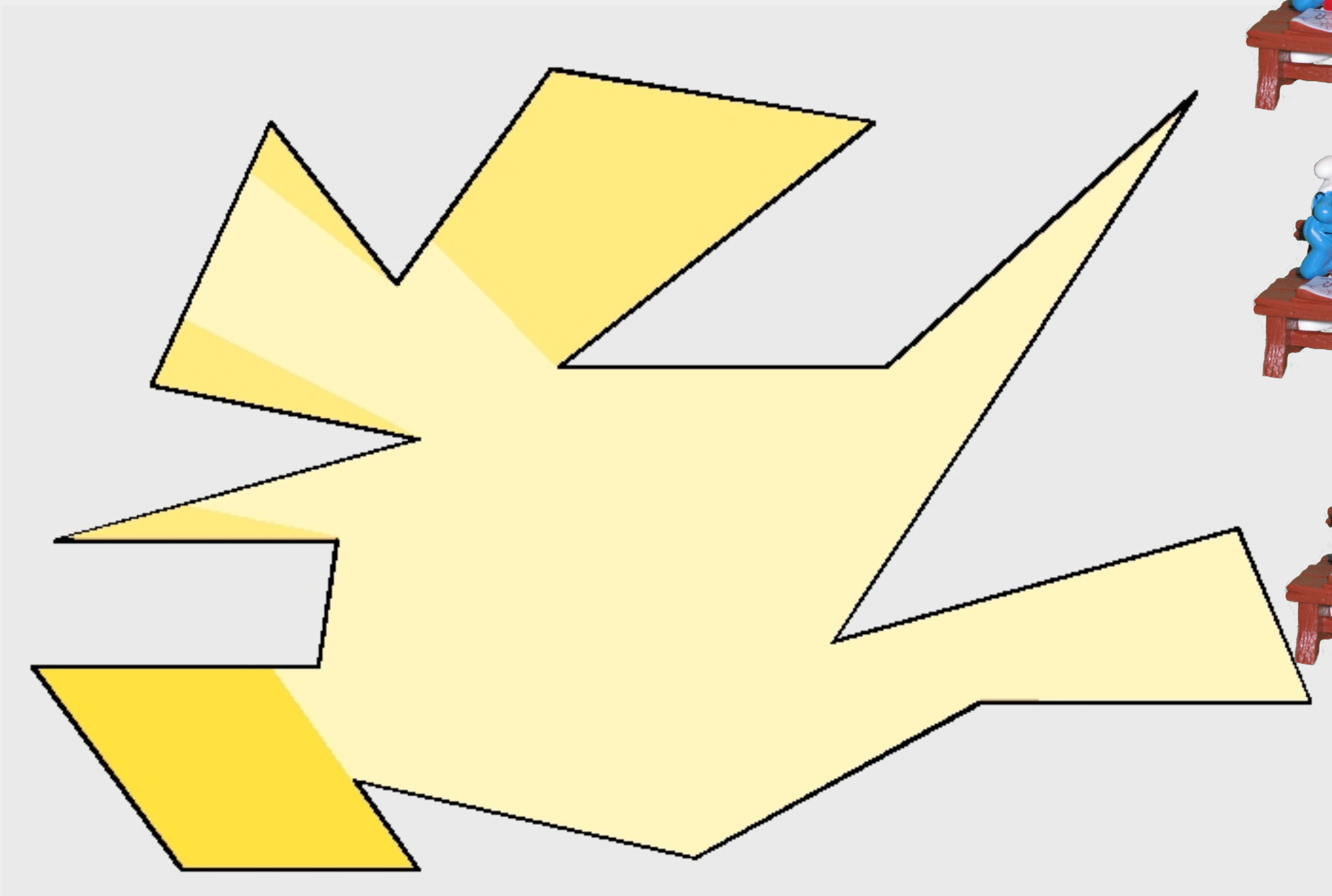
journal article

S.P. Fekete, R. Klein, A. Nüchter:

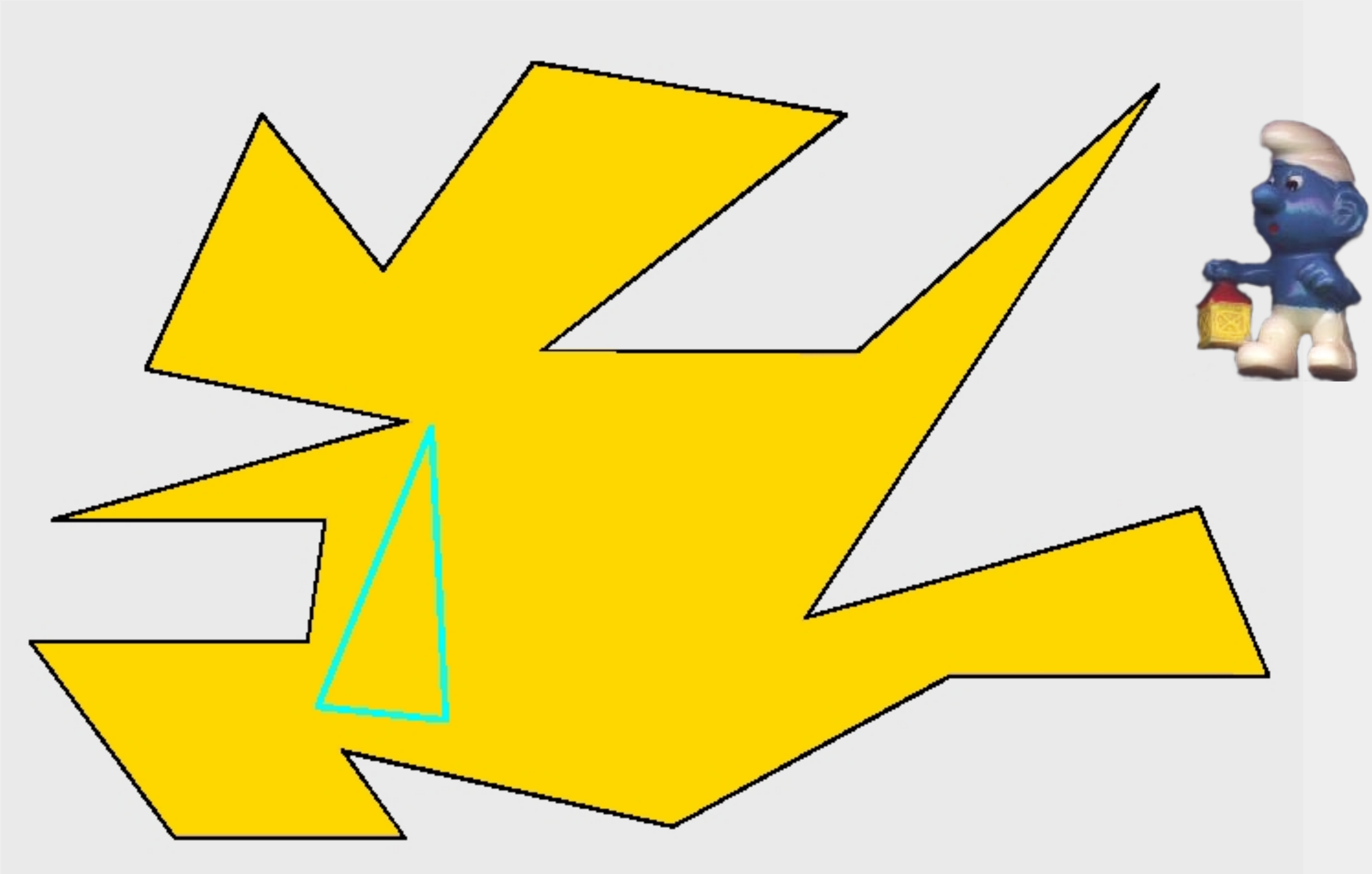
Online Searching with an Autonomous Robot.

Computational Geometry: Theory and Applications, 34 (2), 2006, pp. 102-115.

Art Gallery Problems



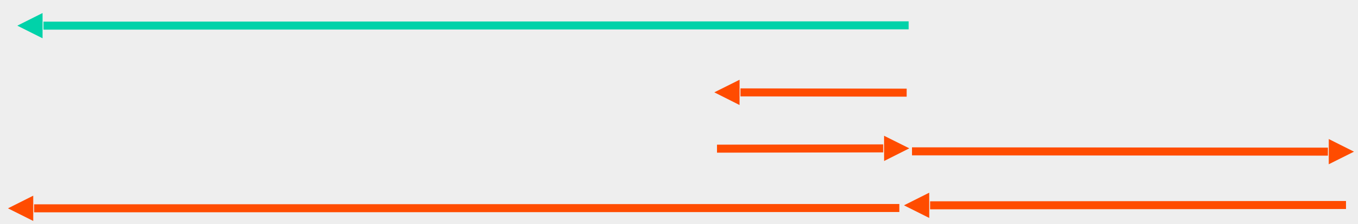
Watchman Problems



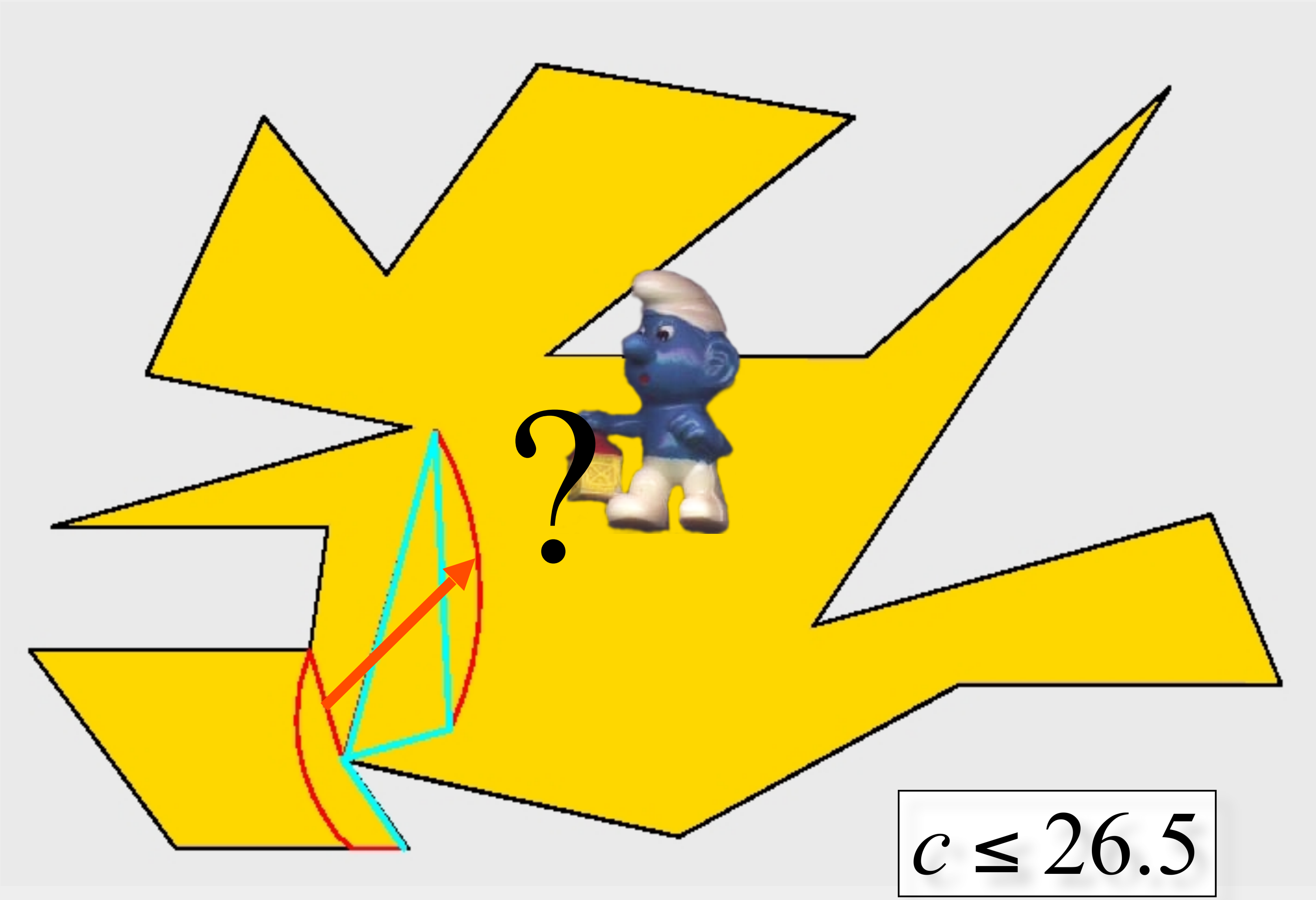
Online Searching

competitive ratio

$$c = \sup \frac{\text{achieved time}}{\text{optimal time}}$$

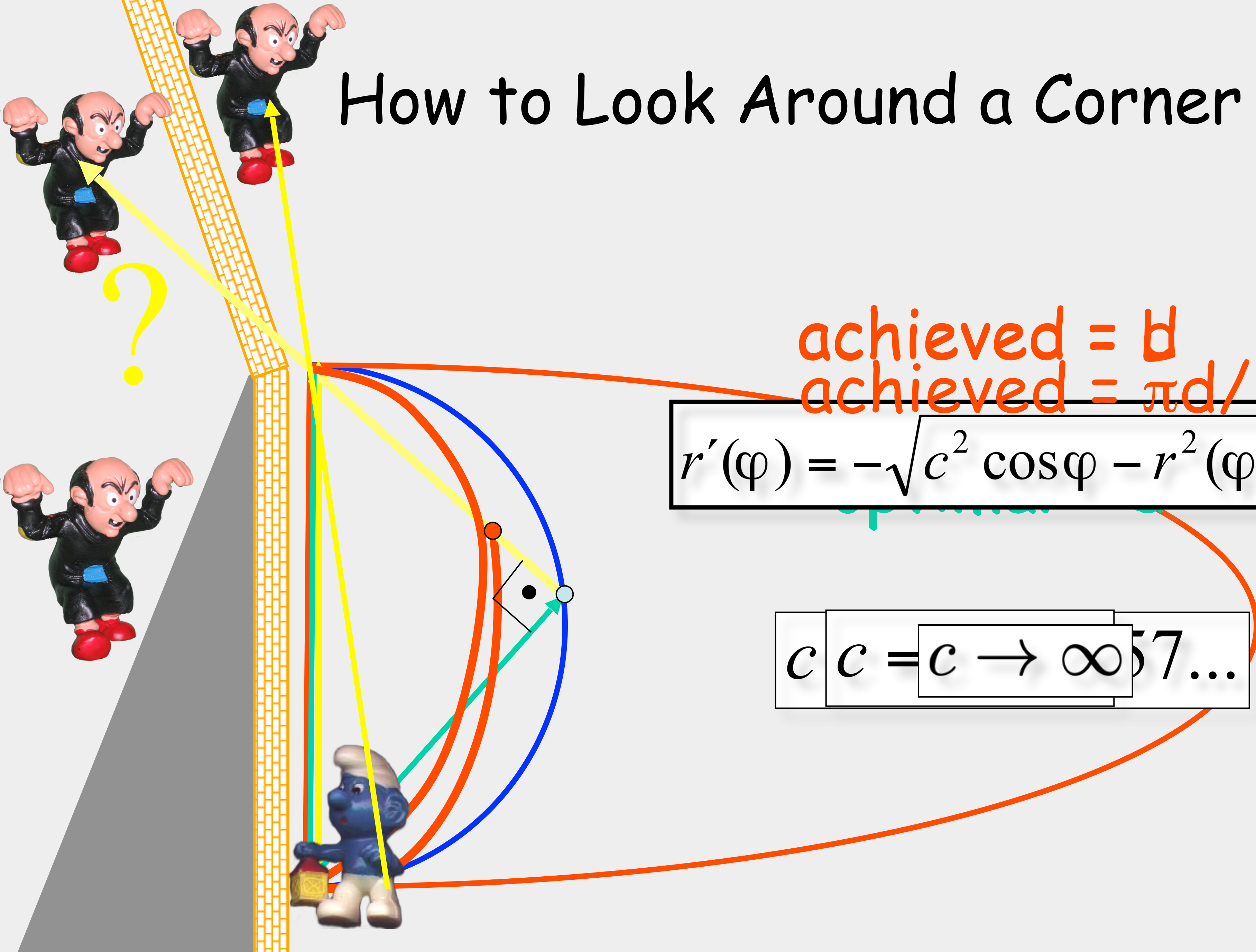


Online Exploration of Simple Polygons



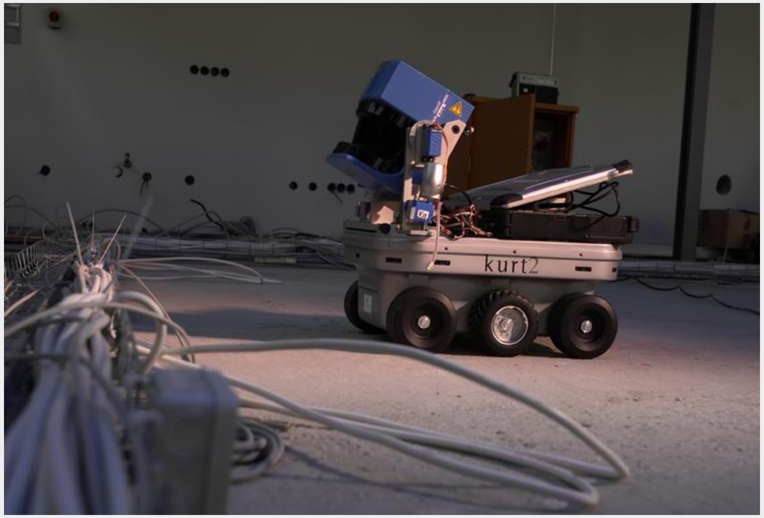
$c \leq 26.5$

How to Look Around a Corner



An Autonomous Robot

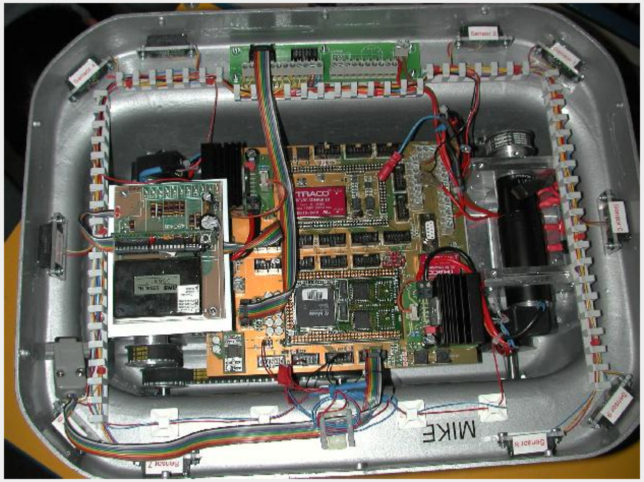
Kurt3D

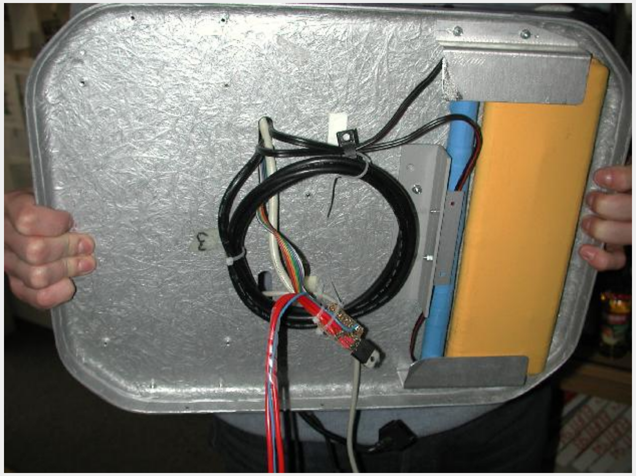


The robot Kurt3D is a leightweight (22.5 kg).

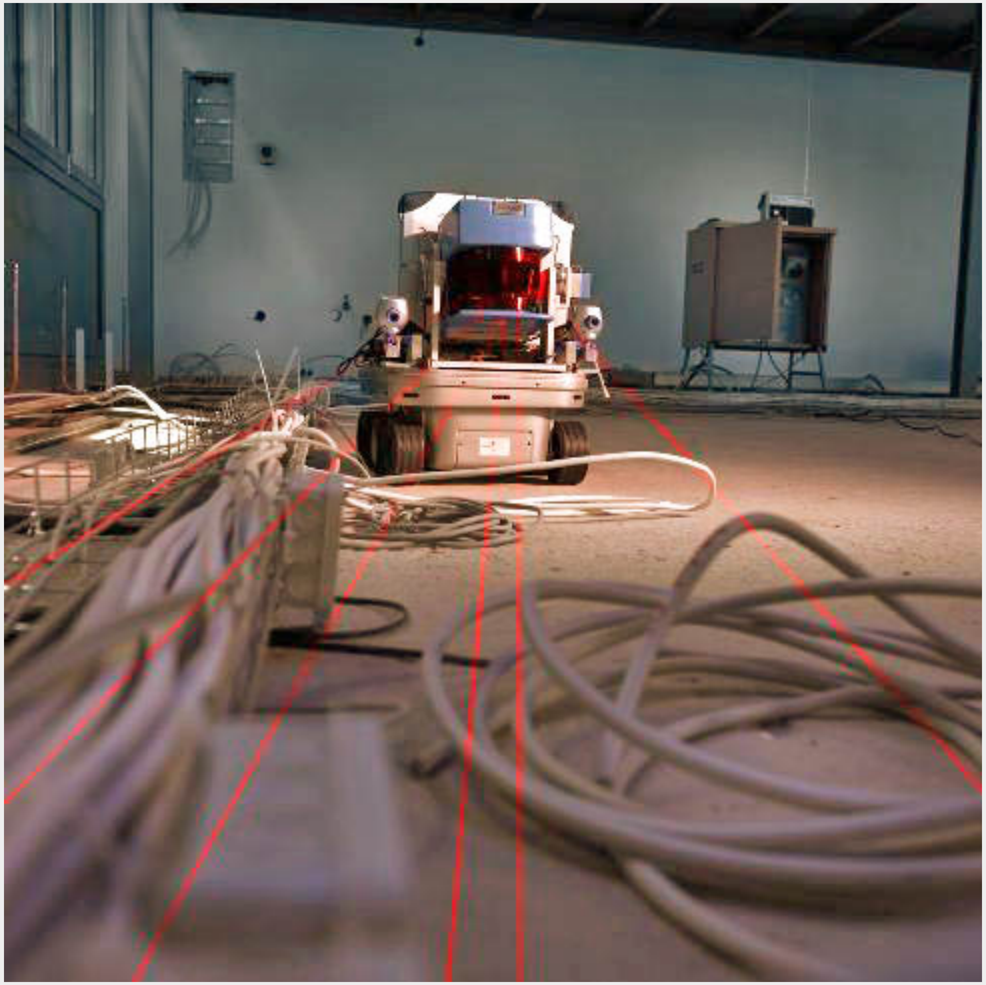
It’s the fastest reliable controlled indoor robot of the world!

- Two 90W (200W) motors
- 48 NiMH a 4500mAh
- C167 Microcontroller
- CAN Controller
- PIII-800 Notebook

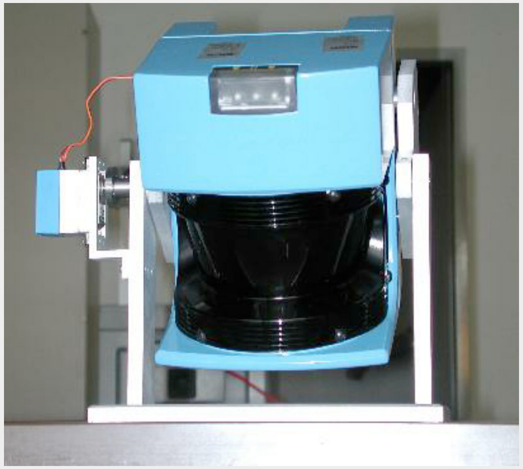




Scan Cost



The AIS 3D Laser Scanner



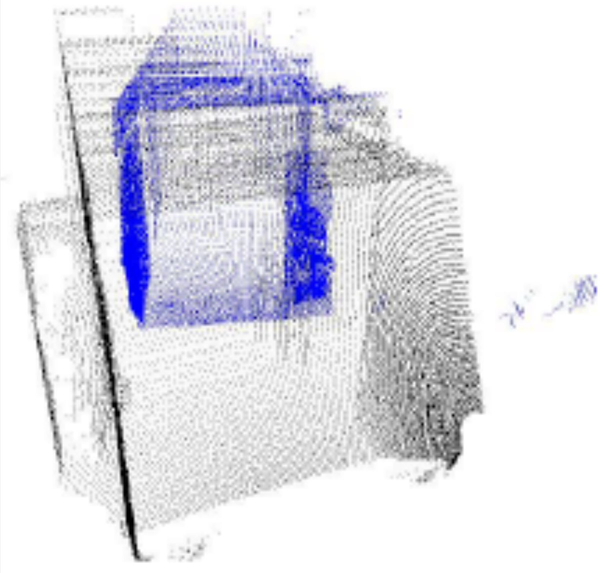
- Based on a regular (e.g., SICK LMS-200) laser scanner
- Relatively cheap sensor
- Controlled pitch motion (120° v)
- Various resolutions and modi, e.g., reflectance measurement
  {181, 361, 721} [h] x {128, …, 500} [v] points
- Fast measurement, e.g., 3.4 sec (181x256 points)

Mounted on mobile robots for 3D collision avoidance and building 3D maps.

The ICP (Scanmatching) Algorithm

Scan registration Put two independent scans into one frame of reference

Iterative Closest Point algorithm [Besl/McKay 1992]



For prior point set M ("model set") and data set D

1. Select point correspondences $w_{i,j}$ in {0,1}
2. Iteratively minimize for rotation **R**, translation **t**

$$E(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} w_{i,j} \, ||\mathbf{m}_i - (\mathbf{R}\mathbf{d}_j + \mathbf{t})||^2$$

- quaternion-based calculation of rotation
- works in 3 translation plus 3 rotation dimensions

-> **6D SLAM**

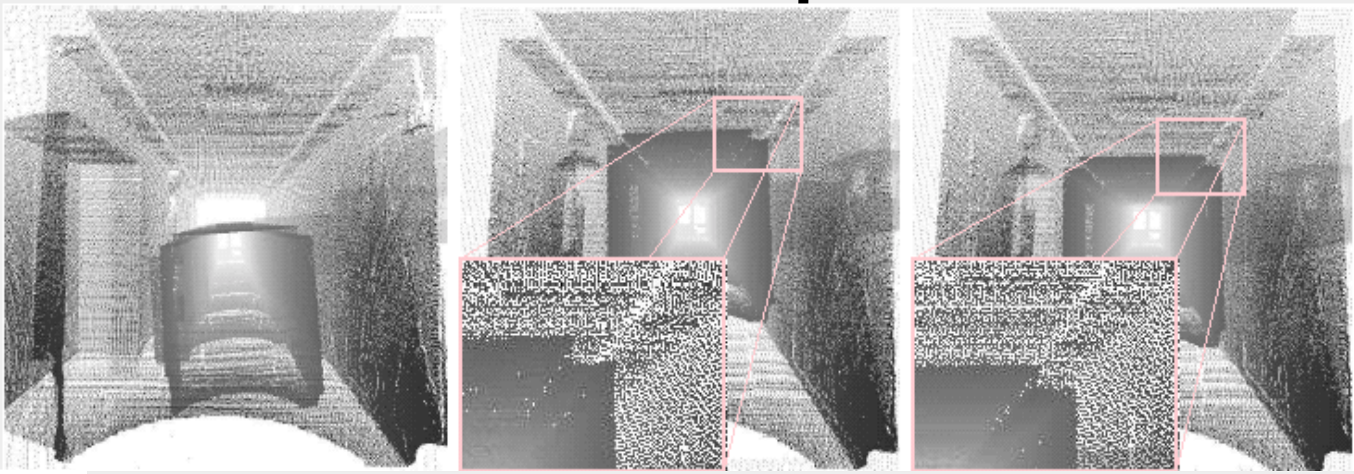
Our **on-line on-board version** of ICP:

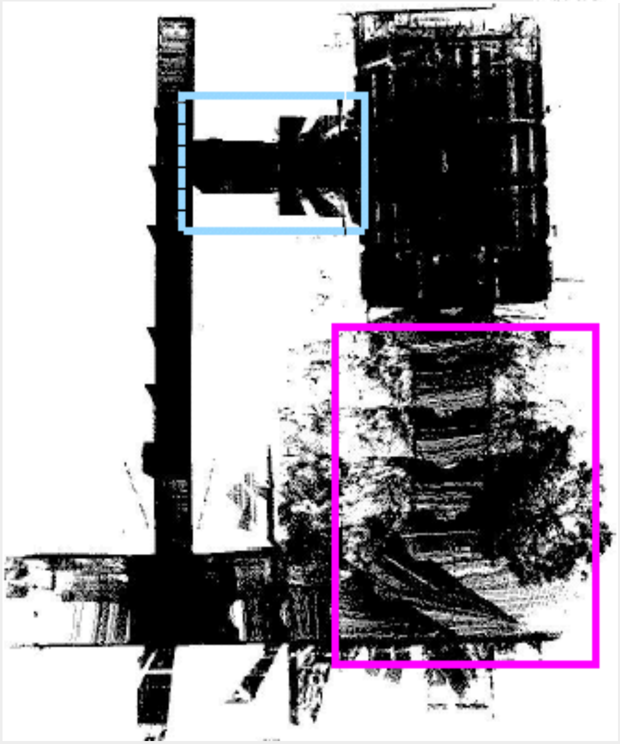
- reduced point sets, representation in Apx-kD-trees
- registers two scans (181x256 pts) in <1.4 sec (P-III-800)

Close-Ups on a Closed Loop Model

Error Distribution:

- Equal distribution (middle)
- Local refinement (right)





Complete Point Model

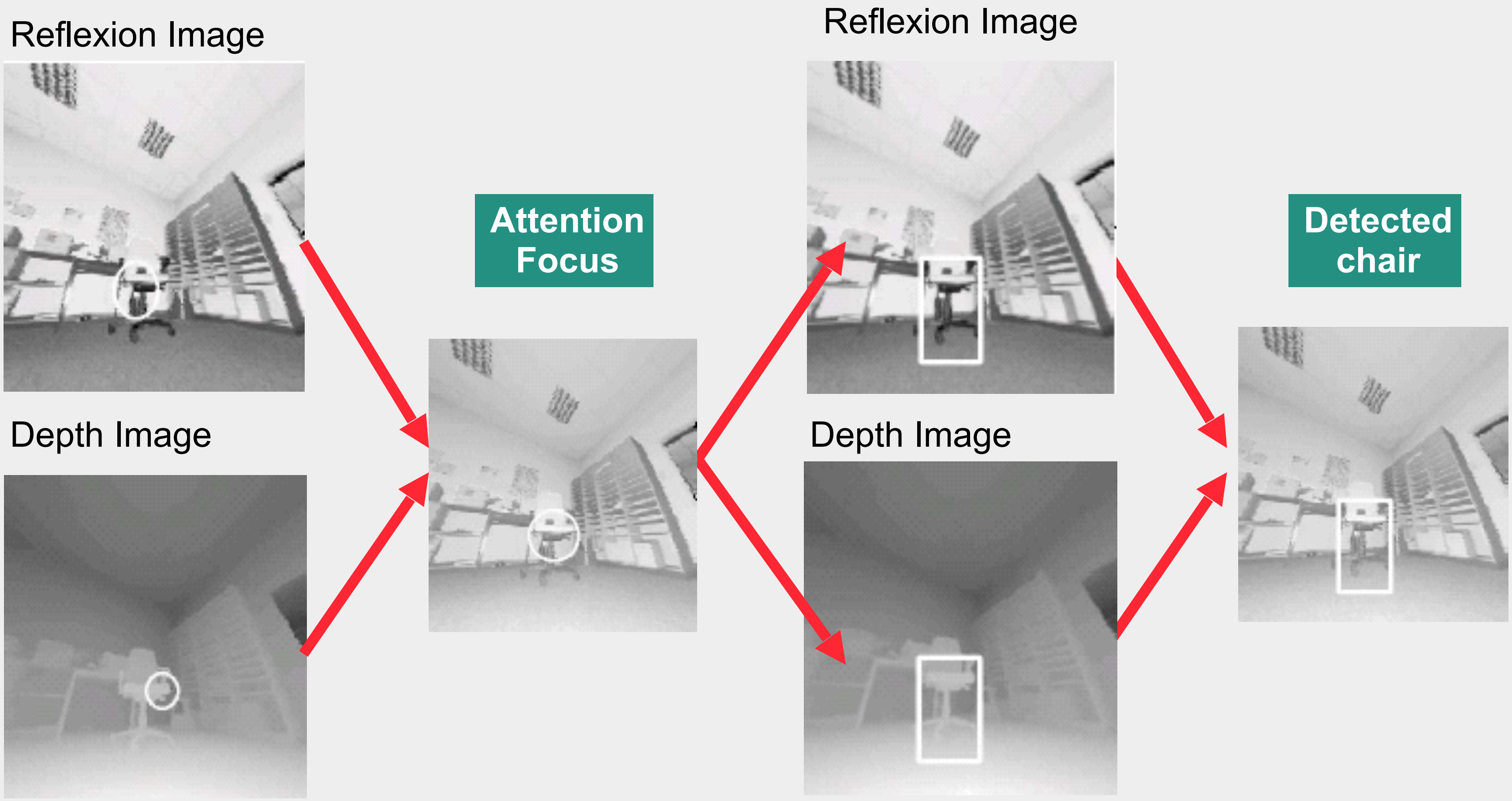
View from top, and two details as viewed from scanner height.

Other Application

Autonomous Mine Mapping (CMU)

Interpreting Maps by Labeling Objects

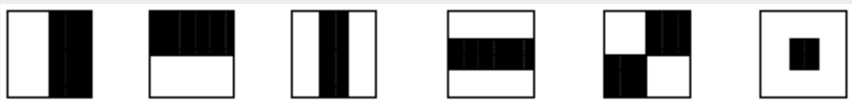
Attention **Classification**

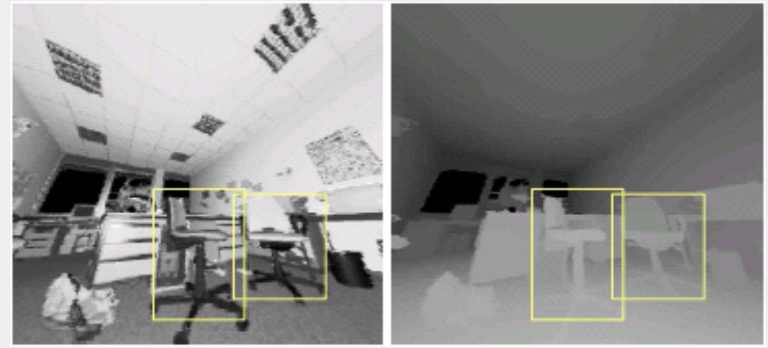


[Frintrop et al, IROS-2004]

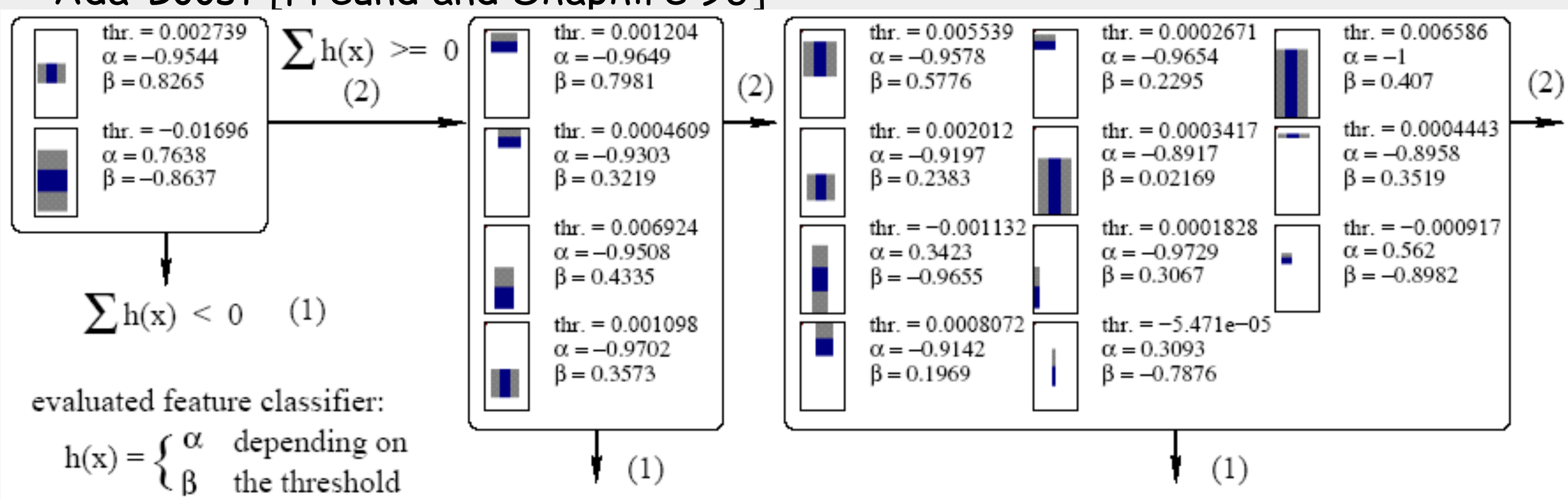
A Learned Cascade of Classifiers

- Learn objects directly from 3D scans
- Simple, efficiently computable features [Viola and Jones 01]





- Learn the combination of these features using Ada-Boost [Freund and Shaphire 96]



[Nüchter et al, IAS-2004]

An Autonomous Robot

Short Distances

$$\min c$$

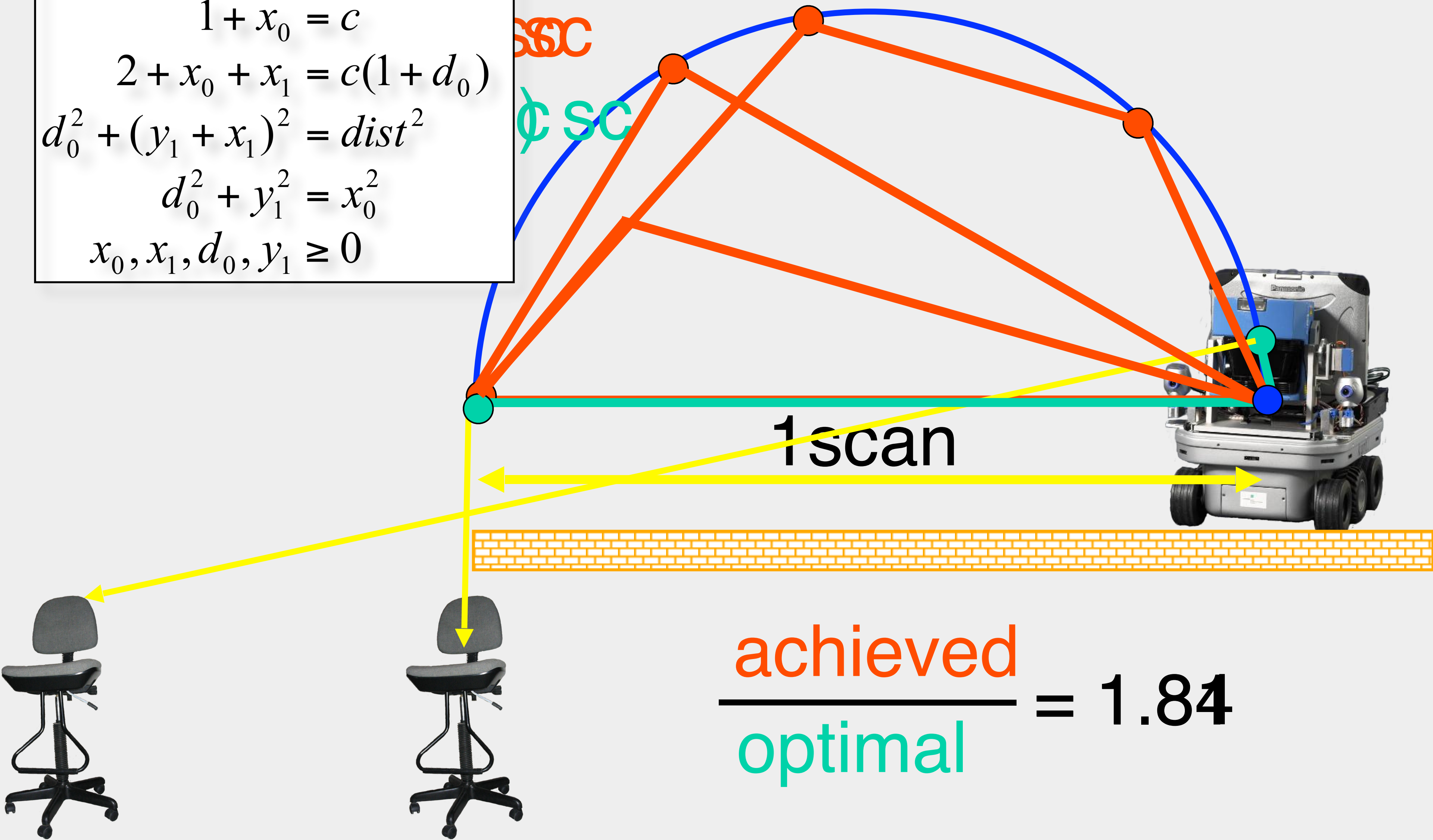
$$1 + x_0 = c$$

$$2 + x_0 + x_1 = c(1 + d_0)$$

$$d_0^2 + (y_1 + x_1)^2 = dist^2$$

$$d_0^2 + y_1^2 = x_0^2$$

$$x_0, x_1, d_0, y_1 \geq 0$$



$$\frac{\text{achieved}}{\text{optimal}} = 1.84$$

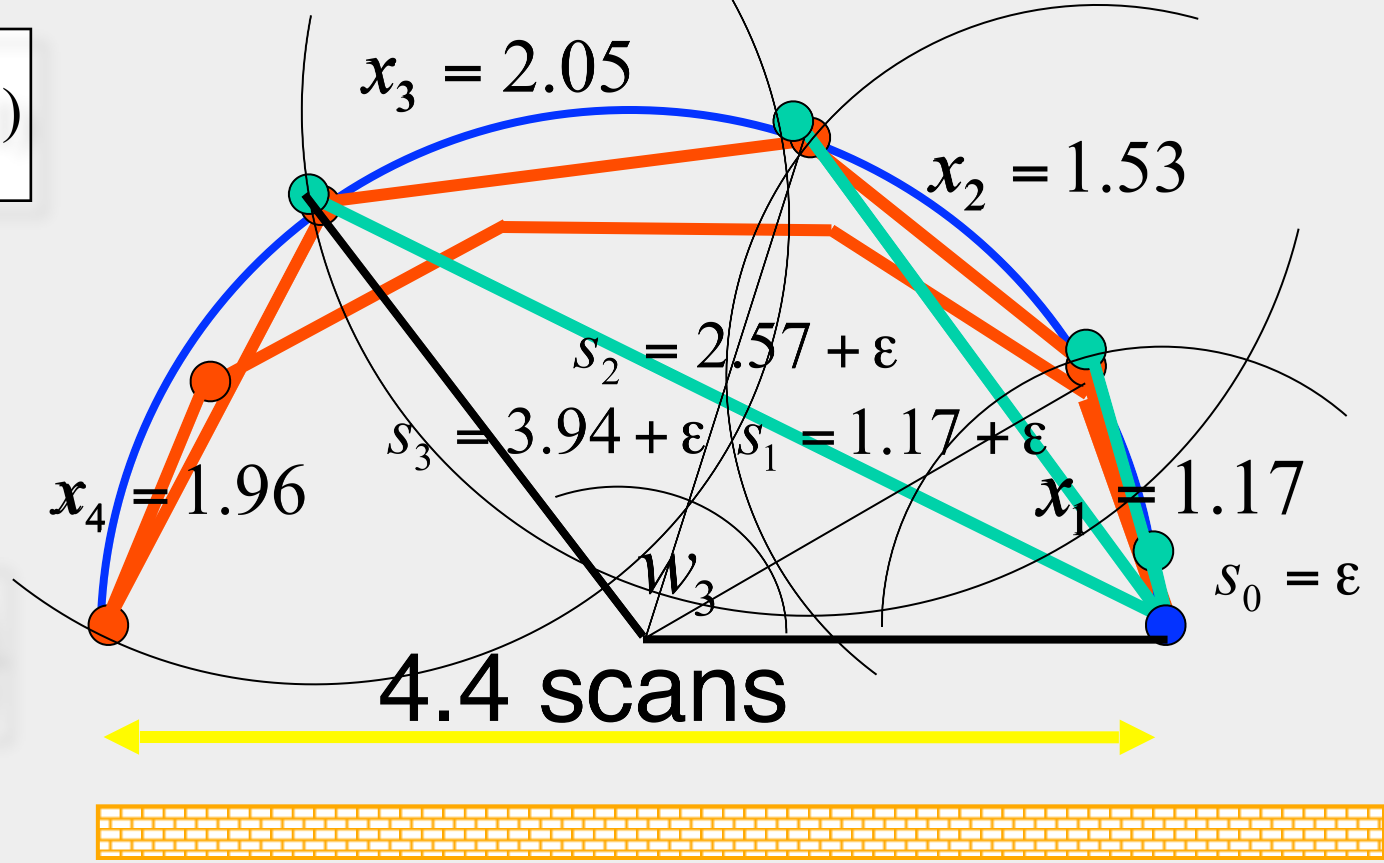
Larger Distances

$$n + \sum_{i=1}^{n} x_i = c(s_{n-1} + 1)$$

$$s_{n-1} = d \sin w_{n-1}$$

$$w_{n-1} = \sum_{i=1}^{n-1} \varphi_i$$

$$\varphi_i = 2 \arcsin\left(\frac{x_i}{d}\right)$$



$$\frac{x_4 + x_3 + x_2 + x_1 + 4}{s_3 + 1} = \frac{\text{achieved}}{\text{optimal}} = 2.17$$

A Lower Bound

Theorem: No strategy can guarantee a competitive factor below 2.

Sketch: Assume factor $c = 2 - \delta$.

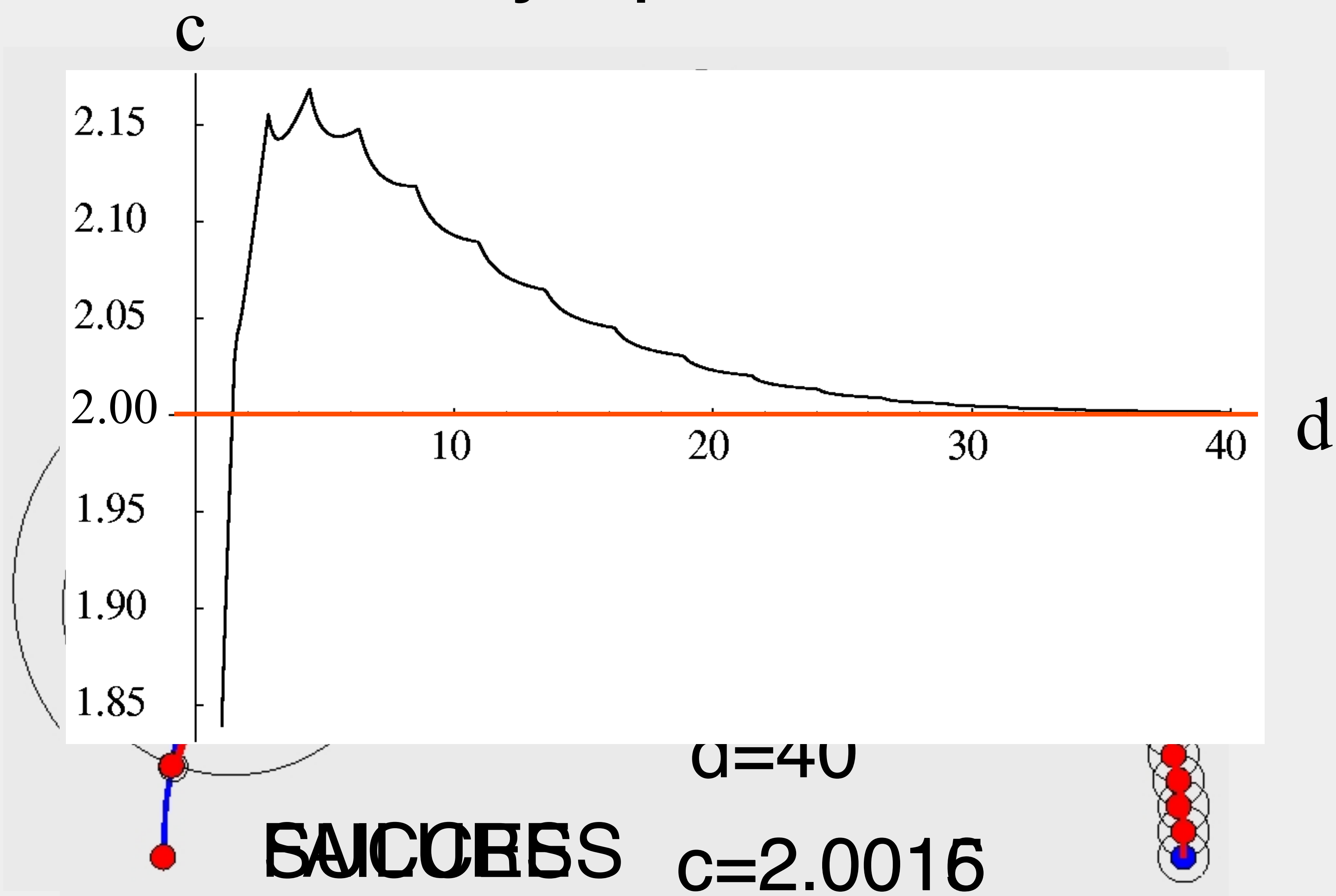
Use induction to show that for the i-th step length, we have $x_i \leq (1 - \delta)^i$.

Then the total distance is bounded by

$$1/\delta,$$

a contradiction.

Asymptotics



An Upper Bound

Theorem: The circle strategy is asymptotically optimal.

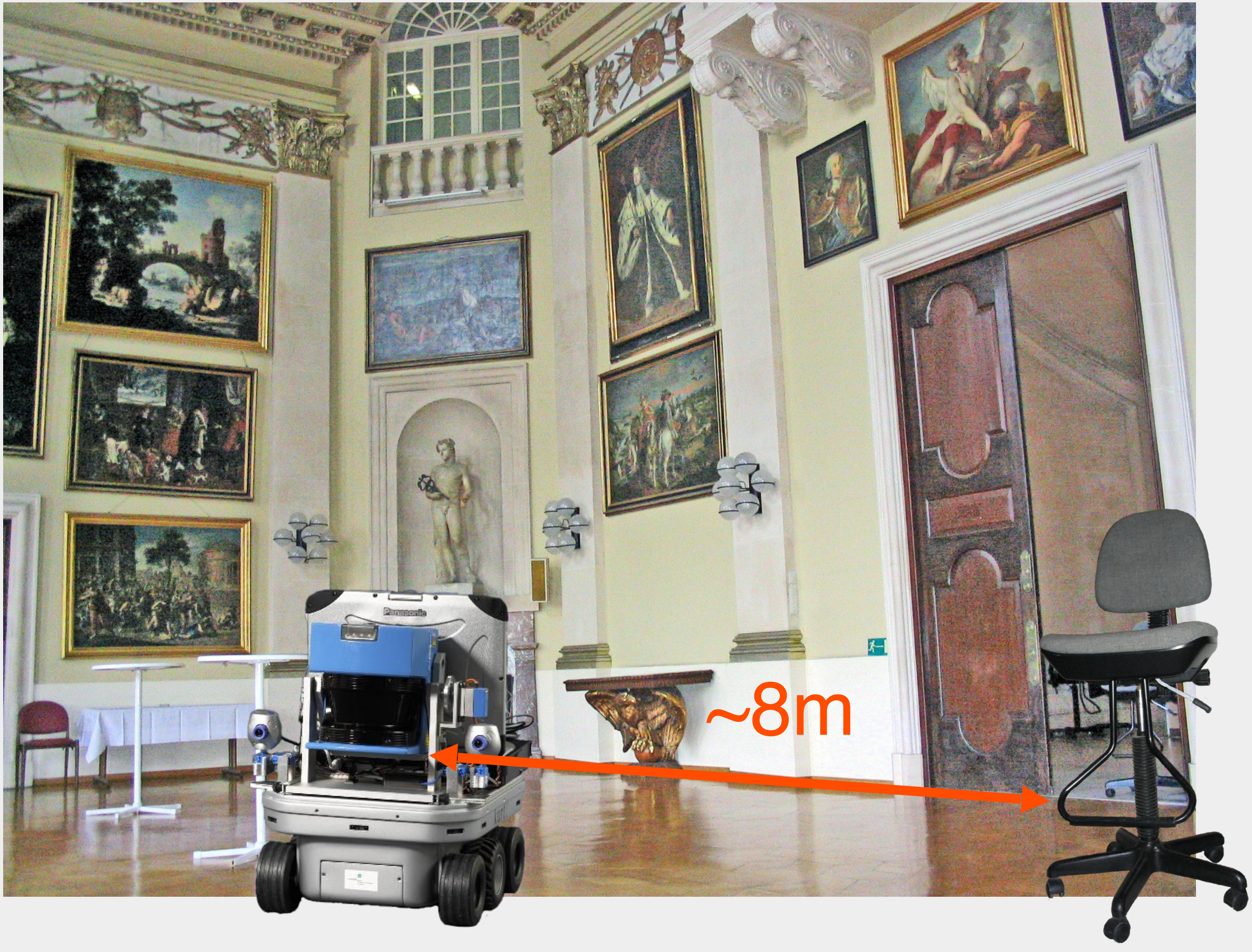
Sketch: Consider a factor $c = 2 + \delta$.

(1) Show that for large circle diameter, the step length grows expontially, as long as the direction is close to being orthogonal to the wall.

(2) Show that in this manner, a large step length can be reached. More specific, show that an average step length of at least 5 can be achieved at some point.

(3) Show that once the average step length is at least 5, it stays above 5. Thus, any necessary total distance can be traveled.

Practical Application



ELSEVIER

Available online at www.sciencedirect.com

SCIENCE DIRECT

Computational Geometry 34 (2006) 102–115

Computational Geometry
Theory and Applications

www.elsevier.com/locate/comgeo

Online searching with an autonomous robot

Sándor P. Fekete [a,*], Rolf Klein [b], Andreas Nüchter [c]

[a] *Institute for Mathematical Optimization, Braunschweig University of Technology, D-38106 Braunschweig, Germany*
[b] *Institute of Computer Science, University of Bonn, D-53117 Bonn, Germany*
[c] *Institute of Computer Science, University of Osnabrück, D-49069 Osnabrück, Germany*

Received 9 October 2004; received in revised form 24 June 2005; accepted 9 August 2005

Available online 27 October 2005

Part 1.2:
Exploring rectilinear polygons

Computational Geometry 43 (2010) 148–168

Contents lists available at ScienceDirect

Computational Geometry: Theory and Applications

www.elsevier.com/locate/comgeo

ELSEVIER

Polygon exploration with time-discrete vision

Sándor P. Fekete *, Christiane Schmidt [1]

Department of Computer Science, Technische Universität Braunschweig, D-38106 Braunschweig, Germany

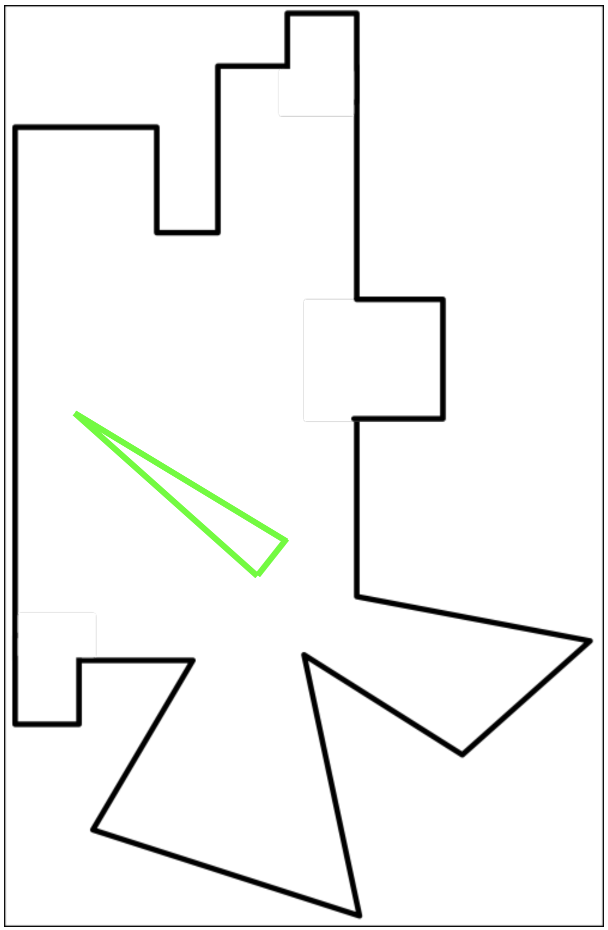
A R T I C L E   I N F O

Article history:
Received 26 October 2008
Accepted 16 June 2009
Available online 21 June 2009

A B S T R A C T

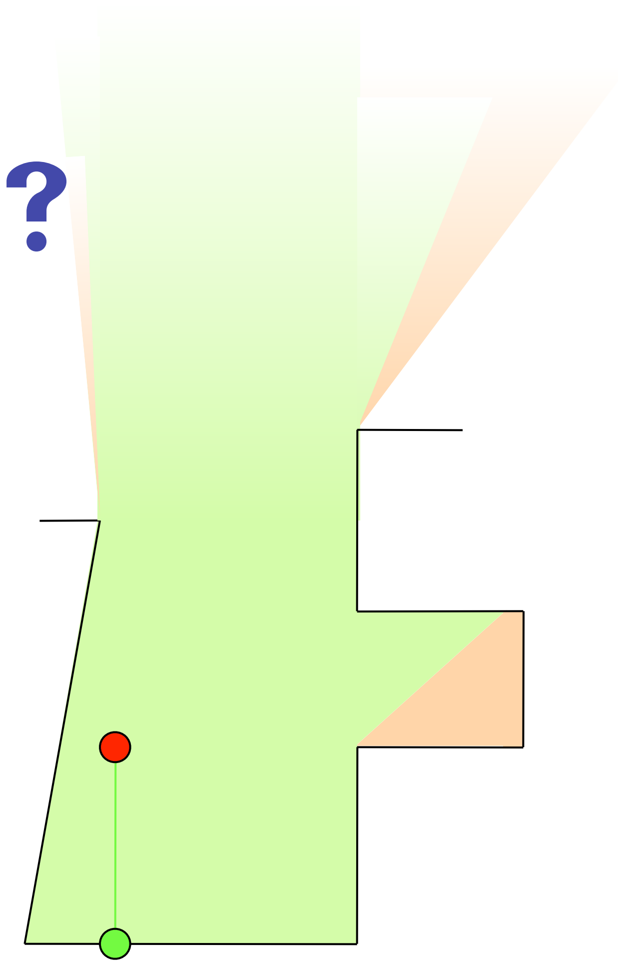
With the advent of autonomous robots with two- and three-dimensional scanning capabilities, classical visibility-based exploration methods from computational geometry have gained in practical importance. However, real-life laser scanning of useful accuracy does not allow the robot to scan conti…

Motivation

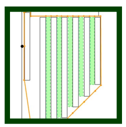


- Watchman problem
- Online, continuous vision:
  - optimum watchman route (L_1-metric) in simple rectilinear polygons (Deng et al.)
  - c=26.5 in simple polygons (Hoffmann et al.)
  - No competitive online algorithm for polygons with holes (Albers et al.)

Motivation



- Autonomous robot without continuous vision (scan costs)
- Watchman route
- Online problem
- Several classes of polygons
- Is it possible to achieve a competitive strategy?



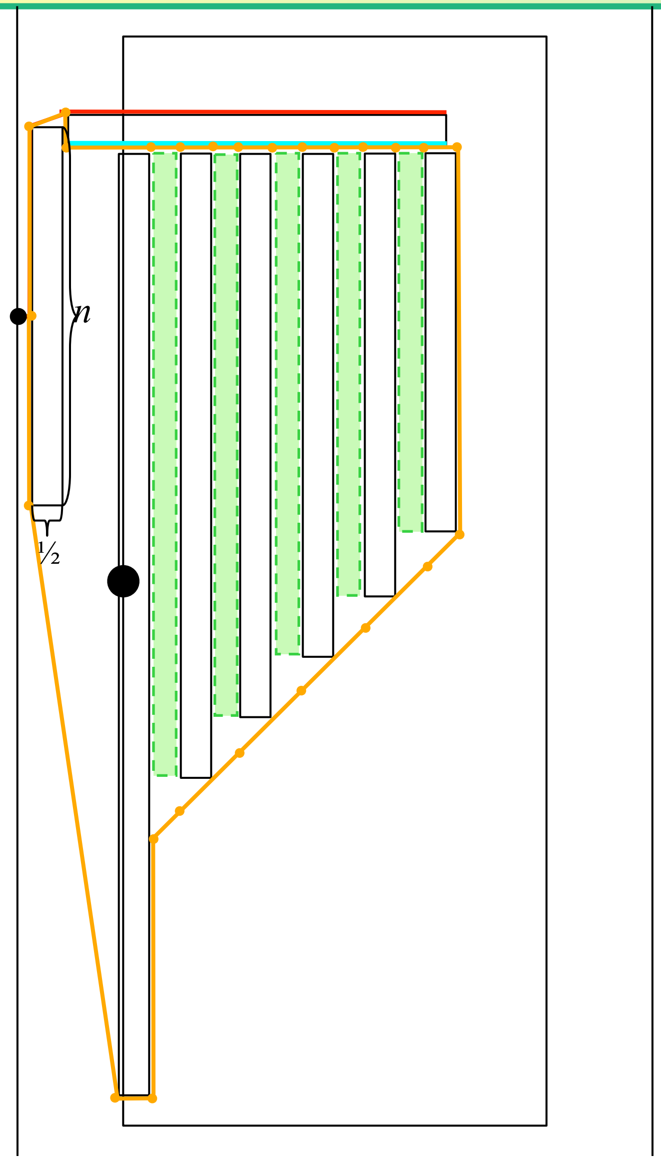
Polygons with Holes

Polygons with holes

Proposition:

There is no strategy that achieves a bounded competitive ratio for the watchman problem with scan costs in case of a polygon with holes/obstacles.
This statement holds even if the polygon is rectilinear.

Proof of the proposition



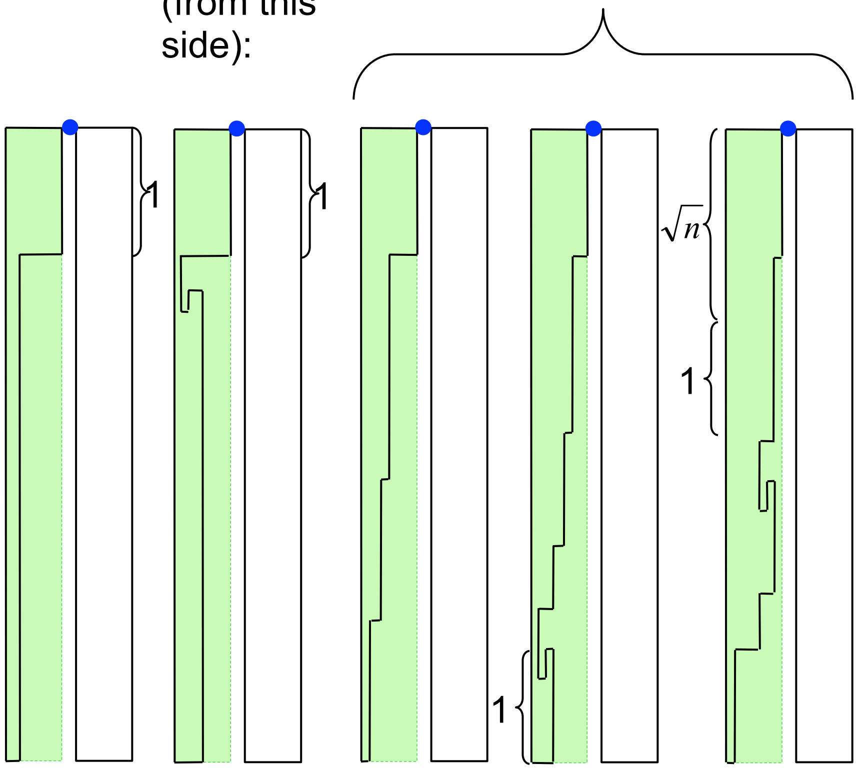
- Show: competitive ratio $\Omega(\sqrt{n})$
- Polygon with obstacles (panpipe)
- Further obstacles: placed depending on the strategy of the robot

Proof of the proposition

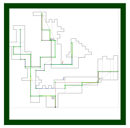
The robot traverses the row:

The robot does not turn into the row (from this side):

The robot walks into the row, but turns back:

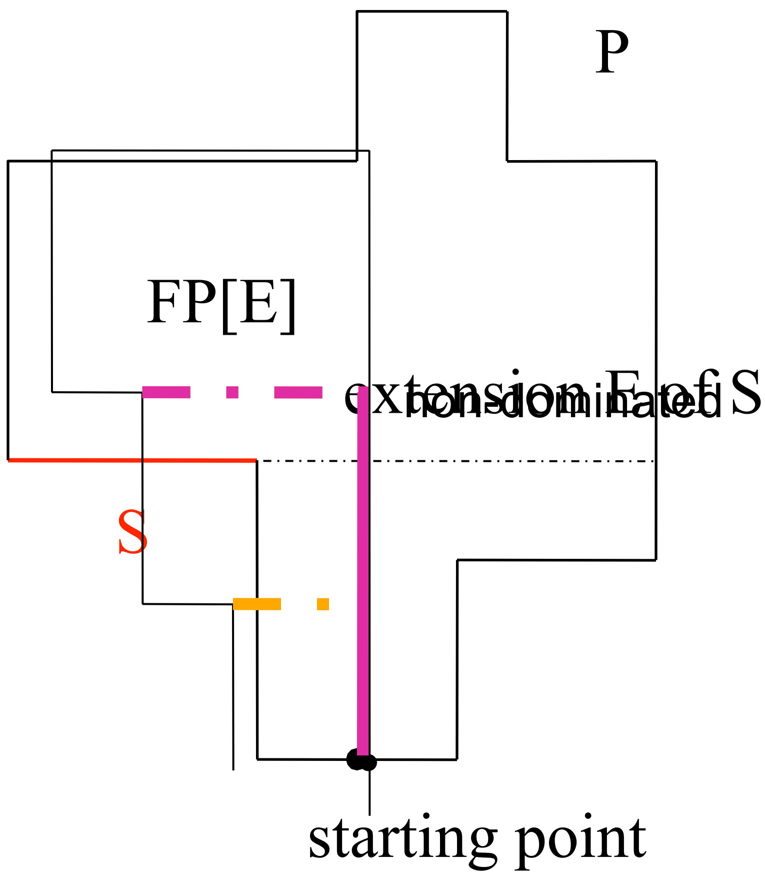


- The shape of the inserted objects depends on the path of the robot.



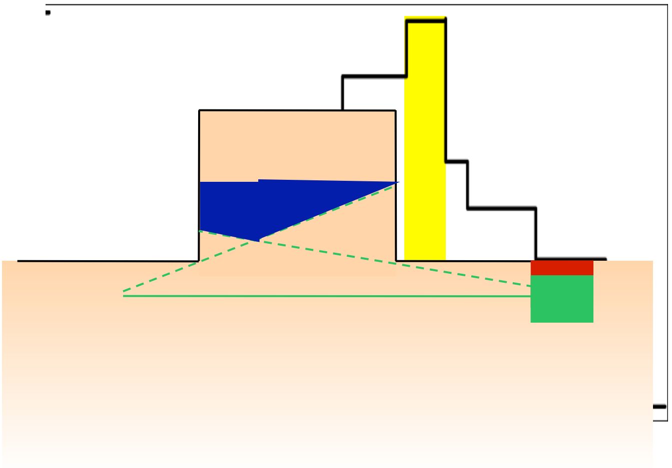
A Competitive Strategy for Simple Rectilinear Polygons

Extensions



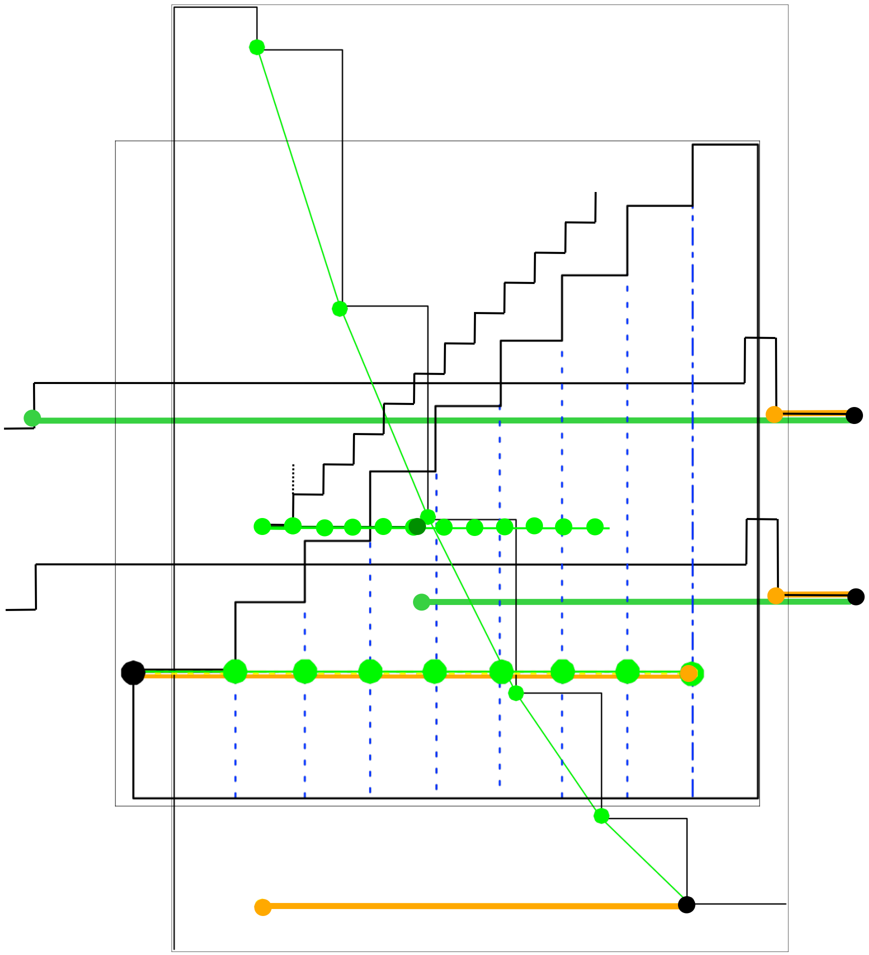
- Two subpolygons
- Necessary and essential extensions
- Advantage in rectilinear polygons

A competitive strategy for simple rectilinear polygons



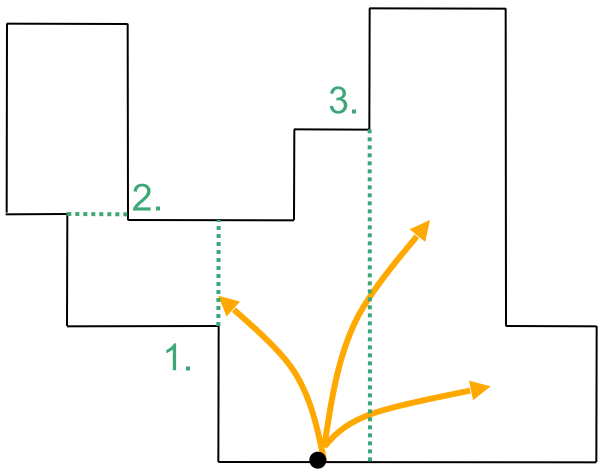
- Problem with niches
- It is necessary to limit the number of scan points

A competitive strategy for simple rectilinear polygons



- Minimum side length a
- Consider the distance to the next corner (reflex vertex): walk beyond the corner if the distance to it is “short”
- Adapt the step length of the robot to the minimal necessary step length
- Move to the projection of a corner and not to the corner itself
- Do not scan on each necessary extension

Order of extensions

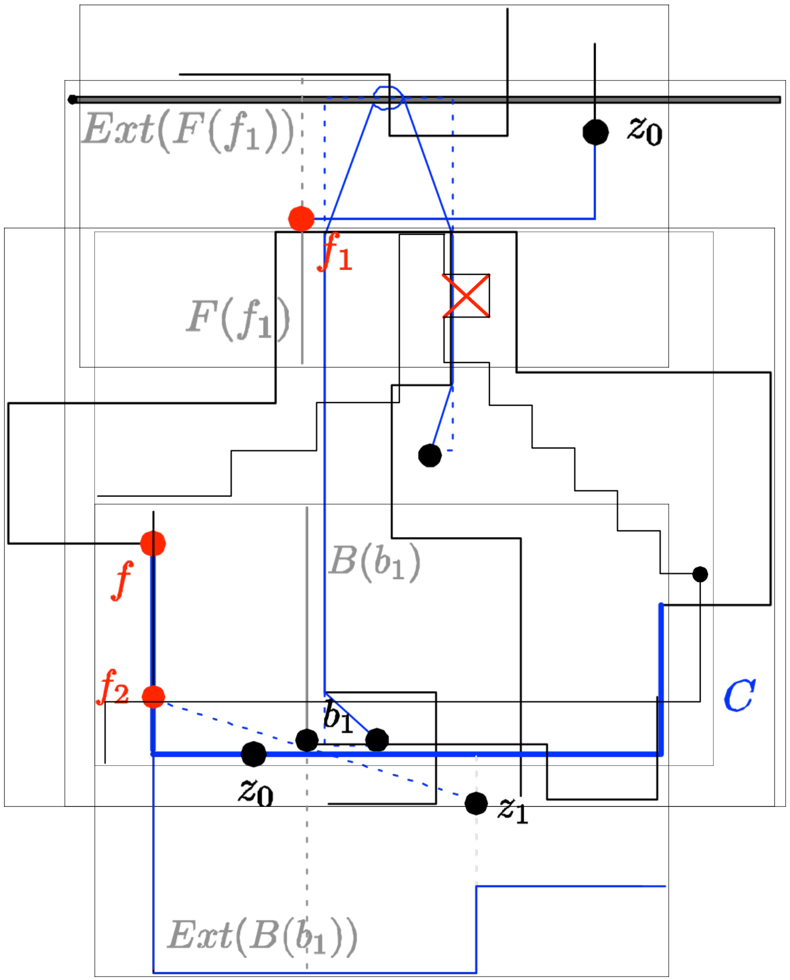


Optimum?

- GREEDY-ONLINE algorithm for a robot with continuous vision.
- Based on a proposition of Chin and Ntafos:

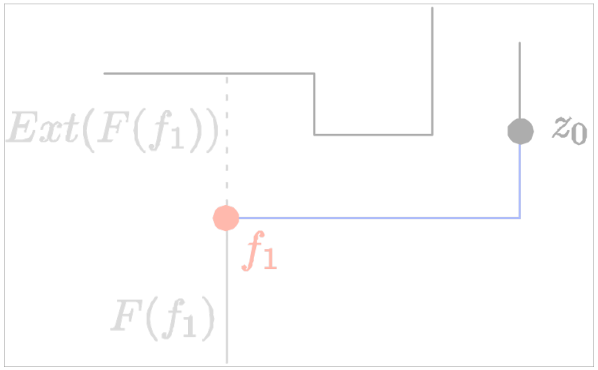
  Any optimum watchman route in P, a simple rectilinear polygon, will have to visit the essential edges in the order in which they appear on the boundary of P’ (the new polygon obtained by removing the “non-essential” portions of the polygon).
- Transfer of this proposition.

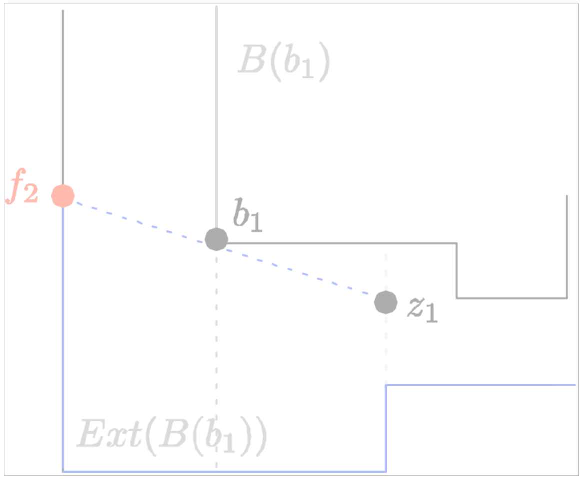
GREEDY-ONLINE algorithm



- Taut-Thread-Principle
- Consider the contiguous part of the boundary that was already visible from some point of the route
- Either f is a 270° corner or a corner blocks the sight such as only f^- is visible

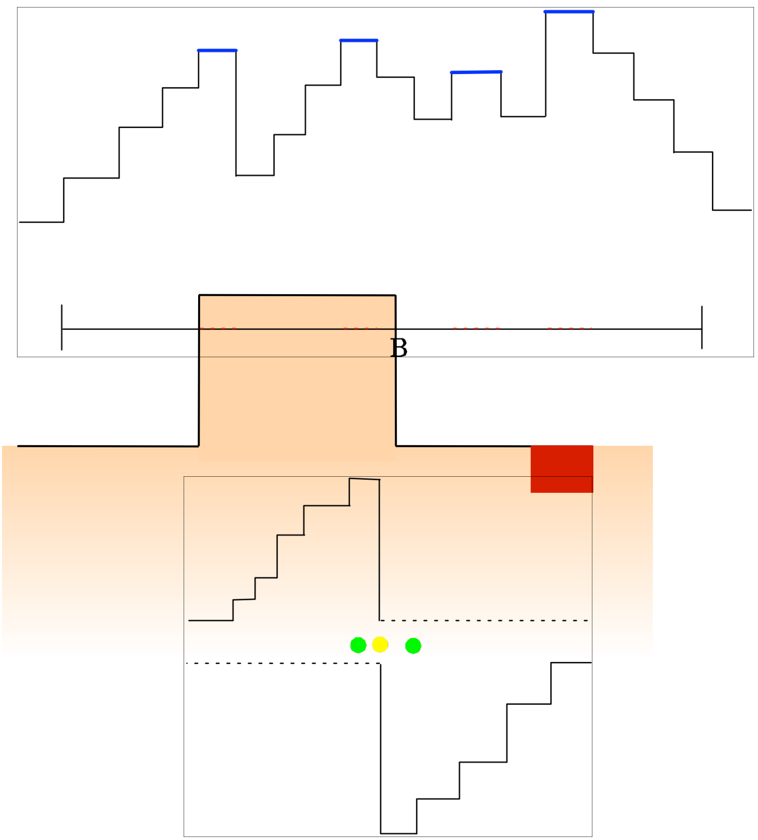
A competitive strategy for simple rectilinear polygons





- Extensions of the GREEDY-ONLINE algorithm
- Interval case vs. extension case
- Reaching the extension on an axis-parallel path without a change of direction is possible/impossible
- In all cases of the case differentiation:
  - In case the robot runs beyond the extension: the robot is (is not) able to cover the total planned length
  - Positive line creation vs. negative line creation

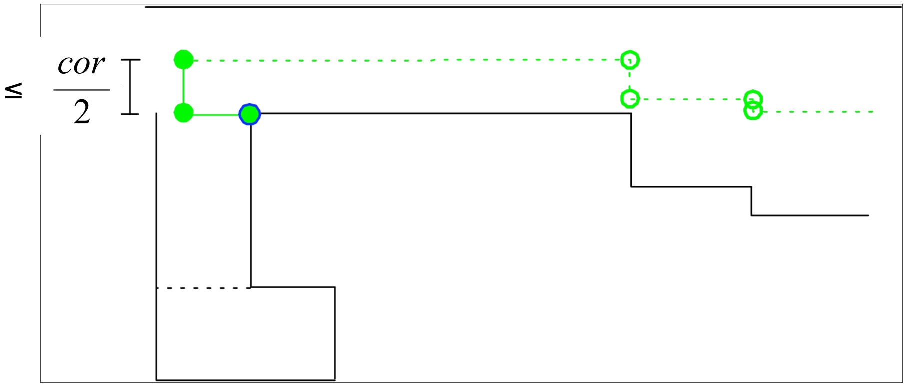
Binary search in the strategy



- If the optimum needs k scans in an interval, the robot which uses the strategy will need maximum
  - k binary searches (if k is given) or
  - 2k binary searches if the NVRs may appear on two sides

- Non-visible region (NVR): An area in which the parts of the boundary, which would be visible by simply passing them with continuous vision, are not yet completely visible.
- Discover passed non-visible regions with binary search.

Turn adjustments



- The optimum may have the opportunity to turn off before the robot, following the strategy, does.
- The robot may discover a corridor inside a non-visible region.

Ø Adjustments to have the best basic position for the next turn

- Minimum corridor width a_k

The strategy

$a \leq 1$:

A. An axis-parallel move to E is possible without a turn

- e≥2a+1: interval case

  Let d_i be the actual distance to the perpendicular of the next counterclockwise extension

  – If d_i >2a+1, move to the perpendicular of the corner

  – If $d_i \leq 2a+1$: If $d_i > a$: cover a distance of 2d_i+1

    If $d_i \leq a$: cover a distance of 2a+1

  Apply binary search if necessary, that means, if non-visible regions appear.

  – If no corner appears on the counterclockwise side, move directly to E.

  In case we run beyond E with a step of length 2d_i+1/2a+1:

  i. If we do not cover the total distance, because of the boundary: Run as far as possible, go back to E, move back in steps of length 1, apply binary search for NVRs (on the counterclockwise side till E, on both sides beyond E) and if a corridor is identified, use it and make turn adjustments

  ii. If we may cover the total distance:

     I. negative line creation: Apply binary search, if a corridor is discovered inside a NVR, use it and make turn adjustments.

     II. Positive line creation: Go back to E, move back in steps of length 1, apply binary search and search for a corridor and the critical extension, make turn adjustments.

- e<2a+1: extension case

  Cover a distance of 2e+1. In case:..(i., ii.)

The strategy

$a \leq 1$:

A. An axis-parallel move to E is possible without a turn

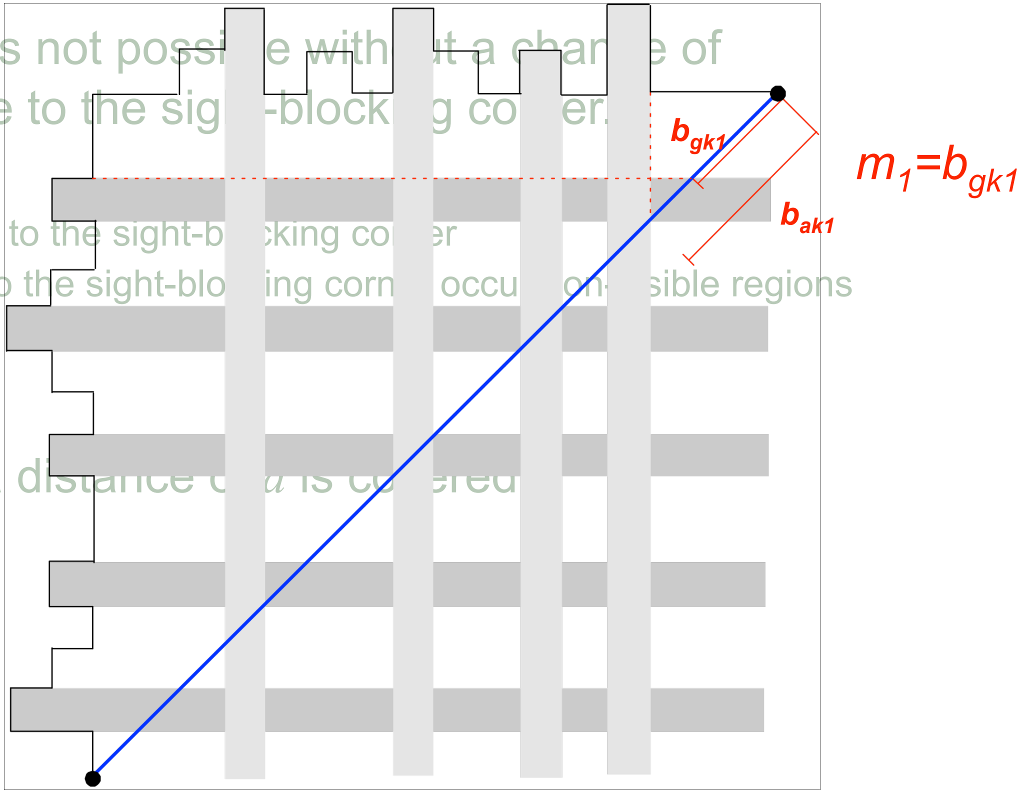
- e≥2a+1: interval case
- e<2a+1: extension case

B. An axis-parallel move to E is not possi e with ut a char e of direction: Let b_i be the distance to the sig -block g co er.

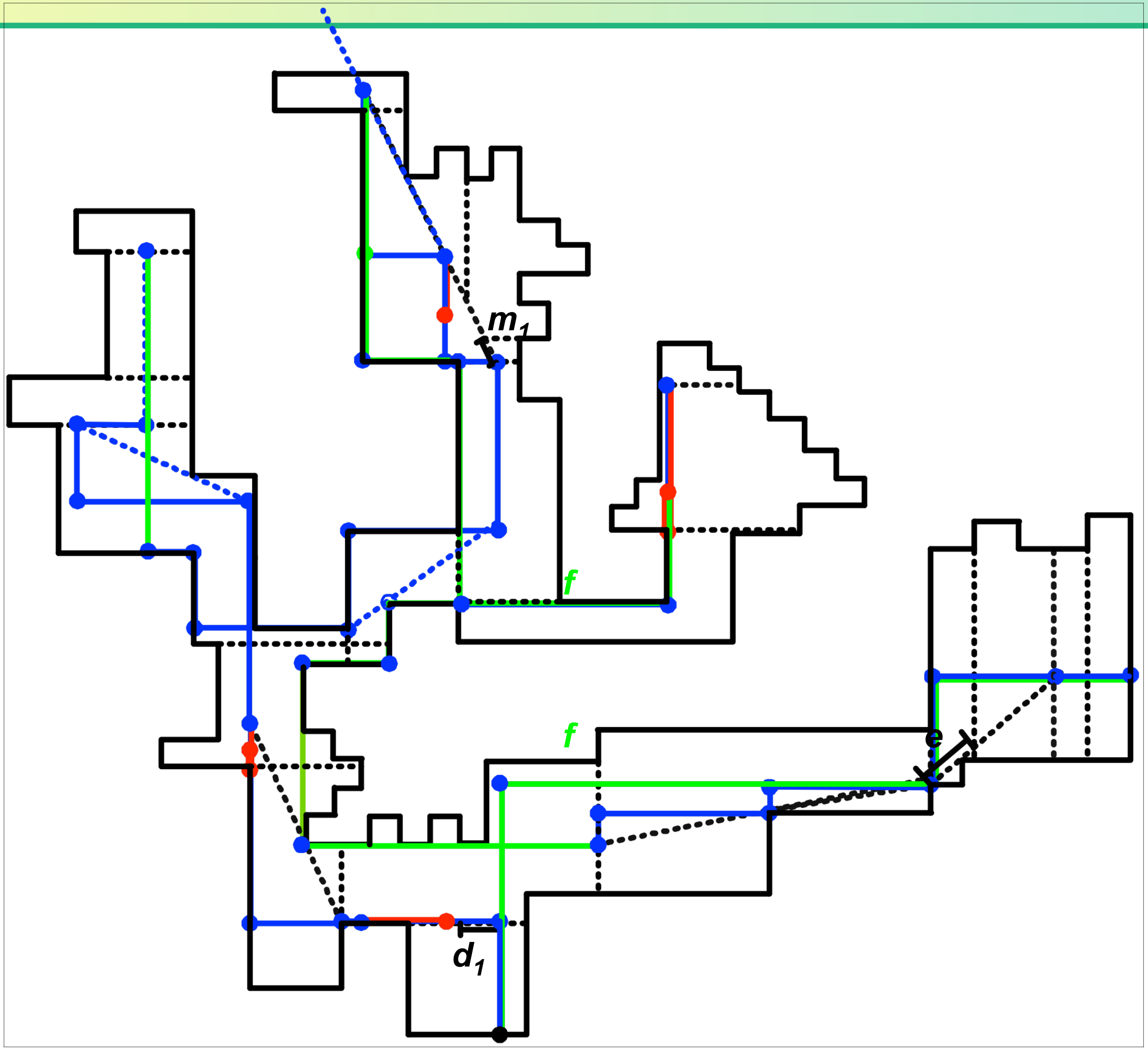
- e ≥a+1: interval case
  - No non-visible region up to the sight-b cking co er
  - Along the boundary up to the sight-blo ing corn occu on- sible regions
- e<a+1: extension case

a > 1:

Similar; with scans every time a distance c a is co ered

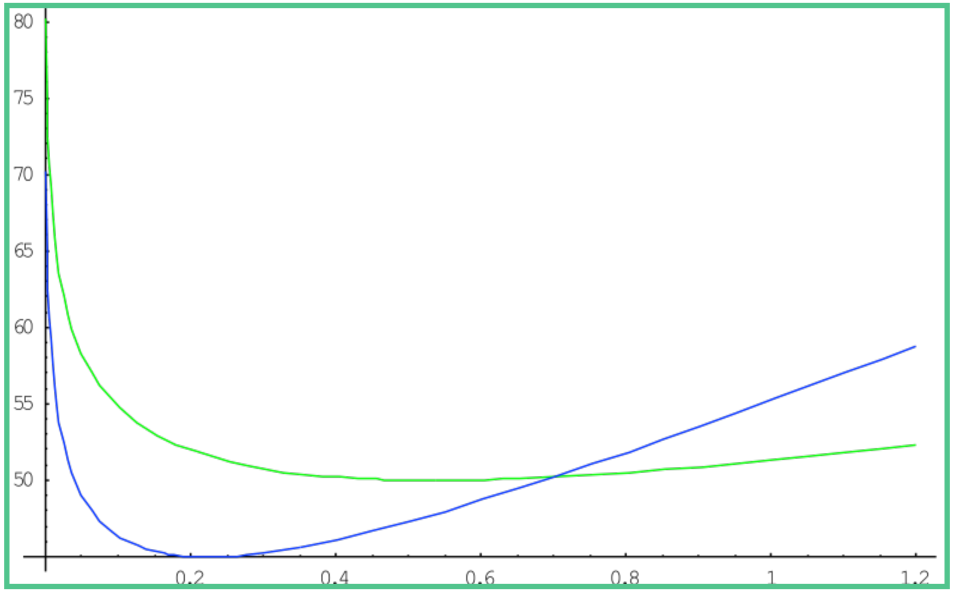


An example



starting point

The competitive ratio of the strategy



a	upper bound for c
1	55.2294
0.8	51.8168
0.7	50.2083
0.5	50.0000
0.1	54.8000
0.01	67.0336
0.0001	93.4919
0.000001	120.0661

- If we assume $a = a_k$:

$$c \le \begin{cases} 8a + 34 + 4\dfrac{\ln\left(\dfrac{2a+3}{a}\right)}{\ln(2)}, & 0 \le a < 0.70043 \\ 20a + 24 + 4\dfrac{\ln\left(\dfrac{4a+3}{a}\right)}{\ln(2)}, & 0.70043 \le a \le 1 \end{cases}$$

Computational Geometry 43 (2010) 148–168

Contents lists available at ScienceDirect

Computational Geometry: Theory and Applications

www.elsevier.com/locate/comgeo

ELSEVIER

Polygon exploration with time-discrete vision

Sándor P. Fekete *, Christiane Schmidt [1]

Department of Computer Science, Technische Universität Braunschweig, D-38106 Braunschweig, Germany

ARTICLE INFO

Article history:
Received 26 October 2008
Accepted 16 June 2009
Available online 21 June 2009

ABSTRACT

With the advent of autonomous robots with two- and three-dimensional scanning capabilities, classical visibility-based exploration methods from computational geometry have gained in practical importance. However, real-life laser scanning of useful accuracy does not allow the robot to scan contin

Part 1.3:
Searching with turn cost

Available online at www.sciencedirect.com
ScienceDirect

ELSEVIER

Theoretical Computer Science 361 (2006) 342–355

Theoretical Computer Science

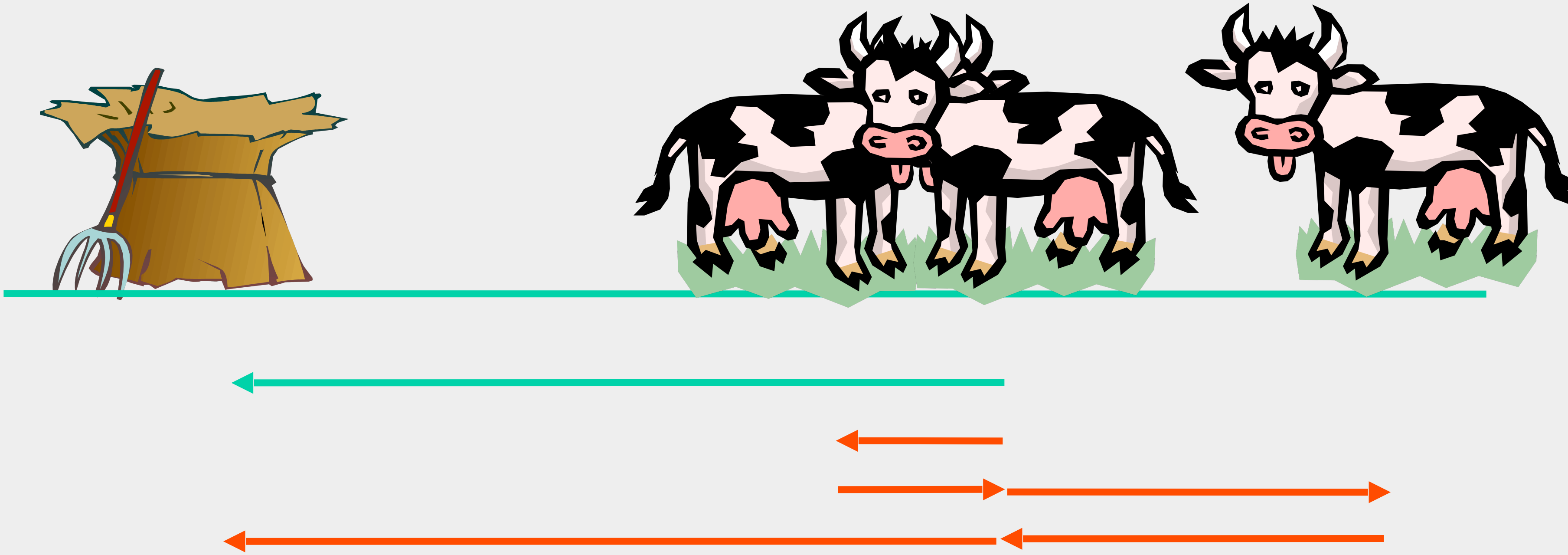
www.elsevier.com/locate/tcs

Online searching with turn cost

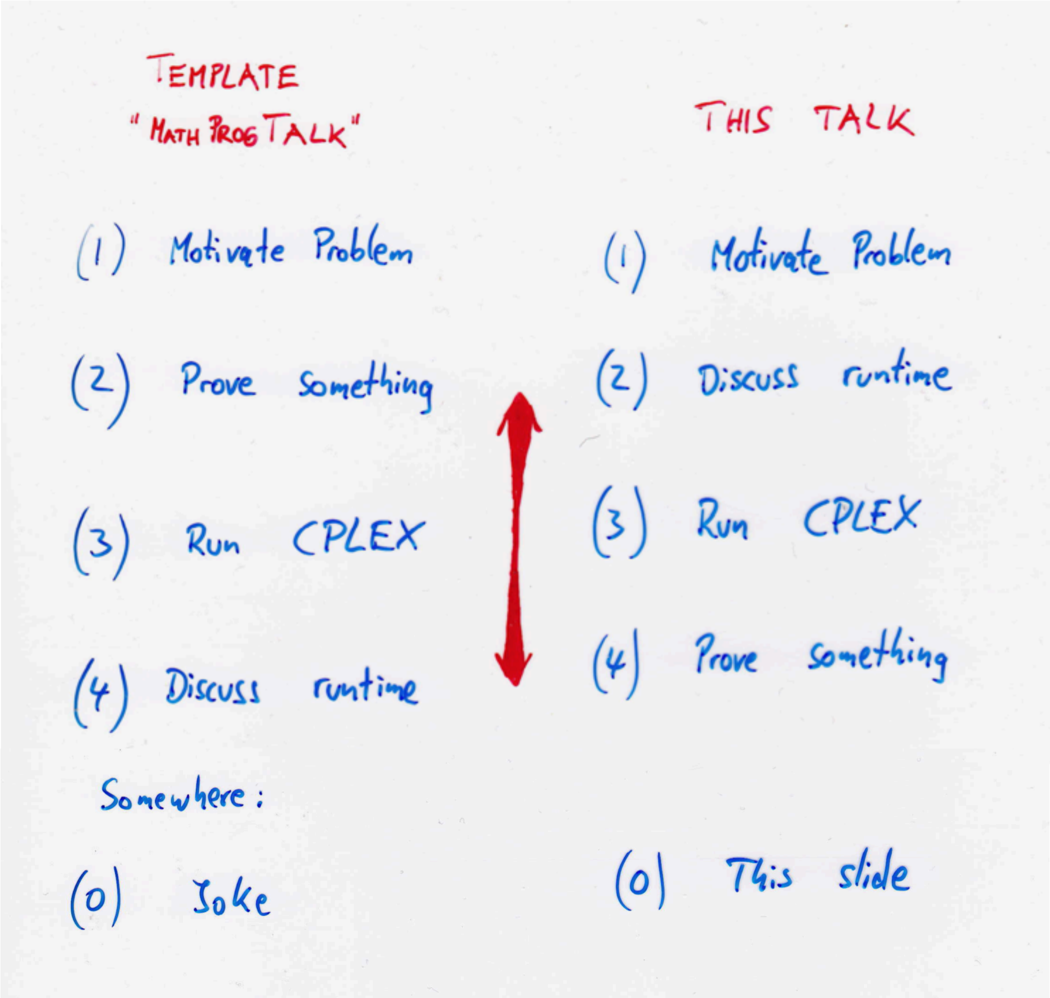
Erik D. Demaine[a], Sándor P. Fekete[b,*], Shmuel Gal[c]

[a] *Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge MA, USA*
[b] *Department of Mathematical Optimization, Braunschweig University of Technology, Braunschweig, Germany*
[c] *Department of Statistics, University of Haifa, Haifa, Israel*

Online Searching



Template: "Math Prog Talk"



Linear Search

GIVEN: A starting position O on a line.

MISSION: Find an object at an unknown location.

UNKNOWN: (1) Direction of the object
(2) Distance OPT of the object

WANTED! A competitive strategy for the searcher that will guarantee that the object is found in time at most c · OPT for some constant "competitive" factor c.

Literature

BELLMAN 1963: Introduced the problem

BECK and NEWMAN 1970: Solved the problem

GAL 1974: Solved a generalization:

Search on m rays
Optimal competitive ratio: $1 + \frac{2m^m}{(m-1)^{m-1}}$
Optimal strategy: Geometric series with ratio $\left(\frac{m}{m-1}\right)$

Literature

KAO Also known as the cow-path problem

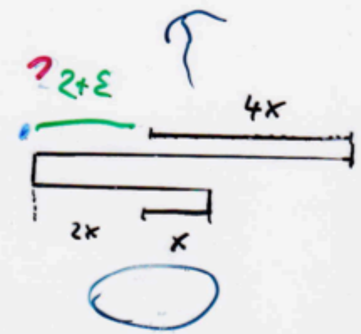
GAL 1980: Optimal trajectory to this type of problem is always a geometric series

BAEZA-YATES, CULBERSON, RAWLINS 1988: (and various others independently) Rediscovered problem and solution

Many variations and applications, in particular for geometric searching.

Doubling

Keep doubling the search distance before returning:



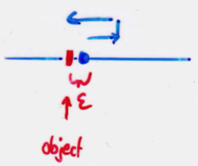
DISADVANTAGE: There is no real "start" of the trajectory – it's just a geometric series, and each previous step was half as long as the latest one!

Turn Cost

Immediate implications:

(1) There has to be a first move.

(2) A competitive factor is no longer possible:



Searching in the wrong direction takes at least one turn, for a cost of d, compared to optimal ε

Fix: Consider $c \cdot OPT + f(d)$

- and possibly $c \cdot OPT + \lambda \cdot d$

An Open Problem

132 BOOK I. SEARCH GAMES

It is worth noting that the worst possible outcome of using the search strategy $s_{\bar{a}}$ ($\bar{a} \doteq 3.6$) is a loss of

$$1 + 2 \sum_{j=-\infty}^{1} \bar{a}^j \doteq 10.9,$$

while the expected cost of the strategy s_2, which uses only minimax trajectories ($a = 2$), is $1 + 3/\ln 2 \doteq 5.3$. Thus, use of $s_{\bar{a}}$ yields (the minimal) expected cost of 4.6 but risks a maximal cost of 10.9, while use of s_2, which yields an expected cost of 5.3, minimizes the maximal cost (which in this case is equal to 9). The expected cost of any search strategy s_a with $2 < a < \bar{a}$ lies between 4.6 and 5.3, while the maximal cost lies between 9 and 10.9. All the strategies s_a with the parameter a lying outside the segment $[2, \bar{a}]$ are dominated by the family $\{s_{\bar{a}} : 2 \le a \le \bar{a}\}$ with respect to the expected and the maximal cost.

8.4 Search with a Turning Cost

In this section we consider a more realistic version of the LSP, which has not been considered before in the literature. In this model the time spent in changing the direction of moving is not 0, as is usually assumed in the LSP, but a constant $d > 0$. Here, any search trajectory with a finite expected search time must have a first step because starting with an infinite number of oscillations takes infinite time. Therefore, assume for convenience that the search trajectory starts by going to $x_0 > 0$, then turning and going to $-x_1$, then turning and going to x_2, etc. (We can obviously assume that the searcher always goes with his maximal speed, 1, as is always the case with an immobile hider.) Thus

$$S = \{x_i\}_{i=0}^{\infty},$$

and denote

$$y_i = x_i + \frac{d}{2}, \quad i = 1, 2, \ldots.$$

In this case the normalized cost function (in the worst case) is not bounded near 0. Thus the reasonable cost function is the time to reach the target, $C(S, H)$, under the restriction $E|H| \le \lambda$. For convenience we assume $\lambda = 1$. Thus we are interested in

$$\hat{V} = \inf_S \sup_{h:E|H| \le 1} c(S, h).$$

We shall show that

$$9 + d \le \hat{V} \le 9 + 2d. \tag{8.13}$$

The left inequality follows from equality (8.7), which implies that for any S and any δ, there always exist an x_i, as large as desired, with

$$\frac{2\sum_{j=0}^{i+1} y_j + x_i}{x_i} \sim \frac{2\sum_{j=0}^{i+1} y_j + y_i}{y_i} > 9 - \delta.$$

CHAPTER 8. SEARCH ON THE INFINITE LINE 133

Thus, if the hider chooses h as

$$H = \begin{cases} -\varepsilon \text{ with probability } 1 - \frac{1}{x_i} & \text{and} \\ x_i + \varepsilon \text{ with probability } \frac{1}{x_i}, \end{cases}$$

then $E|H| \approx 1$ and, for a large enough x_i

$$c(S, h) \approx (2x_0 + d + \varepsilon)\left(1 - \frac{1}{x_i}\right) + \left(2\sum_{j=0}^{i+1} y_j + x_i\right)\frac{1}{x_i} \ge 9 - \delta + d$$

with $\delta > 0$ arbitrarily small.

In order to prove the right inequality of (8.13) we present a trajectory S that satisfies for all $x_I < |H| \le x_{I+2}$:

$$C(S, H) \le 9x_I + 2d \le 9|H| + 2d$$

so that for any h with $E|H| \le 1$

$$c(S, h) \le 9 + 2d.$$

We use the following approach. For any real γ, a sufficient condition for $v(S) \le 9 + \gamma$ is the condition

$$\text{for all } |H| = x_i(+\varepsilon): \quad C(S, H) \le 9x_i + \gamma(+\varepsilon),$$

which will hold if the following conditions hold:

$$2\sum_{0}^{i+1} y_j = 8\left(y_i - \frac{d}{2}\right) + \gamma, \quad i = 0, 1, \ldots \tag{8.14}$$

$$2y_0 = \gamma, \quad (\gamma > d/2)$$

$$y_i \ge d/2, \quad i = 0, 1, \ldots$$

Equality (8.14) is equivalent to (denoting $\frac{\gamma}{2} = b + 2d$)

$$y_{i+1} = 3y_i - \sum_{j=0}^{i-1} y_i + b, \quad i = 0, 1, \ldots \tag{8.15}$$

$$y_0 = b + 2d \left(= \frac{\gamma}{2}\right)$$

$$y_i > d/2, \quad i = 0, 1 \ldots.$$

We now look for the minimal b which satisfies (8.15). It turns out that the general solution of (8.15) is

$$y_i = (y_0 + i\beta)2^i, \tag{8.16}$$

An Open Problem

where $\beta \geq 0$ is a nonnegative parameter. (Because by (8.15) $y_{i+1} - y_i = 3y_i - 4y_{i-1}$, denoting $y_i = 2^i \alpha_i$ it easily follows that $\alpha_{i+1} - \alpha_i = \alpha_i - \alpha_{i-1}$, which leads to (8.16).)

Using (8.16) for $i = 0, 1$ in (8.15) it follows that $\beta = y_0 - d$. Since $\beta \geq 0$ and $\gamma = 2y_0$, it easily follows that $\gamma \geq 2d$. On the other hand, the value $9 + 2d$ can be achieved by the following trajectory

$$y_i = d2^i, \qquad x_i = d2^i - d/2, \quad i = 0, 1, \ldots$$

with the time to reach $x_i + \varepsilon$ being (neglecting $O(\varepsilon)$)

$$2\sum_{0}^{i+1} y_i + x_i = 2d(2^{i+2} - 1) + d2^i - d/2 = 9x_i + 2d.$$

Since $E|H| \leq 1$, the last equation guarantees expected time not exceeding $9 + 2d$.

Is $9 + 2d$ the best possible constant? This is still an open problem. (Note that (8.14) is a sufficient but not a necessary condition.)

Positions

The factor c can be at best 9 !

(→ Consider d arbitrarily small compared to OPT.)

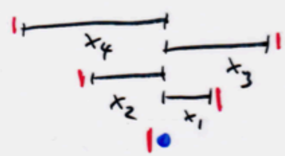
Suppose the searcher moves

x_1 to the right and returns,

x_2 to the left and returns,

x_3 to the right

(etc.)



Critical positions for hiding :

$y_0 = -\varepsilon$

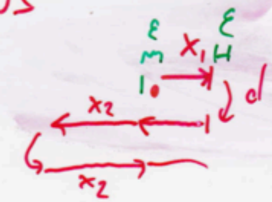
$y_1 = x_1 + \varepsilon$

$y_2 = -x_2 - \varepsilon$

$y_3 = x_3 + \varepsilon$

(etc.)

More Conditions



y_0 must be reached in time:

$$2x_1 + d + \varepsilon \leq 9\varepsilon + \lambda d$$

y_1 must be reached in time:

$$2x_1 + 2x_2 + 2d + x_1 + \varepsilon \leq 9(x_1+\varepsilon) + \lambda d$$

y_2 :

$$2x_1 + 2x_2 + 2x_3 + 3d + x_2 + \varepsilon \leq 9(x_2+\varepsilon) + \lambda d$$

y_n :

$$2x_1 + \ldots + 2x_{n+1} + (n+1)d \leq 8x_n + \lambda d$$

This must hold for all $\varepsilon > 0$, so we get

An Infinite LP

$$\min \lambda$$

$$
\begin{aligned}
2x_1 & & & & + d &\le \lambda d\\
2x_1 &+ 2x_2 & & & + 2d &\le 8x_1 + \lambda d\\
2x_1 &+ 2x_2 &+ 2x_3 & & + 3d &\le 8x_2 + \lambda d\\
\vdots & \quad\vdots & \quad\vdots & \ddots & \vdots\ & \qquad \vdots\\
2x_1 &+ 2x_2 &+ 2x_3 & \cdots + 2x_{n+1} & + (n+1)d &\le 8x_n + \lambda d\\
\vdots & \quad\vdots & \quad\vdots & \quad\vdots\ \ddots & \vdots\ & \qquad \vdots\\
& & & & x_i &\ge 0
\end{aligned}
$$

(1) Infinite primal optimal solution describes optimal strategy of searcher.

(2) Optimal λ is tight value of turn cost penalty.

(3) Infinite dual optimal solution gives explicit proof of tightness.

Solving the Infinite LP

SOLVING SUBSYSTEMS

Only using the first n constraints yields a relaxation, with solutions $x_i^{(n)}$ and λ_n.
Each λ_n is a lower bound for λ.

Approach:

(1) Run CPLEX on subsystems

(2) Consider convergence of solutions

(3) Construct infinite solution

(4) Verify solution

Solutions

n	λ_n	$x_1^{(n)}$	$x_2^{(n)}$	$x_3^{(n)}$	$x_4^{(n)}$	$c_1^{(n)}$	$c_2^{(n)}$	$c_3^{(n)}$	$c_4^{(n)}$
1	1.0000	0.0000				1.0000			

Table 1
Solutions for a number of linear subsystems

n	λ_n	$x_1^{(n)}$	$x_2^{(n)}$	$x_3^{(n)}$	$x_4^{(n)}$	$x_5^{(n)}$
1	1.0000	0.0000				
2	1.2500	0.1250	0.0000			
3	1.4166	0.2083	0.3333	0.0000		
4	1.5312	0.2656	0.5625	0.6875	0.0000	
5	1.6125	0.3062	0.7250	1.1750	1.3000	0.0000
6	1.6718	0.3359	0.8437	1.5312	2.2500	2.3750
7	1.7165	0.3582	0.9330	1.7991	2.9642	4.1607
8	1.7509	0.3754	1.0019	2.0058	3.5156	5.5930
9	1.7782	0.3891	1.0563	2.1692	3.9130	6.6284

n	λ_n	$y_1^{(n)}$	$y_2^{(n)}$	$y_3^{(n)}$	$y_4^{(n)}$	$y_5^{(n)}$
1	1.0000					
2	1.2500	0.7500	0.2500			
3	1.4166	0.6666	0.2500	0.0833		
4	1.5312	0.0625	0.2500	0.0937	0.0312	
5	1.6125	0.6000	0.2500	0.1000	0.0375	0.0125
6	1.6718	0.5833	0.2500	0.1041	0.0416	0.0156
7	1.7165	0.5714	0.2500	0.1071	0.0446	0.0178
8	1.7509	0.5625	0.2500	0.1093	0.0468	0.0195
9	1.7782	0.5555	0.2500	0.1111	0.0486	0.0208
10	1.8001	0.5500	0.2500	0.1125	0.0500	0.0218
20	1.9000	0.5250	0.2500	0.1187	0.0562	0.0265
30	1.9333	0.5166	0.2500	0.1208	0.0583	0.0281
40	1.9500	0.5125	0.2500	0.1218	0.0593	0.0289
50	1.9600	0.5100	0.2500	0.1225	0.0600	0.0293
100	1.9800	0.5050	0.2500	0.1237	0.0612	0.0303
200	1.9900	0.5025	0.2500	0.1243	0.0618	0.0307
400	1.9950	0.5012	0.2500	0.1245	0.0621	0.0310

Verifying the Solution

Choose: $x_i = \left(2^i - \frac{1}{2}\right) d$

$c_j = \frac{1}{2^j}$

Check primal solution, i.e. search strategy:

Inequality n yields

$$\sum_{i=1}^{n+1} 2(x_i) - 8x_n + (n+1)d \leq \lambda d$$

or $$\sum_{i=1}^{n+1} 2\left(2^i - \tfrac{1}{2}\right)d - 8\left(2^{n-1} - \tfrac{1}{2}\right)d + (n+1)d \leq \lambda d$$

or $$2^{n+2} - 2 \quad - 2^{n+2} + 4 \leq \lambda$$

or $$2 \leq \lambda$$

So we have a feasible solution with $\lambda = 2$.

Verifying the Dual

$$\min \lambda$$

$$
\begin{array}{lcl}
2x_1 & +d & \leq \lambda d \\
2x_1 + 2x_2 & +2d & \leq 8x_1 + \lambda d \\
2x_1 + 2x_2 + 2x_3 & +3d & \leq 8x_2 + \lambda d \\
\vdots & \vdots & \vdots \\
2x_1 + 2x_2 + 2x_3 \cdots + 2x_{n+1} & +(n+1)d & \leq 8x_n + \lambda d \\
\vdots & \vdots & \vdots \\
 & x_i & \geq 0
\end{array}
$$

Consider infinite linear combination of with the dual multipliers:

The resulting coefficient of x_n is

$$\sum_{i=n}^{\infty} \frac{2}{2^i} - \frac{8}{2^{n+1}} = 0$$

The resulting coefficient of λd is

$$\sum_{i=1}^{\infty} \frac{1}{2^i} = 1$$

This leaves the inequality

$$\sum_{i=1}^{\infty} i\left(\frac{1}{2}\right)^i d \leq \lambda d$$

Using $\sum_{i=1}^{\infty} i x^i = \frac{x}{(1-x)^2}$, this implies

$$2 \leq \lambda,$$

so we have an explicit lower bound.

More General Problem

COW-PATH PROBLEM WITH TURN COST

SCENARIO: m rays from the origin.

Turn cost on a ray: d_1

Turn cost at the origin: d_2

Total turn cost for changing from one ray to another: $d = d_1 + d_2$

KNOWN: Asymptotic competitive ratio for $d = 0$ is

$$1 + \frac{2m^m}{(m-1)^{m-1}} =: 1 + M$$

Constraints

REWRITE CONSTRAINTS:

$$2 \sum_{j=1}^{n+m-1} x_i + (n+m-1)\, d \leq M x_n + \lambda d$$

AGAIN:

- Infinite LP for determining λ
- Run experiments for fixed m

Solving the Problem

SOLUTION OF THE PROBLEM

Here described: $m = 3$

$$\lambda_{1000} = 3.743996$$
$$x_1^{(1000)} = 0.2492495$$
$$x_2^{(1000)} = 0.6227485$$
$$x_3^{(1000)} = 1.182434$$
$$x_4^{(1000)} = 2.021118$$
$$x_5^{(1000)} = 3.277878$$

After adjusting for logarithmic convergence:

$$\lambda = 3.75 = \frac{15}{4}$$
$$x_1 = 0.25 = \frac{1}{4}$$
$$x_2 = 0.625 = \frac{5}{8}$$

} educated guesses

Assuming all constraints are tight, we get a recursion for x_n, yielding:

$$x_3 = \frac{19}{16} = 1.1875 \quad ✓$$
$$x_4 = \frac{65}{32} = 2.03125 \quad ✓$$
$$x_5 = \frac{211}{64} = 3.296875 \quad ✓$$

Solution II

SOLUTION FOR $m=3$ (Cont.)

Using the structure of the recursion, we conclude

$$x_n = \frac{d}{2}\left(\left(\frac{3}{2}\right)^n - 1\right)$$

Not hard to check:

Together with $\lambda = \frac{15}{4}$, this satisfies all constraints with equality.

Dual Variables

$C_2^{(1000)} = 0.445339$

$C_3^{(1000)} = 0.1481481$

$C_4^{(1000)} = 0.1481481$

$C_5^{(1000)} = 0.08217275$

$C_6^{(1000)} = 0.06022488$

$C_7^{(1000)} = 0.038277$

$C_8^{(1000)} = 0.02610326$

Using (*), we get the recursive condition

$$C_n = \frac{27}{4}\left(C_{n+2} - C_{n+3}\right)$$

or

$$C_{n+3} = \frac{27}{4} C_{n+2} - C_n .$$

Some values:

$C_5 = \frac{60}{3^6} = 0.0823045$ ✓

$C_6 = \frac{132}{3^7} = 0.0603566$ ✓

$C_7 = \frac{252}{3^8} = 0.0384087$ ✓

$C_8 = \frac{516}{3^9} = 0.0262155$ ✓

Technische Universität Braunschweig

Dual Routing

Explicit formula after solving recursion:

$$c_j = \frac{2^{j+1} + (-1)^j 4}{3^{j+1}}$$

Dual Routing

VERIFYING THE DUAL

Consider the infinite linear combination of all constraints, using the computed c_j.

- By assumption, we have

$$\sum_{i=2}^{\infty} c_i = 1$$

so the coefficient of $2d$ is 1.

- By recursion, all coefficients of x_n cancel.

Dual Routing

$$
\begin{aligned}
\sum_{j=m-1}^{\infty} j y_j &= \sum_{j=m-1}^{2m-2} j y_j + \sum_{j=2m-1}^{\infty} j y_j \\
&= \sum_{j=m-1}^{2m-2} j y_j + \sum_{j=m-1}^{\infty} (j+m) y_{j+m} \\
&= \sum_{j=m-1}^{2m-2} j y_j + \sum_{j=m-1}^{\infty} (j+m) \left(y_{j+m-1} - \frac{1}{M} y_j \right) \\
&= (2m-2) y_{2m-2} + \sum_{j=m-1}^{2m-3} j y_j + \sum_{j=m-1}^{\infty} (j+m-1) y_{j+m-1} \\
&\quad + \sum_{j=m-1}^{\infty} y_{j+m-1} - \sum_{j=m-1}^{\infty} \frac{1}{M} j y_j - \sum_{j=m-1}^{\infty} \frac{m}{M} y_j \\
&= \frac{2m-2}{M} + \sum_{j=m-1}^{\infty} j y_j + \left(1 - \sum_{j=m-1}^{2m-3} y_j \right) - \sum_{j=m-1}^{\infty} \frac{1}{M} j y_j - \frac{m}{M},
\end{aligned}
$$

hence

$$
\sum_{j=m-1}^{\infty} j y_j = 2m - 2 + (M - m - (m-2)) - m = M - m,
$$

as claimed. □

ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

Theoretical Computer Science 361 (2006) 342–355

Theoretical Computer Science

www.elsevier.com/locate/tcs

Online searching with turn cost

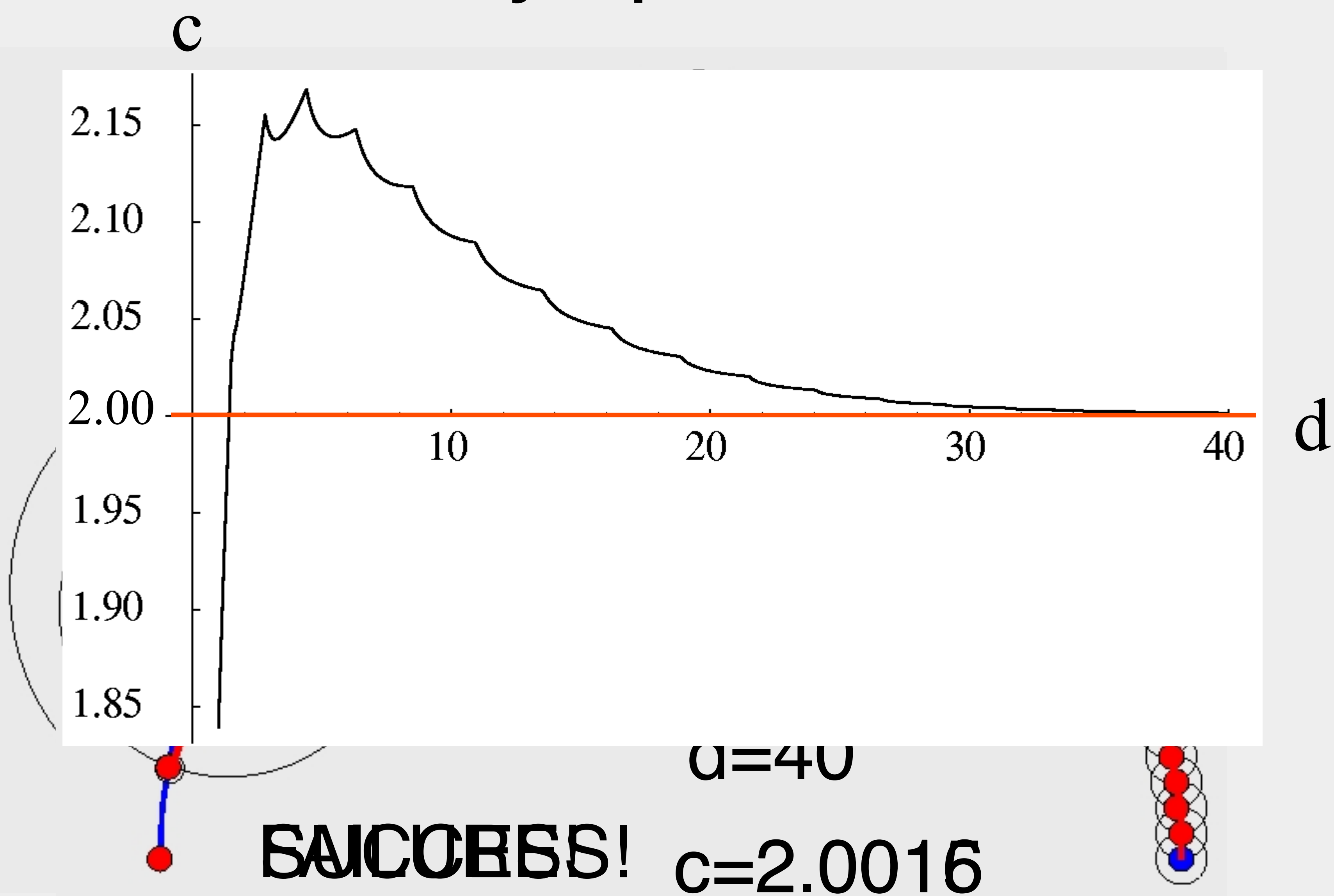
Erik D. Demaine[a], Sándor P. Fekete[b,*], Shmuel Gal[c]

[a] *Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge MA, USA*
[b] *Department of Mathematical Optimization, Braunschweig University of Technology, Braunschweig, Germany*
[c] *Department of Statistics, University of Haifa, Haifa, Israel*

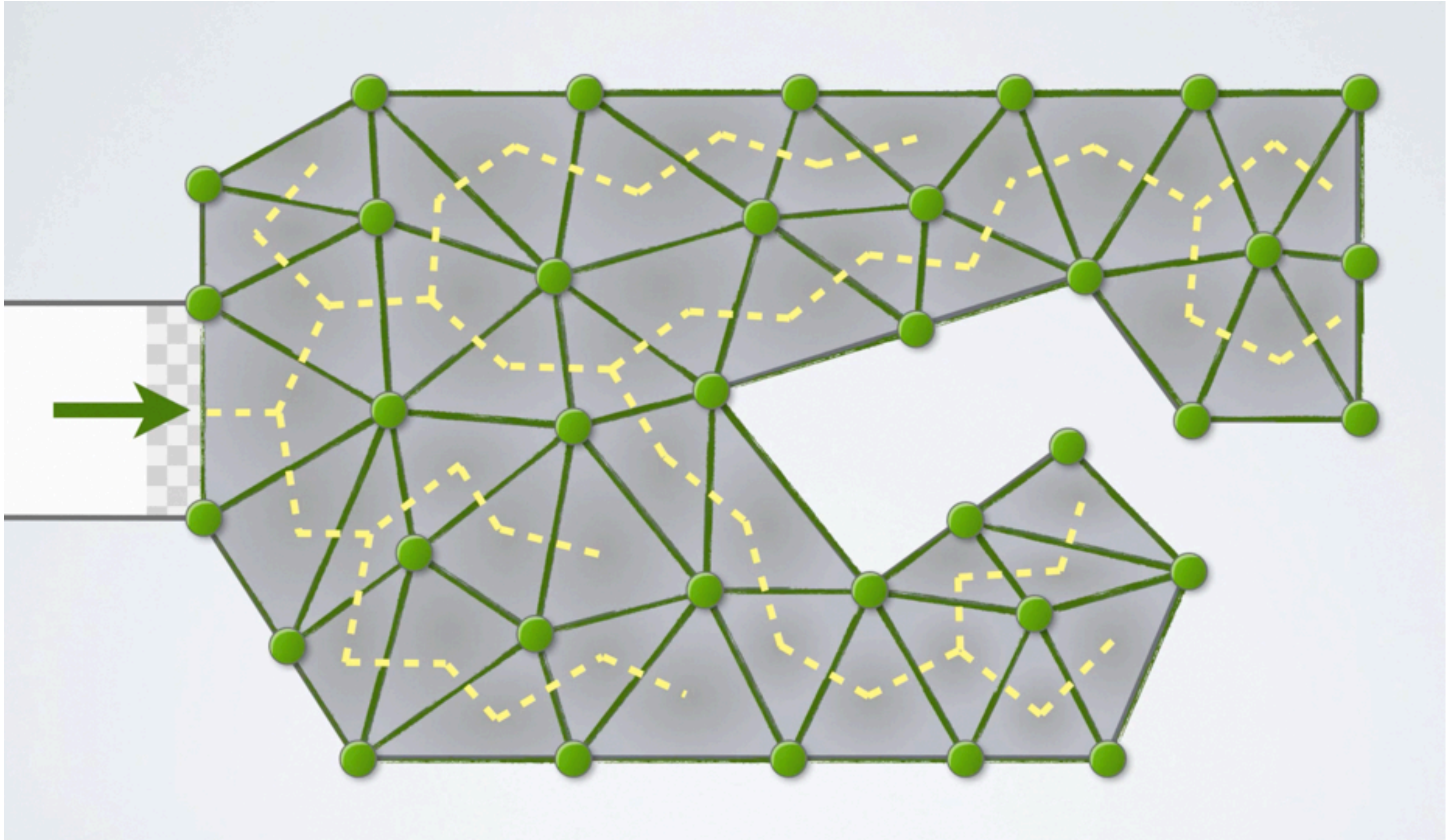
Part 2: Several Robots

Part 2.1: Online Tree Exploration

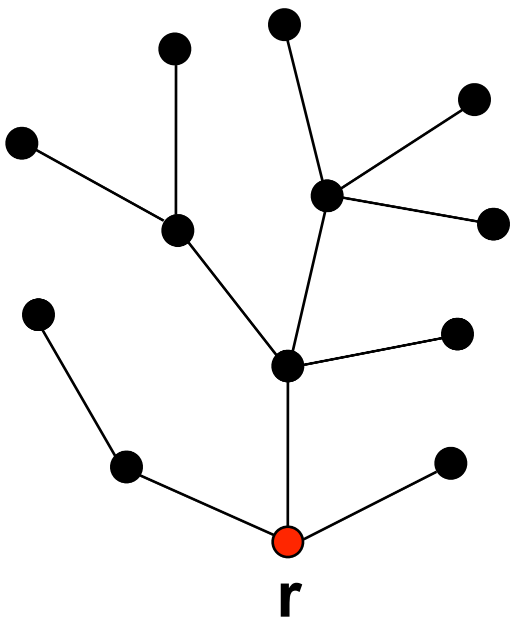
Asymptotics

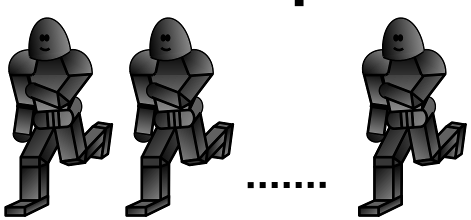


Collective Tree Exploration



Tree Exploration



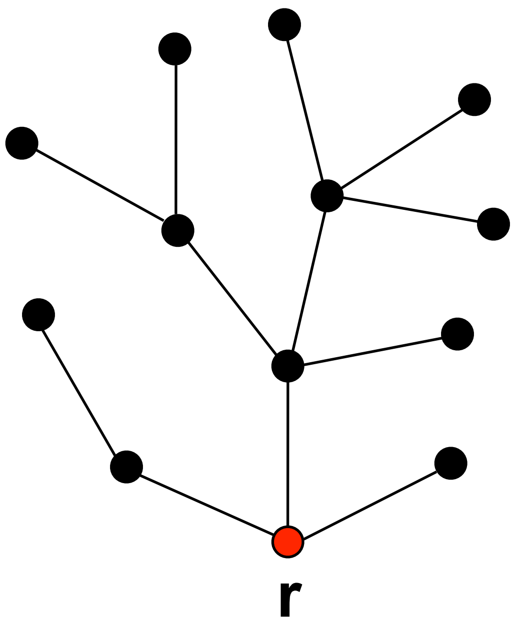


Given:
Unknown tree T, root r
k robots, initially located at r

Task:
Explore T and return to origin

Objective:
Minimize maximum workload

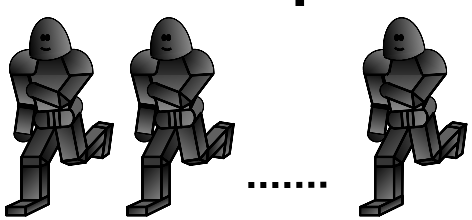
Tree Exploration



Given:
Unknown tree T, root r
k robots, initially located at r

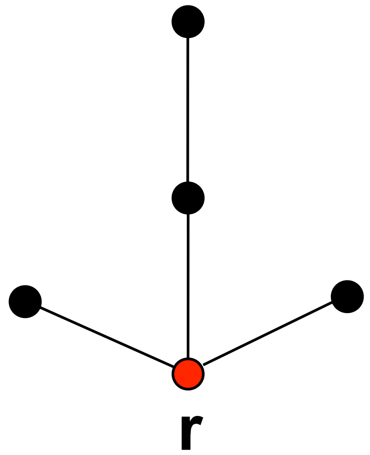
Task:
Explore T and return to origin

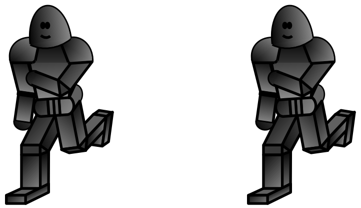
Objective:
Minimize maximum workload



Previous Work

$$\frac{ALG}{OPT} = \frac{6}{4} = \frac{3}{2}$$





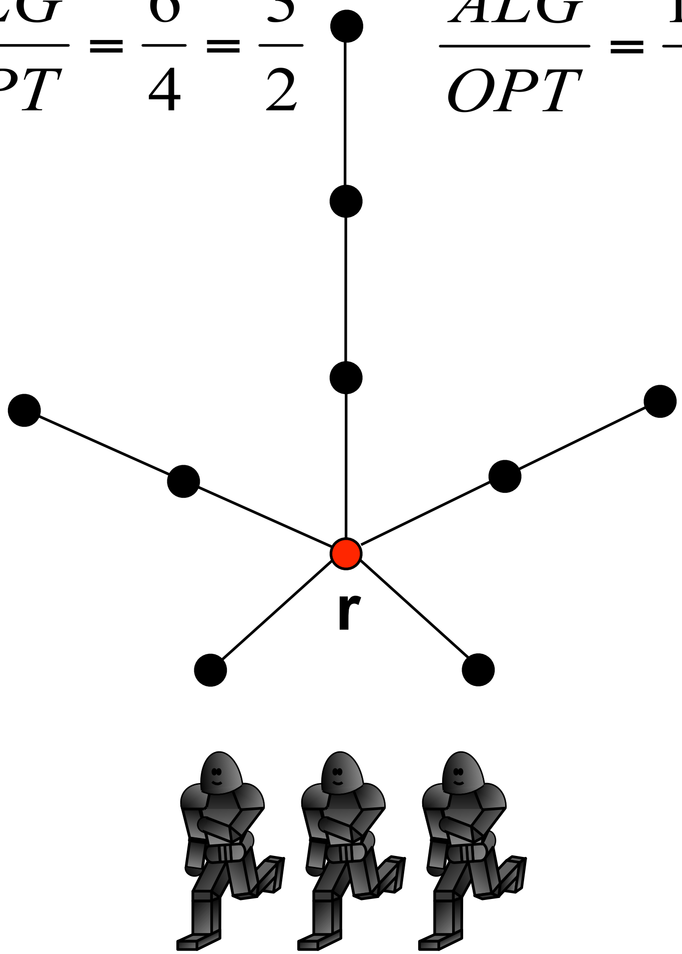
Dynia et al. (2006):

- Lower bound of 3/2 on competitive factor
- An appropriate greedy algorithm achieves competitive factor of 8

Better Lower Bounds

$$\frac{ALG}{OPT} = \frac{6}{4} = \frac{3}{2}$$

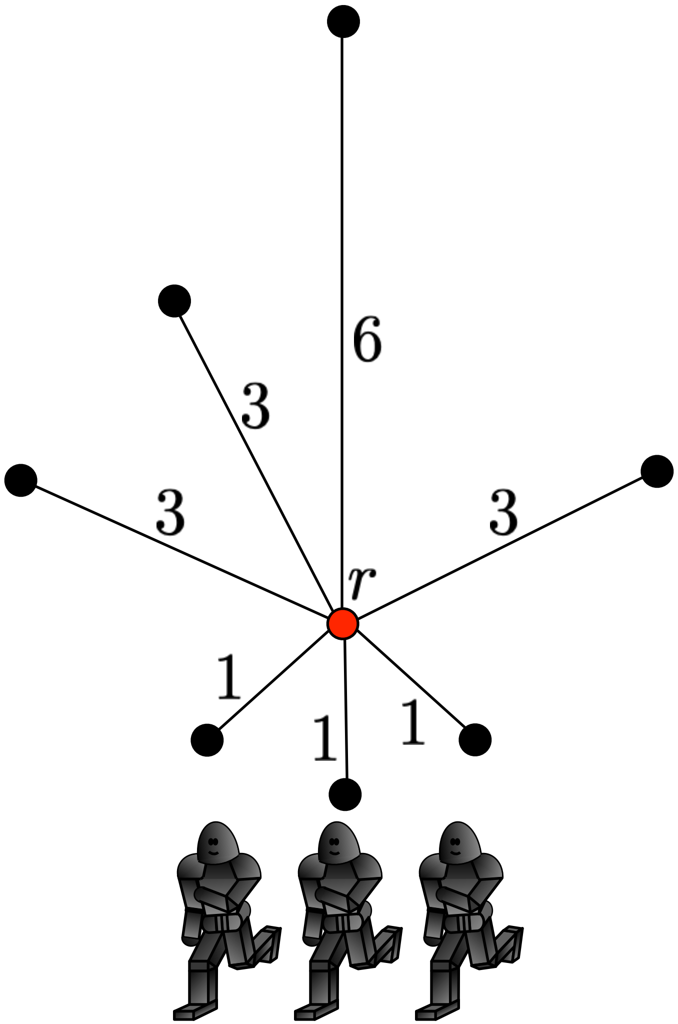
$$\frac{ALG}{OPT} = \frac{10}{6} = \frac{5}{3}$$



- Lower bound of 5/3 on competitive factor for k=3
- More sophisticated examples yield lower bound of

$$1 + \frac{\sqrt{2}}{2} = 1.707...$$

Better Lower Bounds

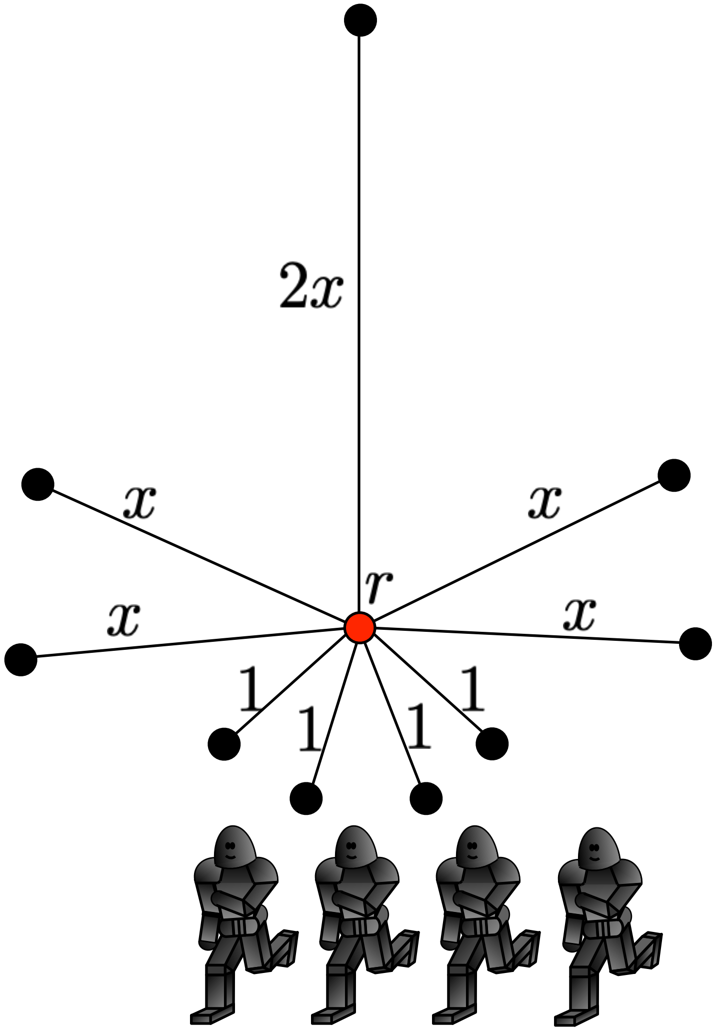


- Lower bound of 5/3 on competitive factor for k=3

- More sophisticated examples yield lower bound of

$$1 + \frac{\sqrt{2}}{2} = 1.707...$$

Better Lower Bounds

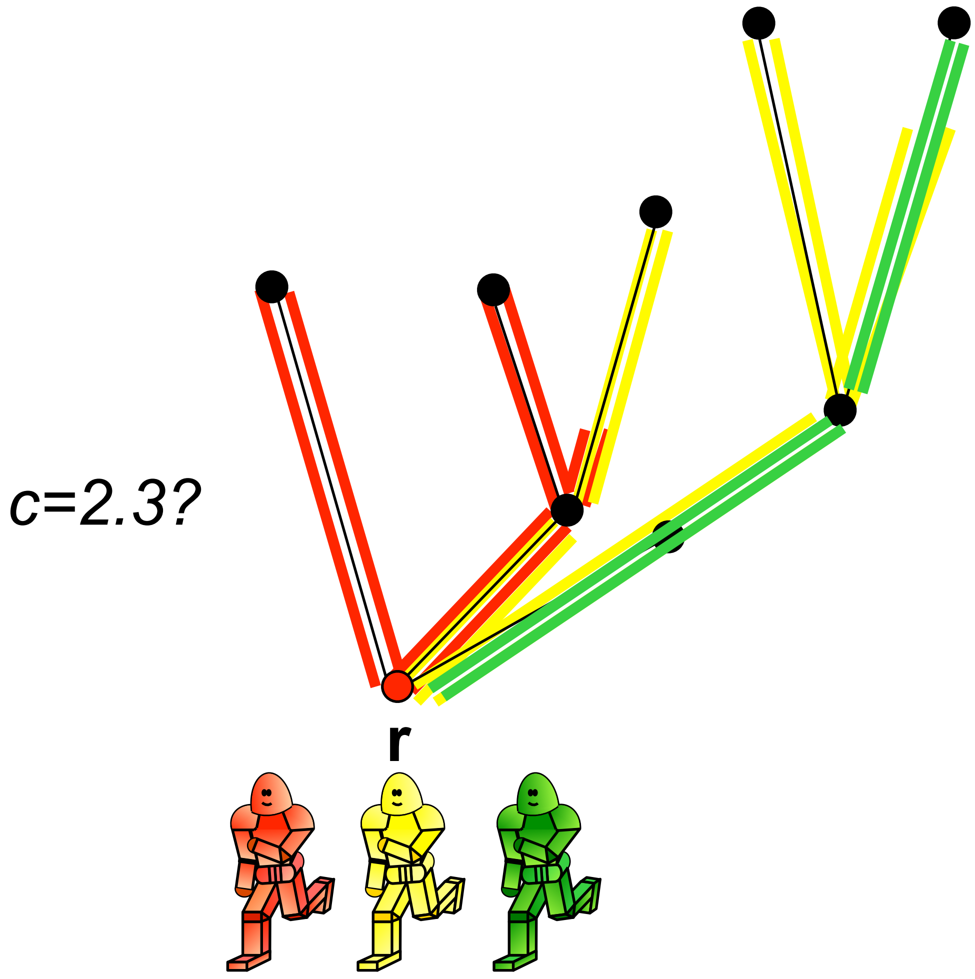


- Lower bound of 5/3 on competitive factor for k=3

- More sophisticated examples yield lower bound of

$$1 + \frac{\sqrt{2}}{2} = 1.707...$$

A New Strategy for General Trees



- Lower bounds on actual OPT:
  - Known MAX distance
  - AVG of known total distance
- Strategy MAX+AVG:
  - Choose some *c*.
  - Robots take turns, one at a time.
  - Keep track of MAX and AVG.
  - Travel c times lower bound.
- Factor *c* is achievable, if we can keep going - so if we can travel arbitrarily far.
- Observations:
  - Duplicated distance DUP is bounded by MAX.
  - In worst case, MAX=AVG=DUP.
  - This yields a recursion for distances traveled.

A New Strategy for General Trees

Online Balanced Tree Exploration

Pravesh Agrawal ✉
Department of CSE, IIT Bombay, Mumbai, India

Sándor P. Fekete ✉ iD
Department of Computer Science, TU Braunschweig, Braunschweig, Germany

Abstract

We study *Online Balanced Tree Exploration*, a class of online optimization problems that can be seen as natural generalizations of both online exploration and machine scheduling: Given an unknown weighted tree $T = (V, E)$ with a distinguished root node r, and a set of $k \geq 2$ identical robots at r, the task is to have all vertices of the tree be visited by some robot and have all robots return to r, such that the largest distance traveled by any robot is minimized. Online Balanced Tree Exploration has been considered before; the best previously known competitive method uses a doubling strategy and yields a factor of 8.

We develop c-GAME, a strategy that proceeds greedily while keeping track of tree depth and average load, and show that it yields a c-competitive strategy for any k and any $c \geq \gamma = 3.146193220582\ldots$, which is tight. Here $\gamma = -W_{-1}(-\frac{1}{e^2})$, where W_{-1} is the lower branch of Lambert's W-function, which is also known as the product logarithm. We also provide a tight characterization of the critical competitive factors γ_k for any specific $k \geq 3$; in particular, we establish $\gamma_3 = 2.27883\ldots$, $\gamma_4 = 2.49221\ldots$, $\gamma_{18} = 2.99961\ldots$, implying that 3-GAME is 3-competitive for all $k \leq 18$.

2012 ACM Subject Classification Theory of computation → Online algorithms; Computing methodologies → Planning and scheduling

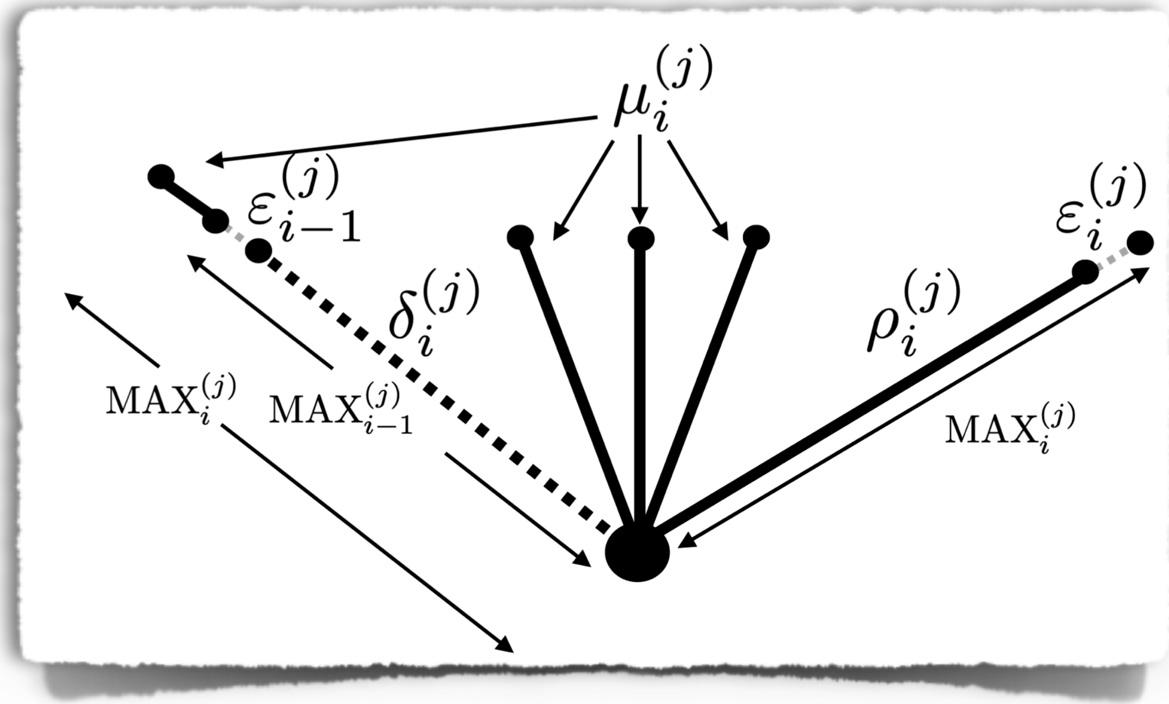
Keywords and phrases Online search, group exploration, balanced allocation, competitive analysis

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2022.118

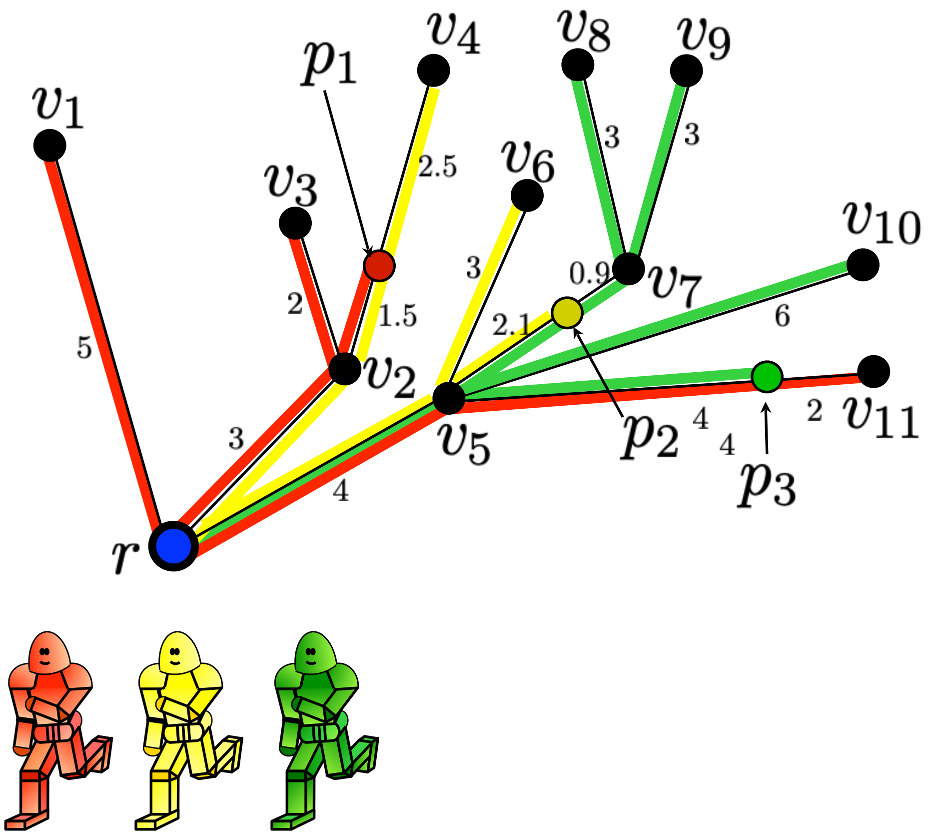
A Useful Lemma

▶ **Lemma 3.** *For analyzing the worst case for strategy c-GAME with $k > c > 2$, it suffices to consider*

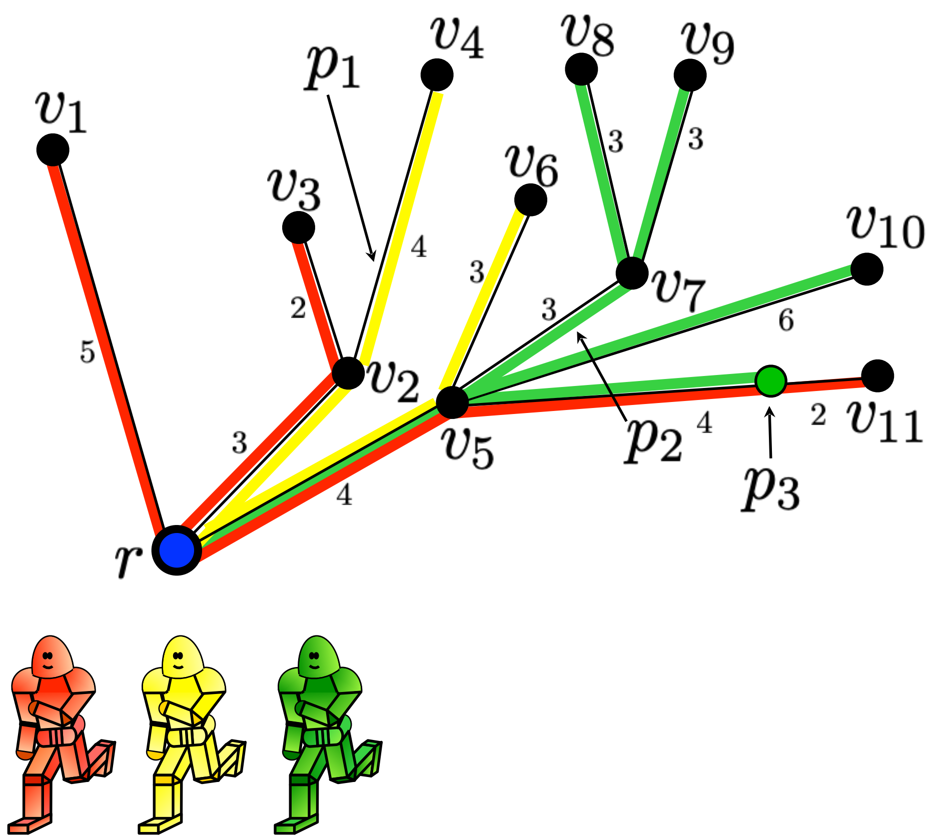
1. $\delta_i^{(j)} = MAX_{i-1}^{(j)} - \varepsilon_i^{(j)}$ *with* $\varepsilon_i^{(j)} > 0$ *arbitrarily small for all* i, j *after* $i = 1, j = 1$.
2. $MAX_i^{(j)} = AVG_i^{(j)}$ *for all* i *and all* $j \geq 2$.



A New Strategy for General Trees



A New Strategy for General Trees



Recursion

D_i : total distance traveled by a robot after iteration i

d_i : new distance traveled by a robot in iteration i

New

$$\underbrace{d_i}_{\text{new}} + \underbrace{D_{i-k}}_{\text{old total}} + \underbrace{\frac{D_{i-1}}{c}}_{\text{duplicated}} = c\left(\underbrace{\frac{D_{i-1}}{c}}_{\text{old average}} + \underbrace{\frac{d_i}{k}}_{\text{added to average}}\right)$$

Rearrange

$$D_i = \left(\frac{k-1}{k-c}\right) D_{i-1} - \left(\frac{c}{k-c}\right) D_{i-k}$$

Analysis

▶ **Lemma 1.** *For any fixed k, Strategy MAX+AVG is c_k-competitive, where c_k satisfies* $c_k = \frac{kx_k^k-(k-1)x_k^{k-1}}{x_k^k-1}$, *with $x_k > 1$ being a zero of the function $f(x) = (k-1)x^k - k^2x + k^2 - 2k + 1$.*

$$x_k^k - \frac{k-1}{(k-c_k)}x_k^{k-1} + \frac{c_k}{k-c_k} = 0$$

$$x_k^{k-1} = \frac{c_k}{c_k-1}$$

$$x_k > 1$$

$$(1 + \frac{1}{c_k-1})^{\frac{1}{k-1}} = 1.$$

Analysis

▶ **Lemma 1.** *For any fixed k, Strategy MAX+AVG is c_k-competitive, where c_k satisfies* $c_k = \frac{kx_k^k-(k-1)x_k^{k-1}}{x_k^k-1}$, *with $x_k > 1$ being a zero of the function* $f(x) = (k-1)x^k - k^2x + k^2 - 2k + 1$.

$$x_k^k - \frac{k-1}{(k-c_k)}x_k^{k-1} + \frac{c_k}{k-c_k} = 0$$

$$(k-c_k)x_k^k - (k-1)x_k^{k-1} + c_k = 0$$

$$c_k = \frac{kx_k^k - (k-1)x_k^{k-1}}{x_k^k - 1}.$$

Derivative

$$\frac{x_k^{k-2}\left((k-1)x_k^k - k^2x_k + k^2 - 2k + 1\right)}{\left(x_k^k - 1\right)^2}$$

Analysis

▶ **Lemma 1.** *For any fixed k, Strategy MAX+AVG is c_k-competitive, where c_k satisfies $c_k = \frac{kx_k^k-(k-1)x_k^{k-1}}{x_k^k-1}$, with $x_k > 1$ being a zero of the function $f(x) = (k-1)x^k - k^2x + k^2 - 2k + 1$.*

k	c_k
2	
3	
4	
5	
6	
7	
8	
9	
10	
20	
40	
100	
1,000	
10,000	
100,000	
1,000,000	

▶ **Theorem 2.** *Strategy MAX+AVG is c_k-competitive, for the values shown in Table 1. Moreover, these values are tight.*

Analysis

$$x_k = \left(1 + \frac{z_k}{k}\right)$$

$$c_k = \frac{k x_k^k - (k-1) x_k^{k-1}}{x_k^k - 1}$$

$$c_k = \frac{k\left(1 + \frac{z_k}{k}\right) - (k-1)}{\left(1 + \frac{z_k}{k}\right) - \frac{1}{\left(1+\frac{z_k}{k}\right)^{k-1}}} = \frac{1 + z_k}{\left(1 + \frac{z_k}{k}\right) - \frac{1}{\left(1+\frac{z_k}{k}\right)^{k-1}}}$$

Analysis

$$c_k = \frac{k(1+\frac{z_k}{k}) - (k-1)}{(1+\frac{z_k}{k}) - \frac{1}{(1+\frac{z_k}{k})^{k-1}}} = \frac{1+z_k}{(1+\frac{z_k}{k}) - \frac{1}{(1+\frac{z_k}{k})^{k-1}}}$$

$$\lim_{k\to\inf} \frac{1+z_k}{(1+\frac{z_k}{k}) - \frac{1}{(1+\frac{z_k}{k})^{k-1}}} = \frac{1+z}{1-e^{-z}}$$

Derivative

$$\frac{e^z(-z+e^z-2)}{(e^z-1)^2}$$

Zero of

$$e^z = z + 2$$

$$c = W_{-1}(-\frac{1}{e^2}) = 3.146193220582\ldots$$

Analysis

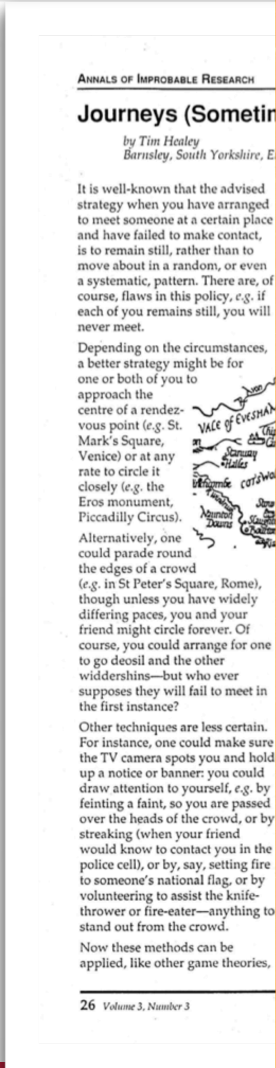
▶ **Theorem 3.** *Algorithm MAX+AVG is c-competitive for all k, where c is the solution of the equation $e^c = c + 2$. This is the value $W_{-1}(-\frac{1}{e^2}) = 3.146193220582\ldots$, where W_{-1} is the lower branch of Lambert's W-function. Moreover, this is tight: For any $c' < c$, MAX+AVG is not c'-competitive for large enough k.*

$$\frac{e^z(-z+e^z-2)}{(e^z-1)^2}$$

$$e^z = z + 2$$

$$c = W_{-1}(-\tfrac{1}{e^2}) = 3.146193220582\ldots$$

Part 2.2:
Rendezvous Search


ANNALS OF IMPROBABLE RESEARCH

Journeys (Sometin

by Tim Healey
Barnsley, South Yorkshire, E

It is well-known that the advised strategy when you have arranged to meet someone at a certain place and have failed to make contact, is to remain still, rather than to move about in a random, or even a systematic, pattern. There are, of course, flaws in this policy, *e.g.* if each of you remains still, you will never meet.

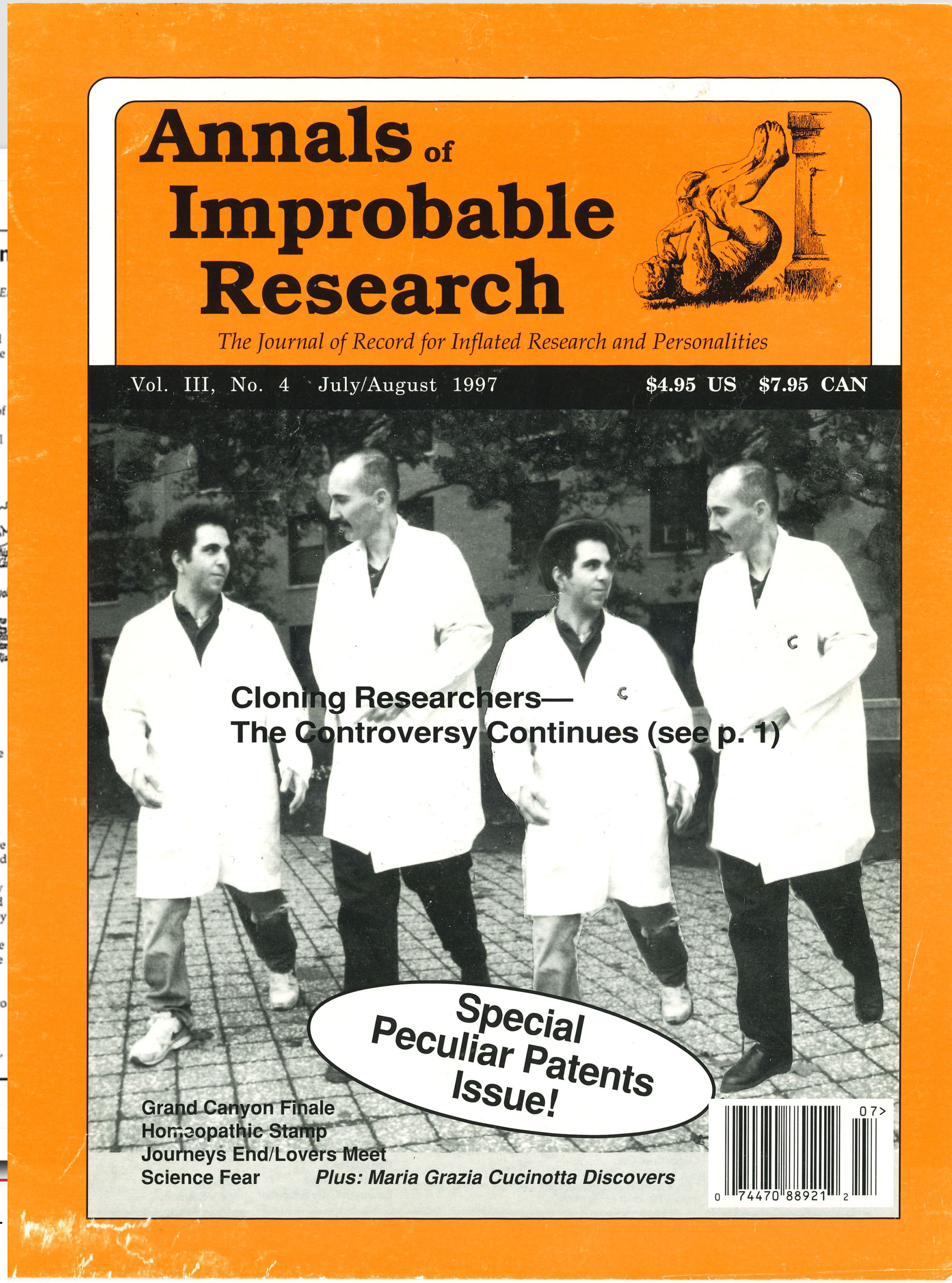
Depending on the circumstances, a better strategy might be for one or both of you to approach the centre of a rendez-vous point (*e.g.* St. Mark's Square, Venice) or at any rate to circle it closely (*e.g.* the Eros monument, Piccadilly Circus).

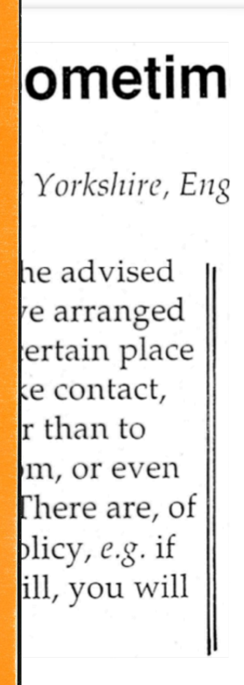
Alternatively, one could parade round the edges of a crowd (*e.g.* in St Peter's Square, Rome), though unless you have widely differing paces, you and your friend might circle forever. Of course, you could arrange for one to go deosil and the other widdershins—but who ever supposes they will fail to meet in the first instance?

Other techniques are less certain. For instance, one could make sure the TV camera spots you and hold up a notice or banner: you could draw attention to yourself, *e.g.* by feinting a faint, so you are passed over the heads of the crowd, or by streaking (when your friend would know to contact you in the police cell), or by, say, setting fire to someone's national flag, or by volunteering to assist the knife-thrower or fire-eater—anything to stand out from the crowd.

Now these methods can be applied, like other game theories,

26 *Volume 3, Number 3*





Publication

OPERATIONS RESEARCH

JOURNAL HOME ARTICLES IN ADVANCE CURRENT ISSUE ARCHIVES ABOUT SUBMIT SUBSCRIBE

Home > Operations Research > Vol. 49, No. 1 >

View PDF

Tools | Share

Two Dimensional Rendezvous Search

Edward J. Anderson, Sándor P. Fekete

Published Online: 1 Feb 2001 | https://doi.org/10.1287/opre.49.1.107.11191

Go to Section

Abstract

Abstract

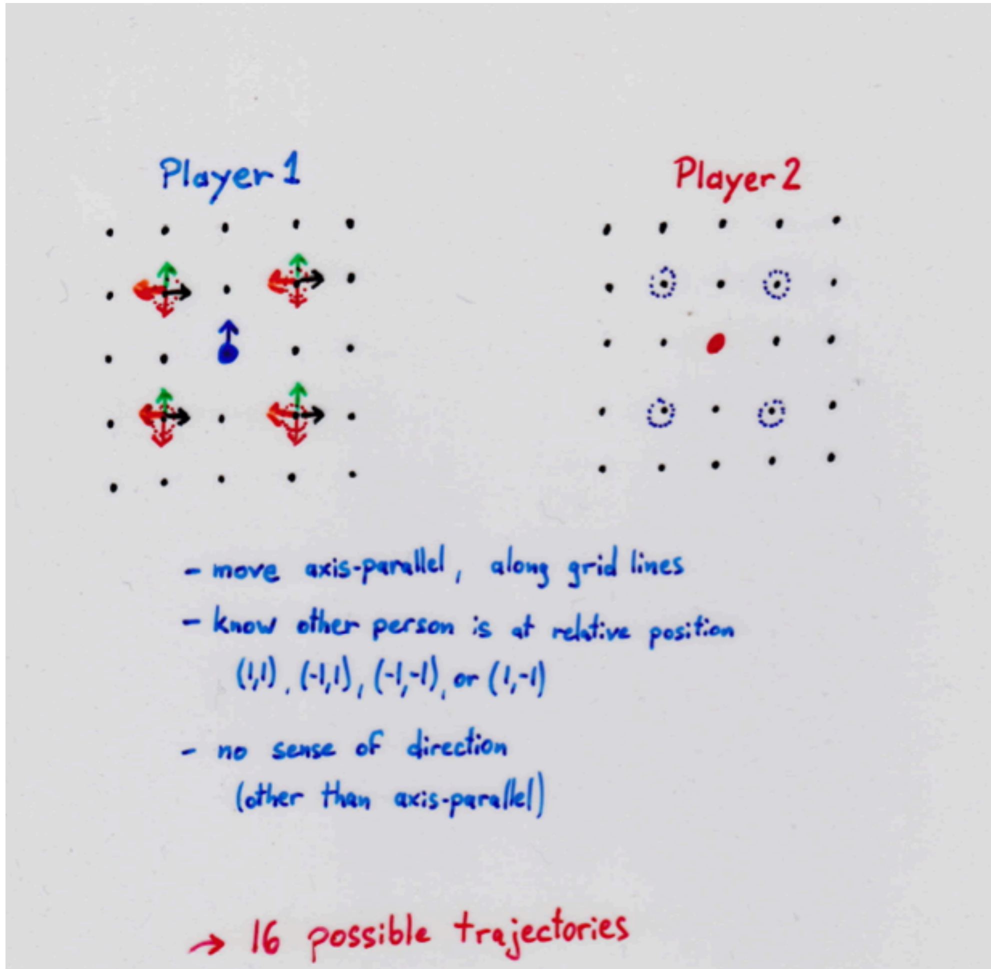
We consider rendezvous problems in which two players move on the plane and wish to cooperate to minimise their first meeting time. We begin by considering the case where both players are placed such that the vector difference is chosen equiprobably from a finite set. We also consider a situation in which they know they are a distance *d* apart, but they do not know the direction of the other player. Finally, we give some results for the case in which player 1 knows the initial position of player 2, while player 2 is given information only on the initial distance of player 1.

Rendezvous Search

One Dimension

A Grid Scenario

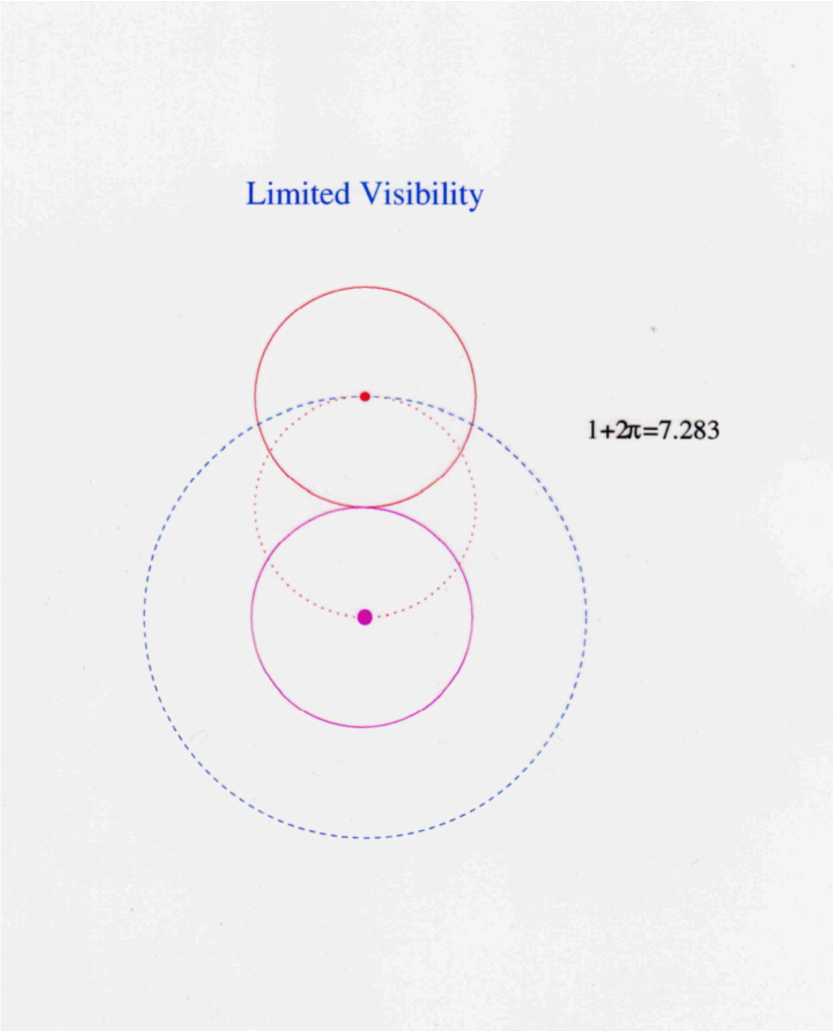
Grid Scenario



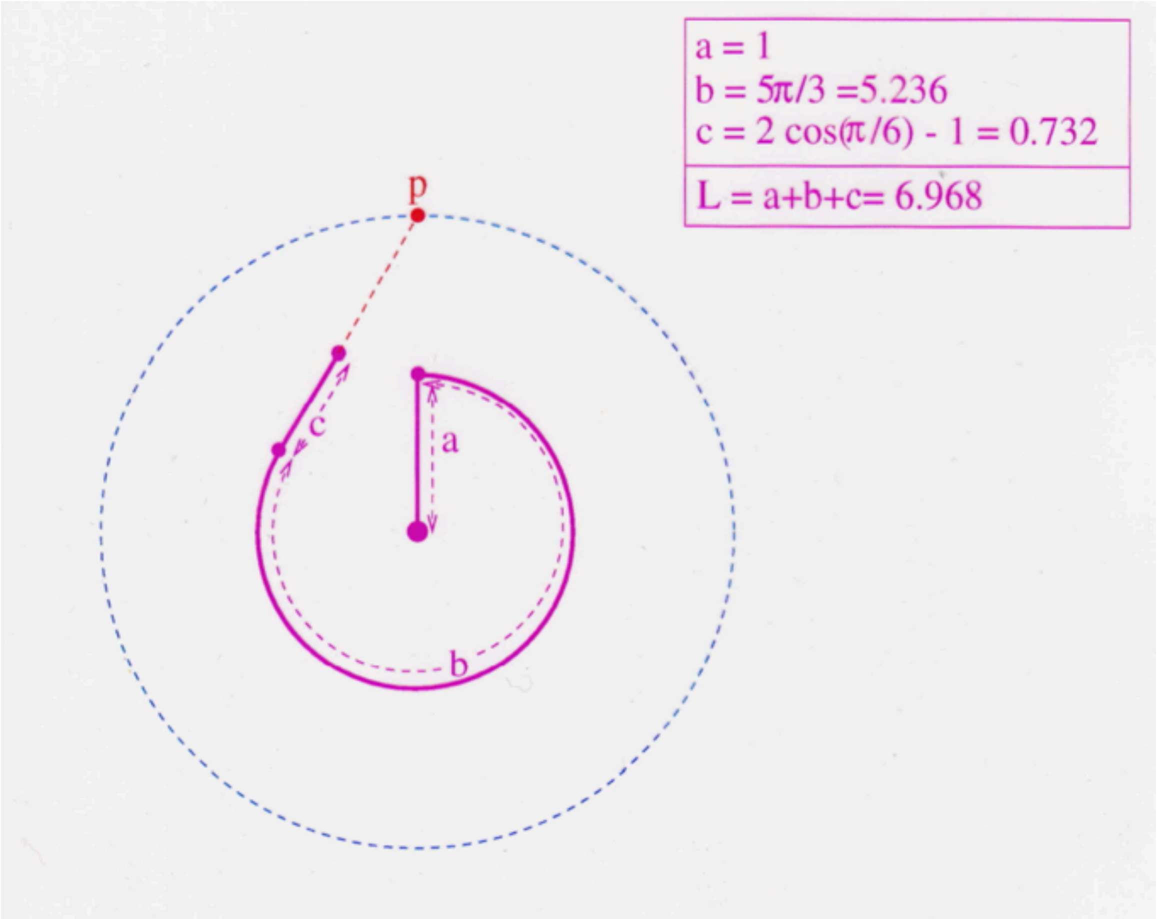
Meeting Vectors

A Proof of Optimality

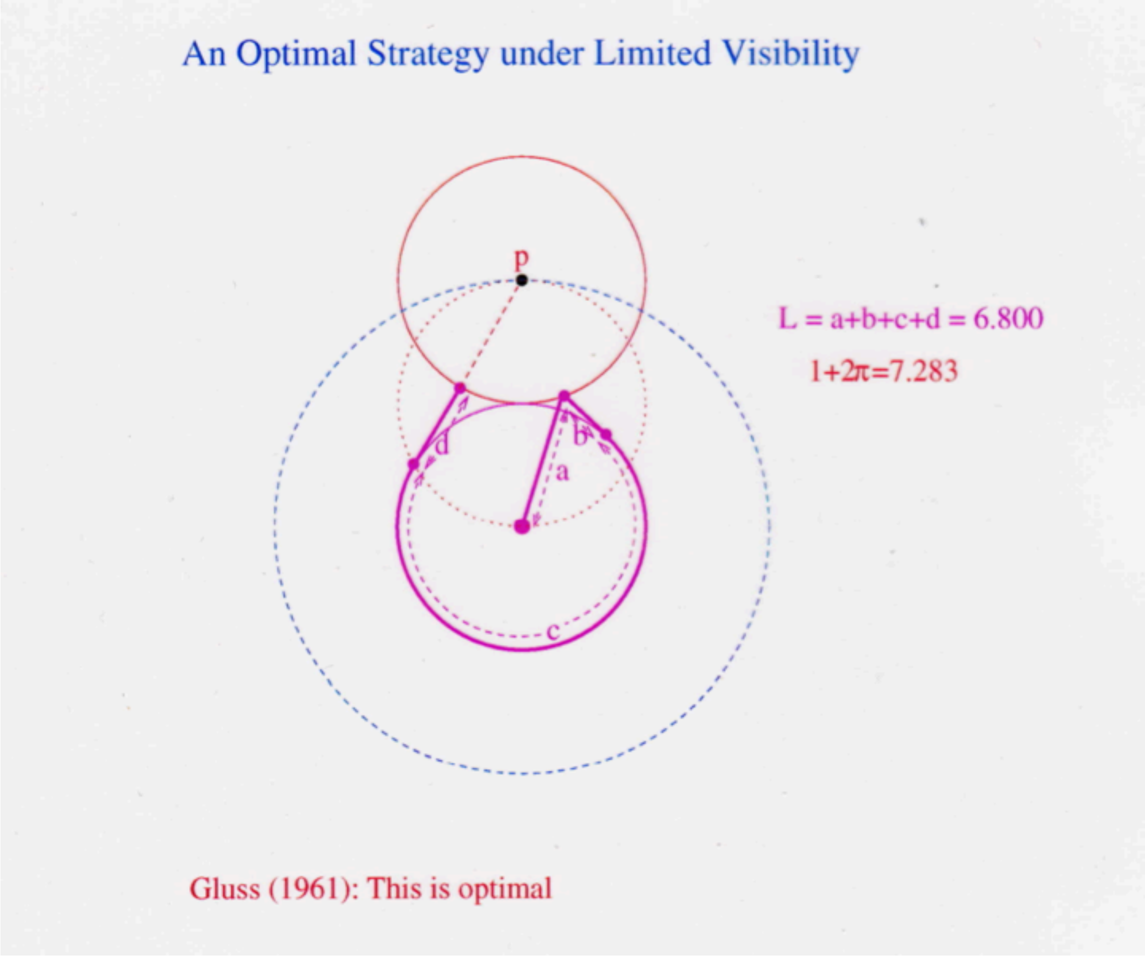
Searching



A Better Solution



An Optimal Solution



2-Dimensional Rendezvous

A Feasible Solution

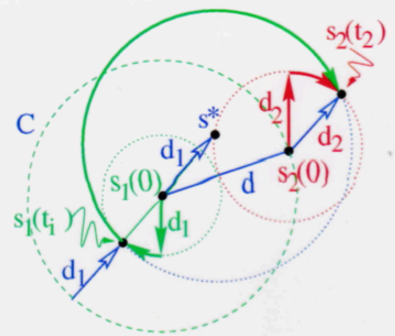
A Different Strategy

A Proof of Optimality

More General Scenario

Limited Fuel

If $s_1+s_2<d$, the chance of meeting is 0, so consider $s_1+s_2 \geq d$.



Let $d_1 = d \dfrac{v_1}{v_1+v_2}$ and $d_2 = d \dfrac{v_2}{v_1+v_2}$, and

assume $s_1/v_1 \geq s_2/v_2$

Player 1 moves at speed v_1 ; first a distance of d_1 south, then following a clockwise circle of radius d_1 .

Player 2 moves at speed v_2 ; first a distance of d_2 north, then following a clockwise circle of radius d_2 .

When player 2 rounds out of fuel, player 1 changes to a circle of radius d around an appropriate center.

Again, we can argue that this is optimal.
This results in a probability of meeting of

$$\min \left\{ \frac{s_1+s_2-d}{2\pi d}, 1 \right\}$$

Furthermore, we get optimal time, if there is a meeting.

Two Dims in Distress

LOST AT SEA

Solving a Special Case

Consider $v_1 = v_2 = 1$, $d = 2$

To express trajectory in terms of angle, write $y(\alpha)$ with $R(t) = y(\Theta(t))$.

Then minimize

$$\int_0^{\pi} \frac{t(\alpha)}{\pi}\, d\alpha,$$

thus

$$\int_0^{\pi} \left(1-\frac{\alpha}{\pi}\right) \frac{dt}{d\alpha}\, d\alpha \qquad \text{(integrate by parts)}$$

Now

$$\left(\frac{dt}{d\alpha}\right)^2 = y(\alpha)^2 + \left(\frac{dy}{d\alpha}\right)^2,$$

so

$$\min \int_0^{\pi} \left(1-\frac{\alpha}{\pi}\right) \left(y(\alpha)^2 + \left(\frac{dy}{d\alpha}\right)^2\right)^{1/2} d\alpha.$$

Using some calculus of variations, we get the differential equation

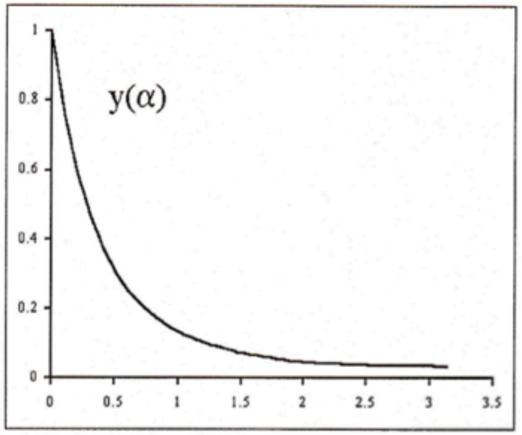
$$\left(\frac{d}{d\alpha}\right)\left[\left(1-\frac{\alpha}{\pi}\right)\left(y(\alpha)^2 + \left(\frac{dy}{d\alpha}\right)^2\right)^{-1/2} \frac{dy}{d\alpha}\right]$$

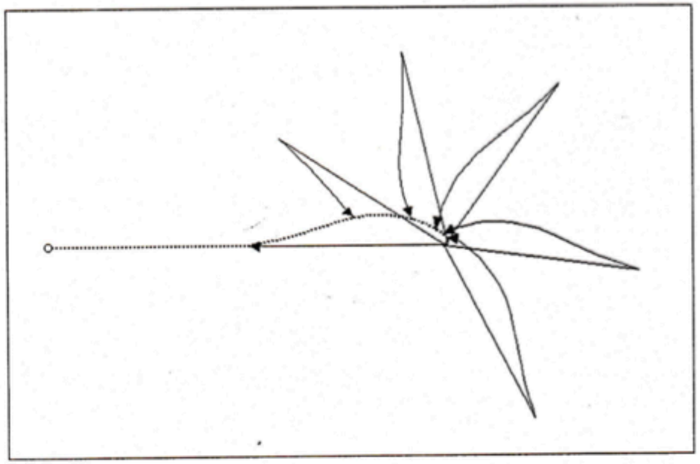
$$= \left(1-\frac{\alpha}{\pi}\right)\left(y(\alpha)^2 + \left(\frac{dy}{d\alpha}\right)^2\right)^{-1/2} y.$$

Simplified:

$$(\pi-\alpha)\left(y^3 + 2y\left(\frac{dy}{d\alpha}\right)^2 - y^2\frac{d^2y}{d\alpha^2}\right) + \left(\frac{dy}{d\alpha}\right)^3 + \frac{dy}{d\alpha}y^2 = 0$$

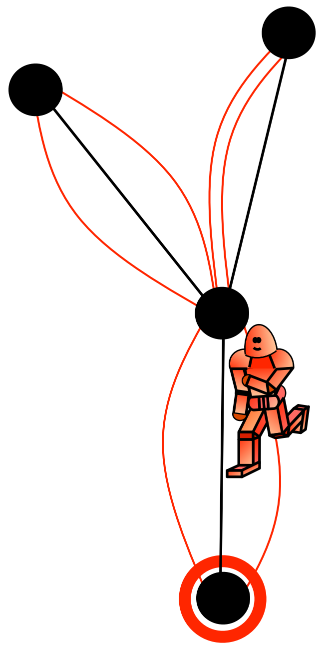
Trajectories





Part 3: Robot Swarms

Tree Exploration with k=1: (1) Local Rules



Depth-First Search (DFS)

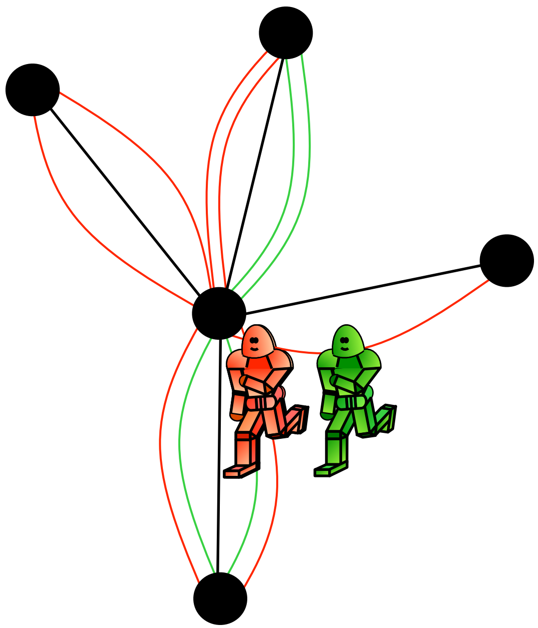
Robot colors the edges

- Initially uncolored
- Color added each time an edge is traversed

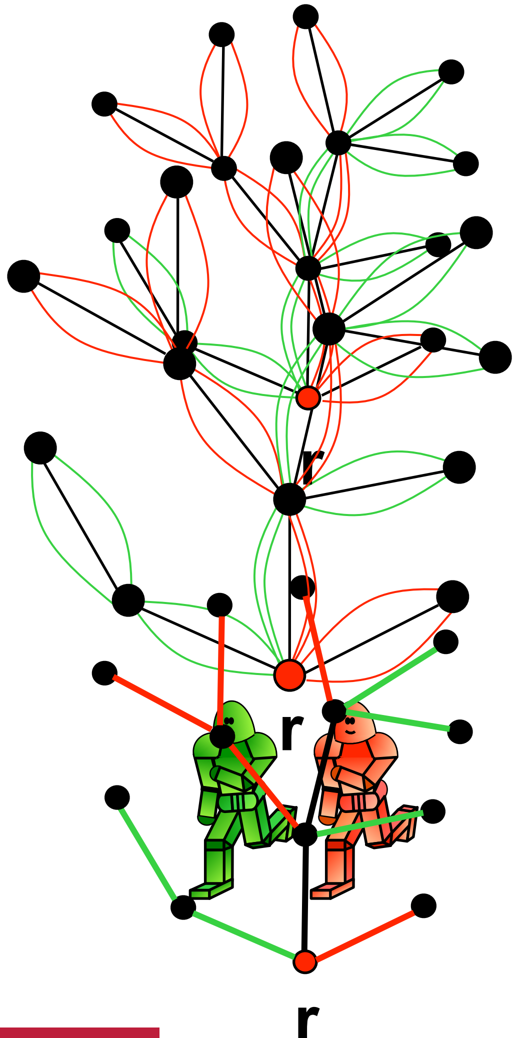
Localized strategy:

- At any crossing, pick an edge according to the following priorities:
  - uncolored edge
  - colored once
  - STOP when all edges colored twice

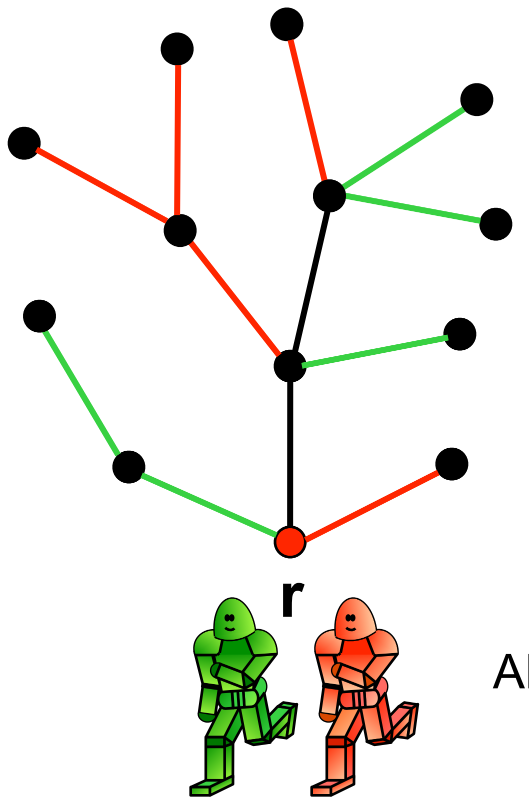
Tree Exploration with k=2:
(1) Local Rules



Tree Exploration with k=2:
(2) Emergent Structure



Upper Bound k=2:
(3) Global Quality



ALG=14

$$OPT \geq 2 \times B \qquad \times\frac{1}{2}$$

$$OPT \geq 2 \times (R + G + B) \times \frac{1}{2}$$

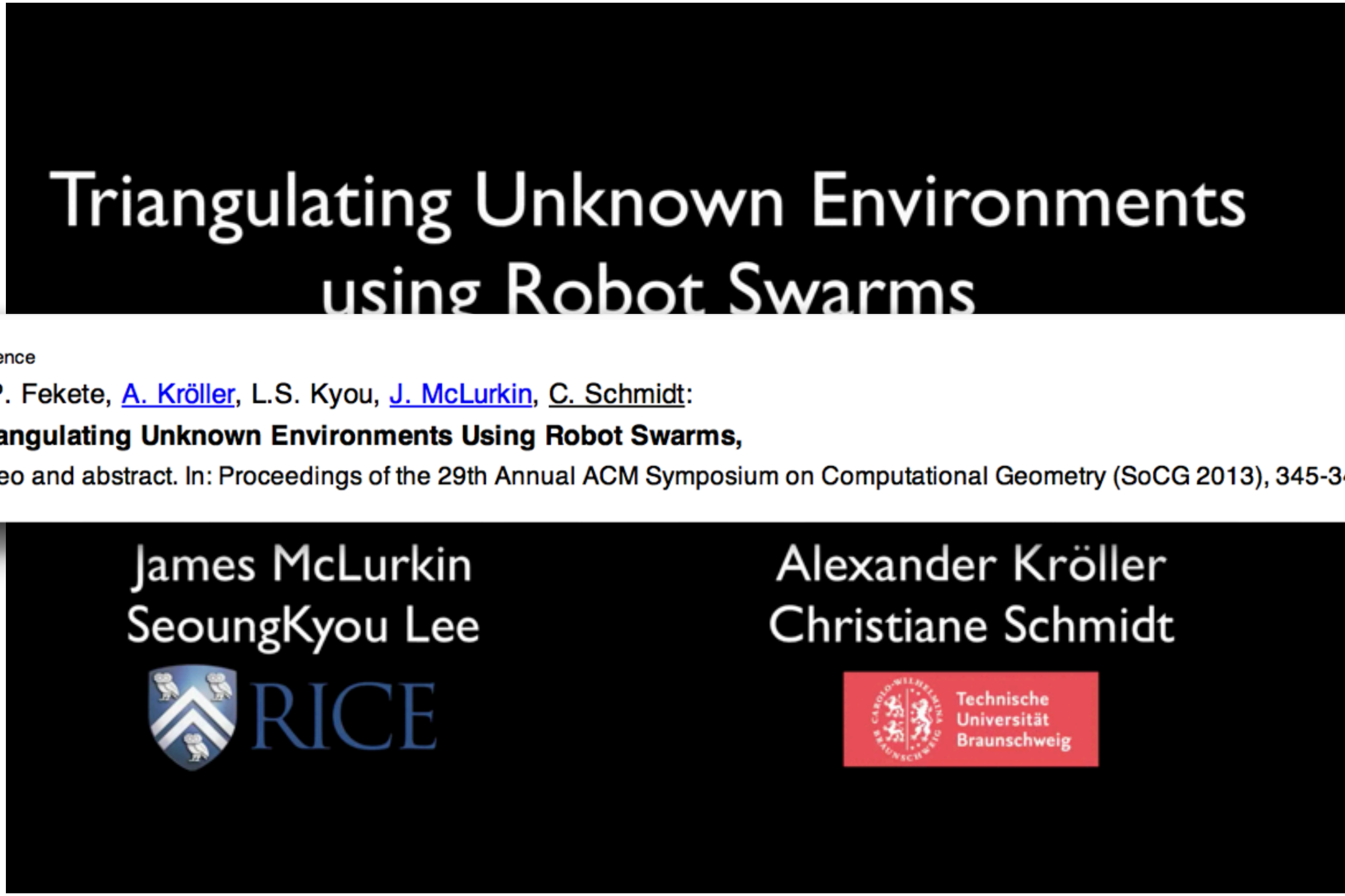
$$= 2 \times R + B \qquad \times 1$$

$$ALG \leq \frac{1}{2} \times OPT + OPT$$

$$= \frac{3}{2} \times OPT$$

Part 3.1: Online Triangulation

Video!

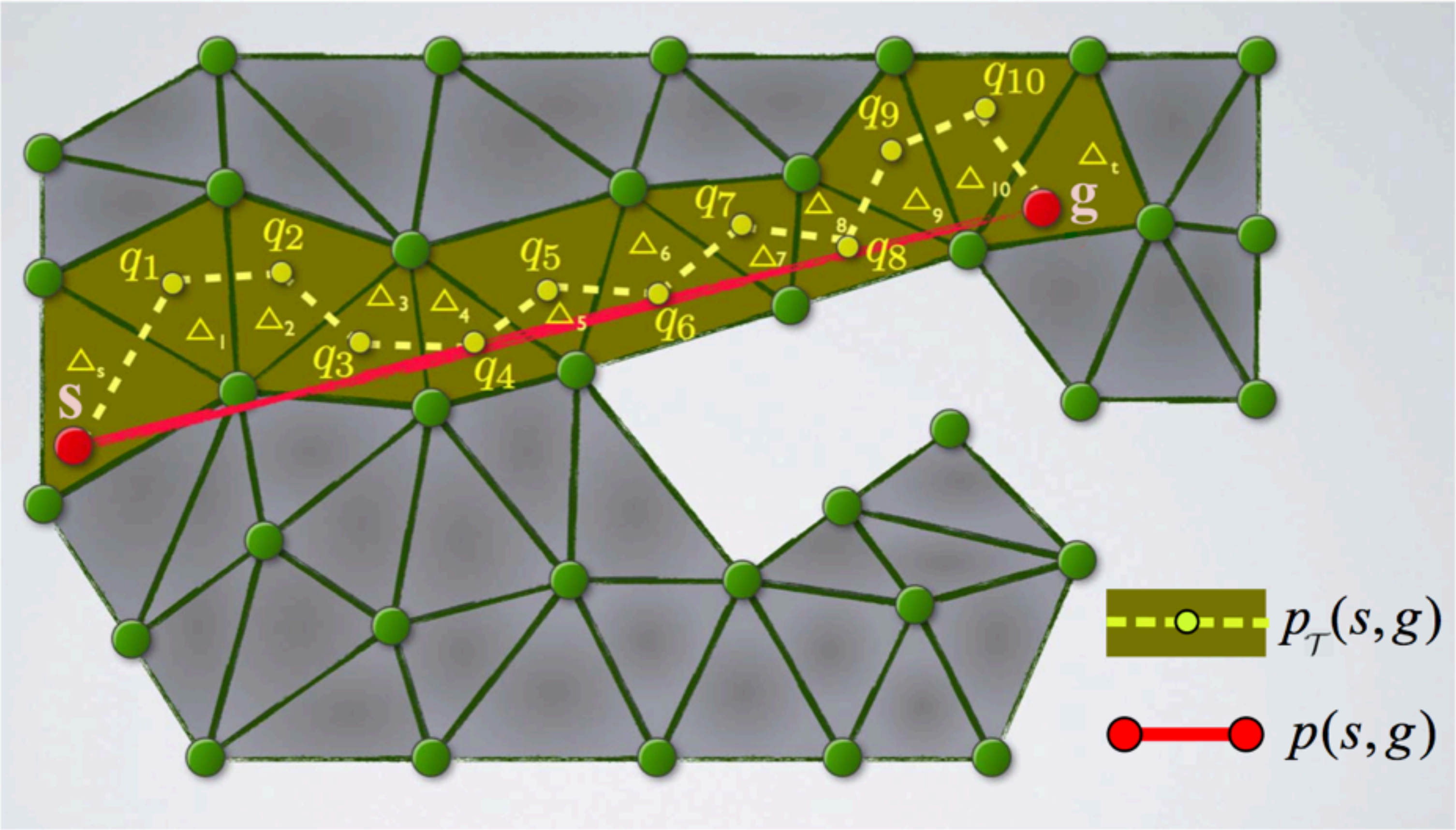


conference

S.P. Fekete, A. Kröller, L.S. Kyou, J. McLurkin, C. Schmidt:
Triangulating Unknown Environments Using Robot Swarms,
Video and abstract. In: Proceedings of the 29th Annual ACM Symposium on Computational Geometry (SoCG 2013), 345-346.

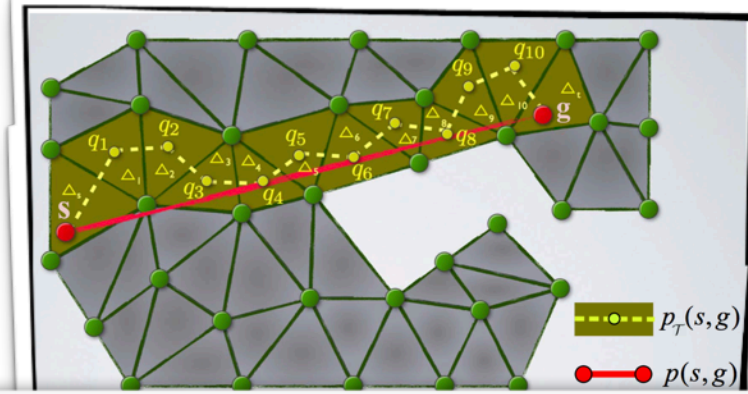
Part 3.2: Local Routing

Dual Routing



Note*:* The dual graph is stored implicitly in *primal* vertices!

Dual Routing



conference

S. K. Lee, A. Becker, S.P. Fekete, A. Kröller, J. McLurkin:

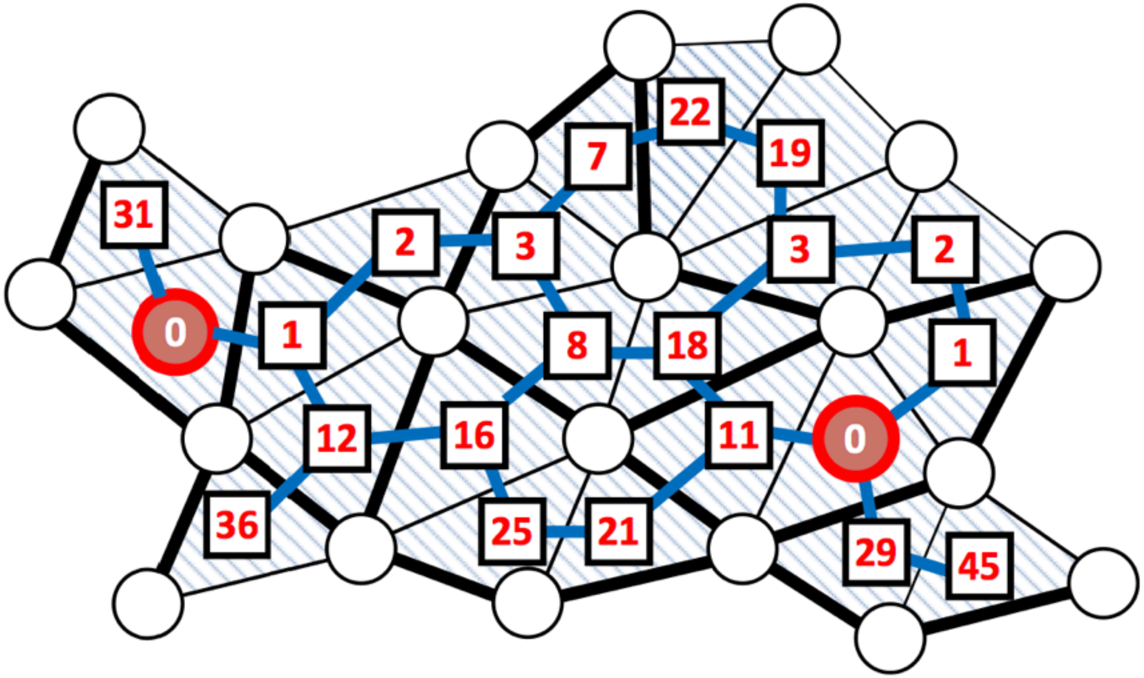
Exploration via Structured Triangulation by a Multi-Robot System with Bearing-Only Low-Resolution Sensors,

NEW To appear in: 2014 IEEE International Conference on Robotics and Automation (ICRA 2014)

in $\mathcal{R}$ that are separated by at least one triangle, i.e., the triangles Δ_s, Δ_g in $\mathcal{T}$ that contain s and g do not share a vertex. Let $p(s,g)$ be a shortest polygonal path in $\mathcal{R}$ that connects s with g, and let $d_p(s,g)$ be its length. Let $p_{\mathcal{T}}(s,g)$ be a $\mathcal{T}$-greedy path between s and g, of length $d_{p_{\mathcal{T}}}(s,g)$. Then $d_{p_{\mathcal{T}}}(s,g) \leq c \cdot d_p(s,g) + 2$, for $c = \lfloor \frac{2\pi}{\alpha} \rfloor \frac{\rho}{\sin(\alpha/2)}$, and $d_{p_{\mathcal{T}}}(s,g) \leq c' \cdot d_p(s,g)$, for $c' = \lfloor \frac{6\pi}{\alpha} \rfloor \frac{\rho}{\sin(\alpha/2)}$.

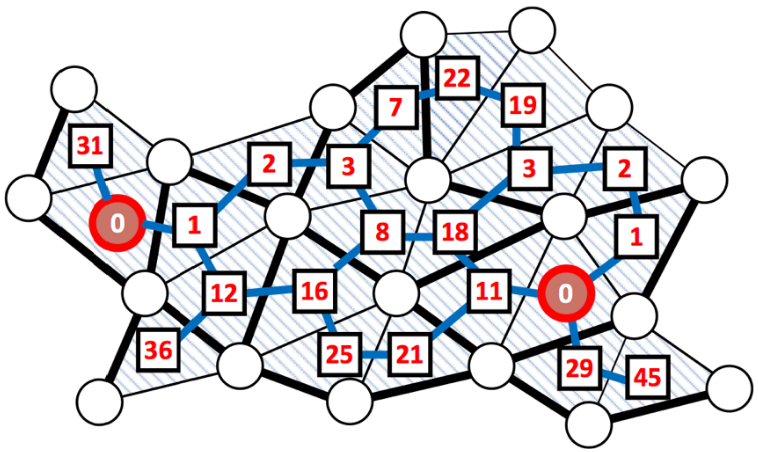
Part 3.3: Local Patrolling Policies

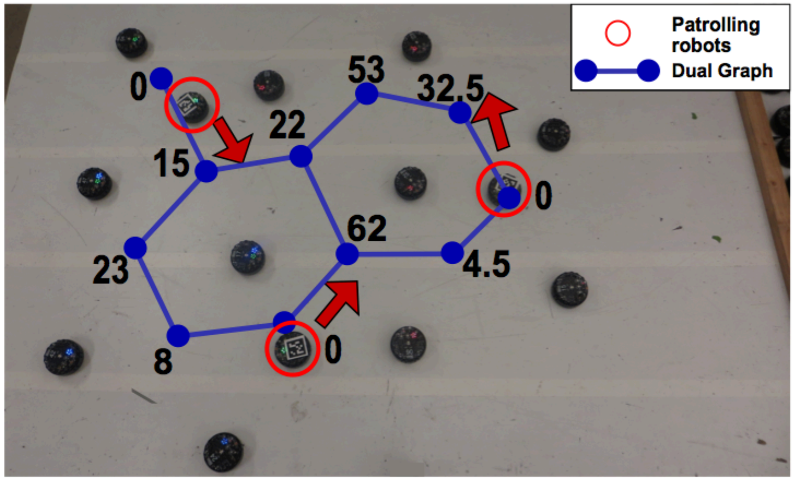
Time Stamps in the Dual Graph



Numbers: Time of last visit

Least Recently Visited



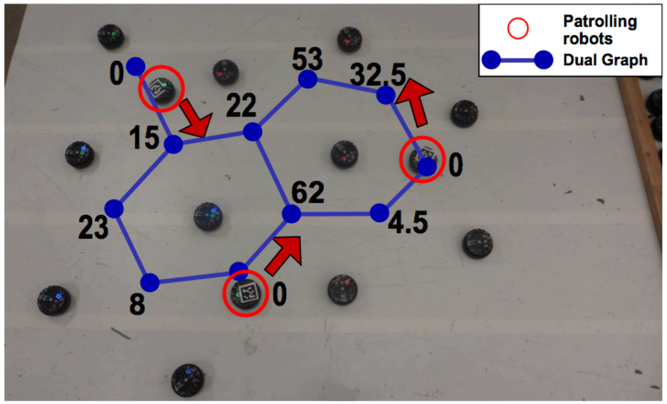


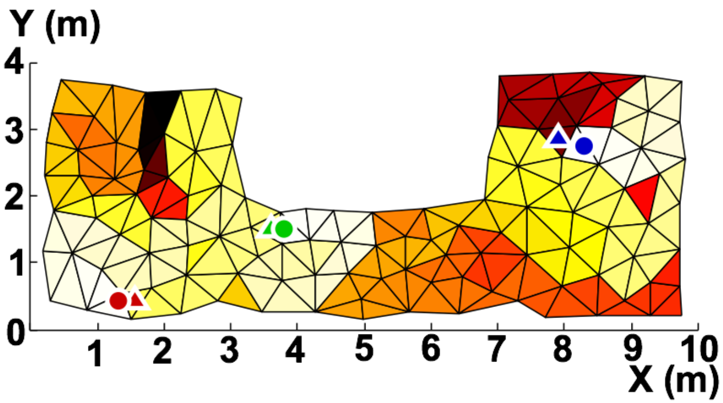
Least Recently Visited (LRV):
Move to vertex with oldest time stamp

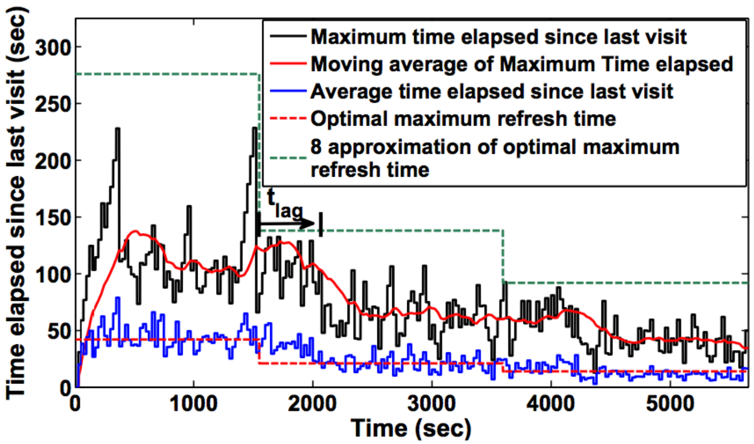
Good news: LRV achieves full coverage.

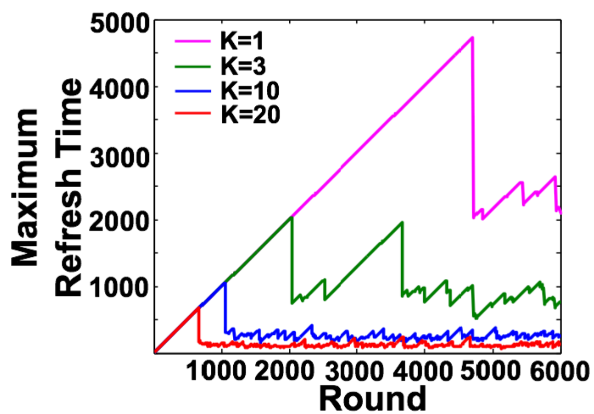
Bad news: The coverage time of LRV can be exponentially large.

LRV: Experimental Results









Part 4:
Controlling Massive Particle Swarms

Moving Small Objects

Tetrahymena pyriformis

This Part

- Massive particle swarms

conference

A. Becker, E.D. Demaine, S.P. Fekete, G. Habibi, J. McLurkin:

Reconfiguring Massive Particle Swarms with Limited, Global Control,

NEW In: ALGOSENSORS 2013, pp. 51-66, Springer LNCS 8343, 2014.

- *We establish positive results for*

conference

A. Becker, E.D. Demaine, S.P. Fekete, J. McLurkin:

Particle Computation: Controlling Robot Swarms with only Global Signals,

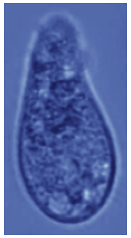
NEW To appear in: 2014 IEEE International Conference on Robotics and Automation (ICRA 2014)

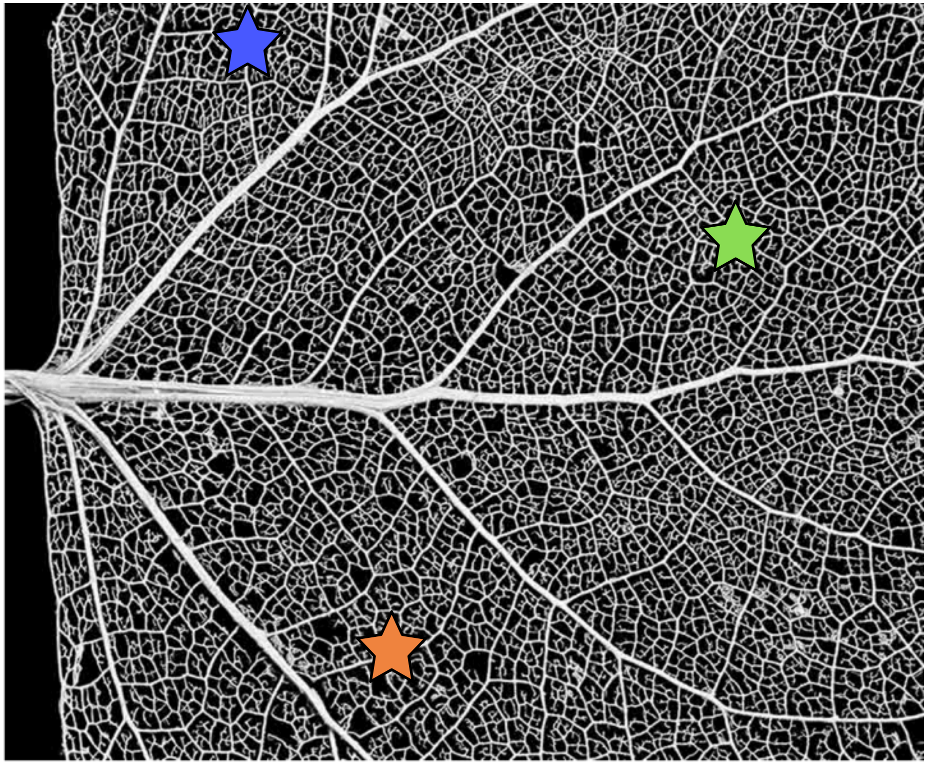
combining theory and practice

Part 4.1: Why Obstacles Are a Nuisance

Obstacles as Opponents

- Targets may not be easy to reach.
- Motion planning gets quite tricky in parallel.

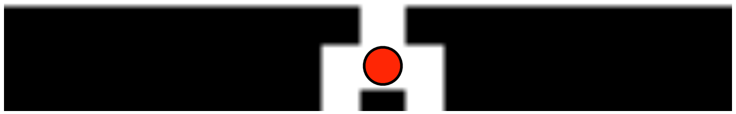


Cottonwood leaf vascular network

Complexity: Binary Variables

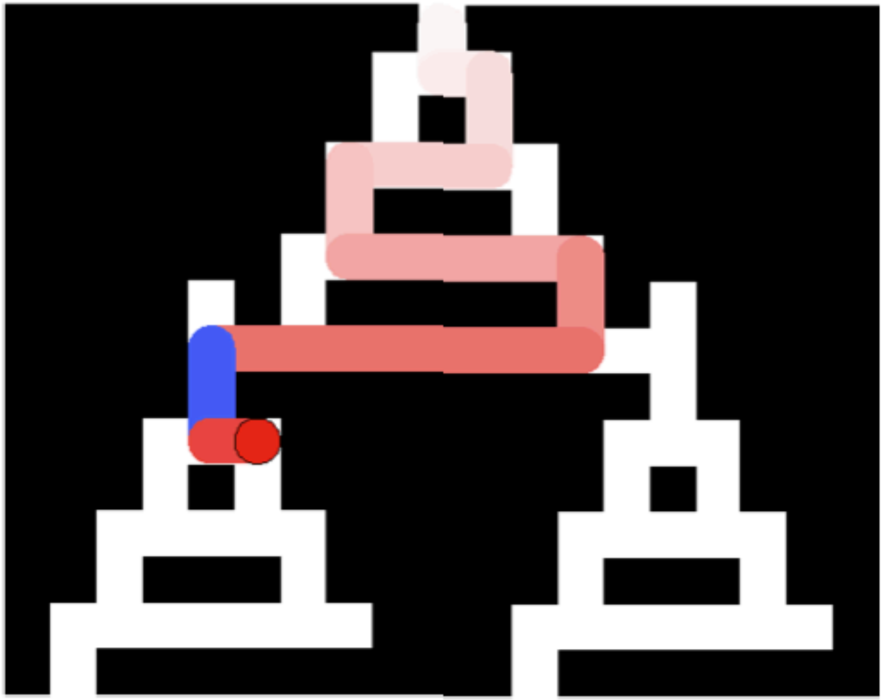
Choice: left or right?
Independent choices?!



x_2 x_3 x_4

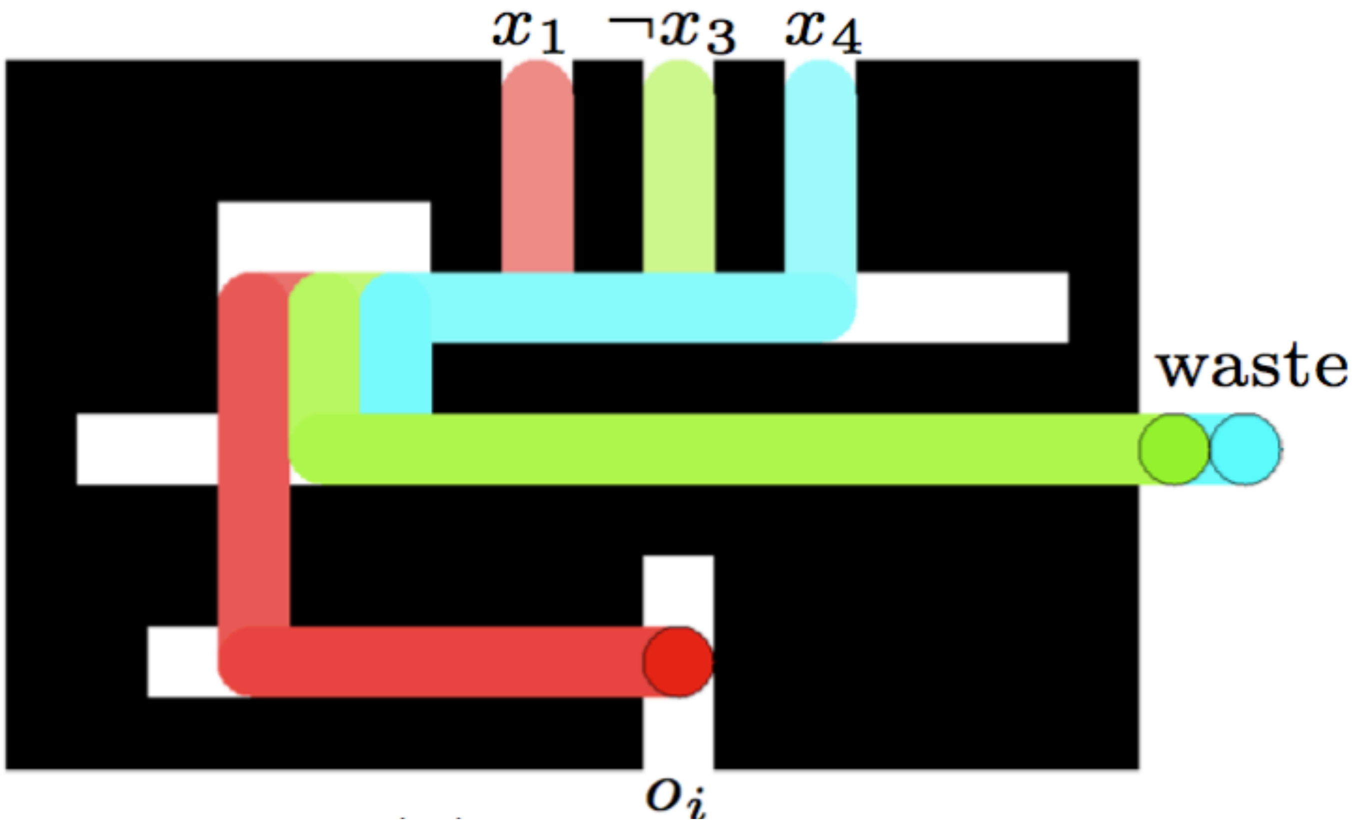
Choice only matters when it is a variable's "turn"!

Complexity: Binary Variables

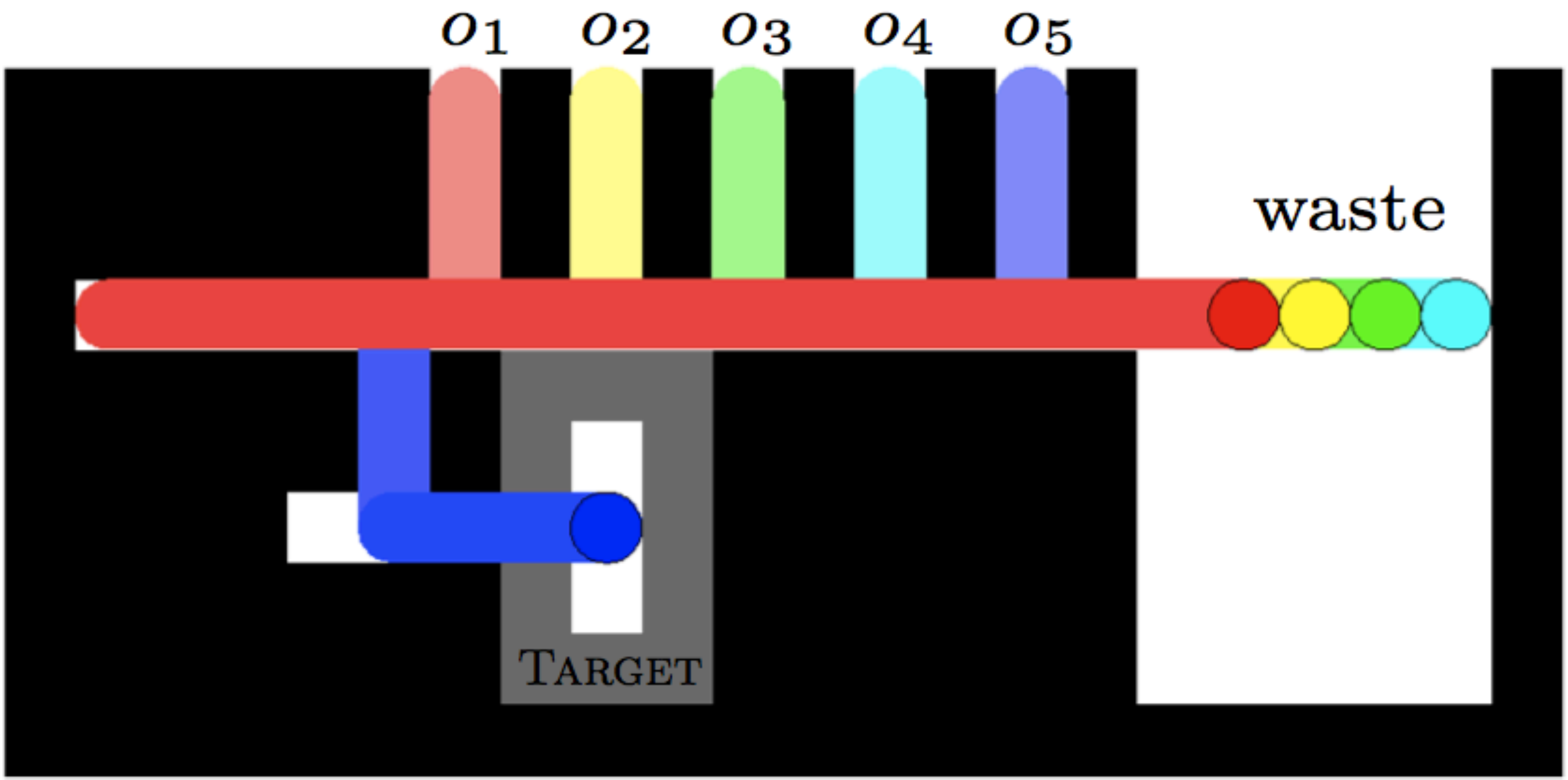


Minor detail: Avoid reversible choices!

Complexity: Clauses



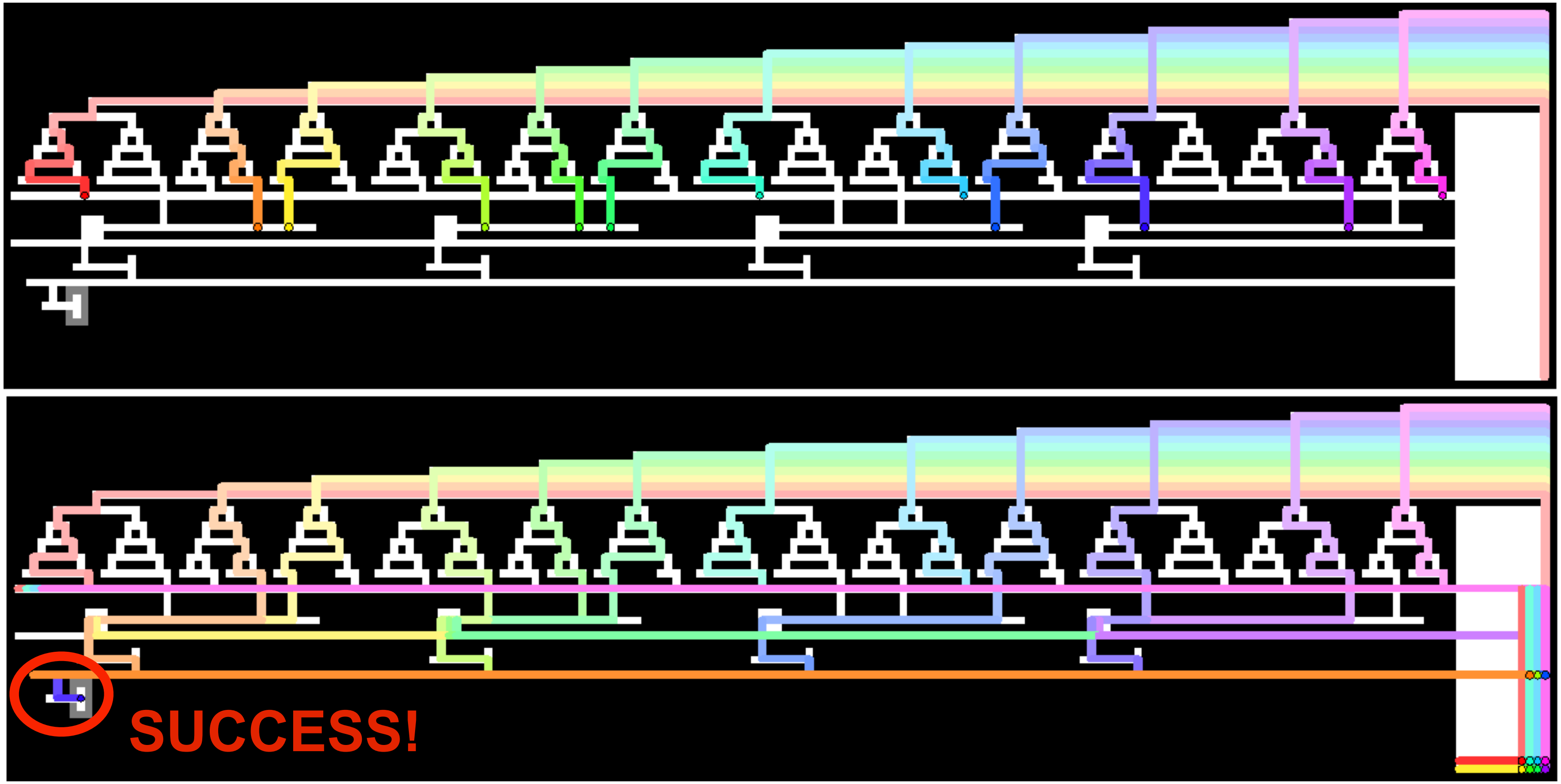
Complexity: Truth Checking



Complexity: Overall Construction

$(\neg x_1 \vee \neg x_3 \vee x_4) \wedge (\neg x_2 \vee \neg x_3 \vee x_4) \wedge (\neg x_1 \vee x_2 \vee x_4) \wedge (x_1 \vee \neg x_2 \vee x_3)$

$x_1 = 1, x_2 = 0, x_3 = 0, x_4 = 1$



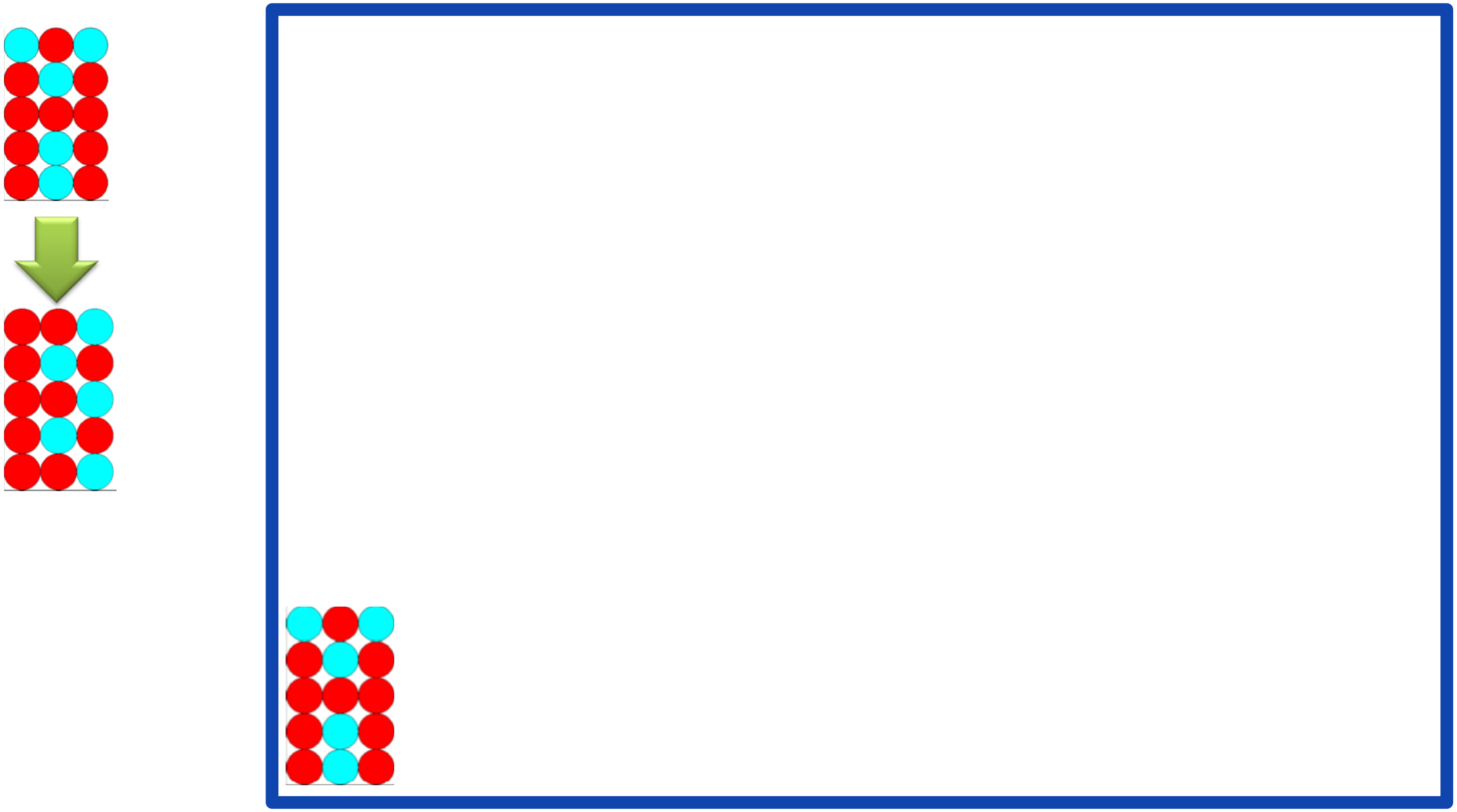
Complexity: Summary

Theorem 1. GLOBALCONTROL-MANYPARTICLES *is NP-hard: given an initial configuration of movable particles and fixed obstacles, it is NP-hard to decide whether any particle can be moved to a specified location.*

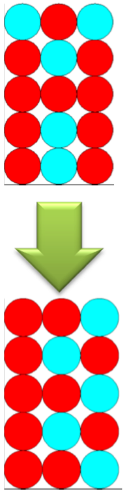
Part 4.2: Why Obstacles Are a Blessing

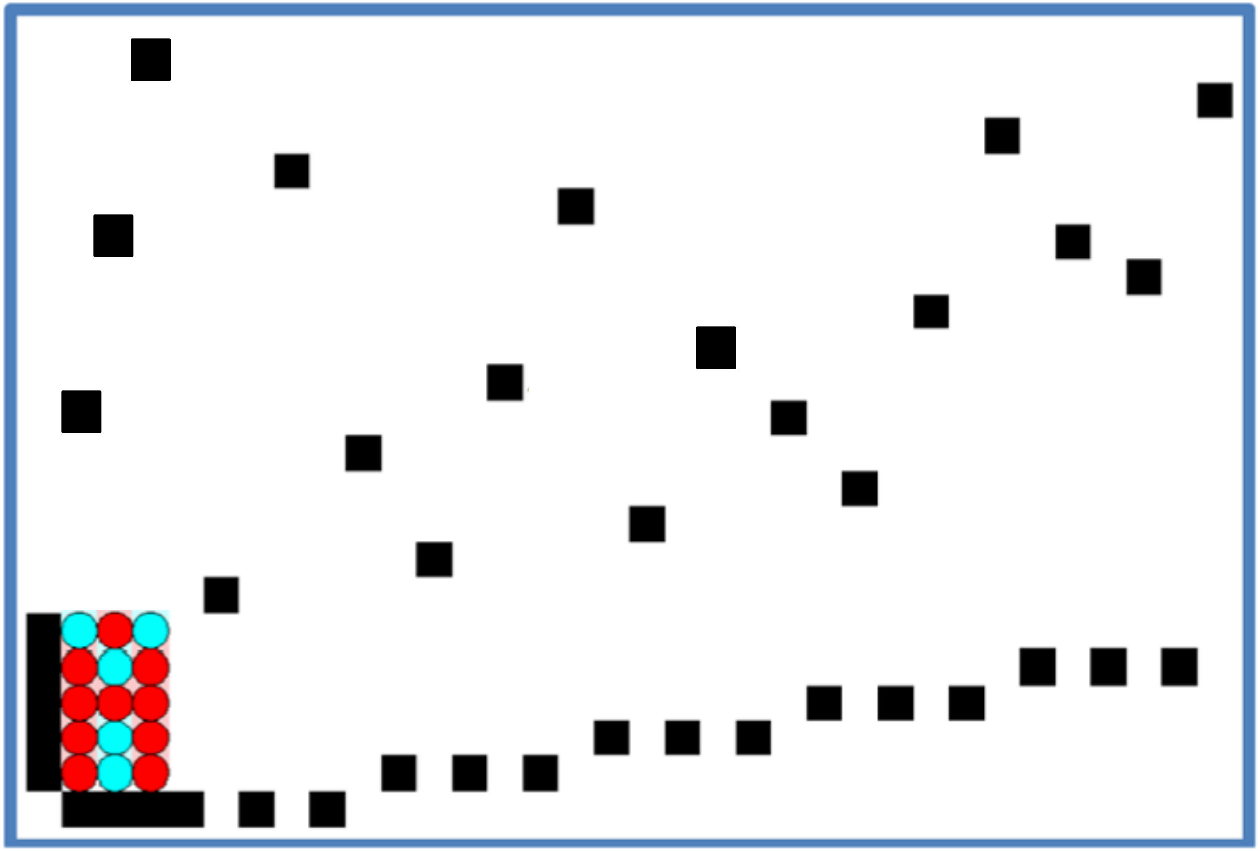
Life without Obstacles

Lack of obstacles can be harmful!

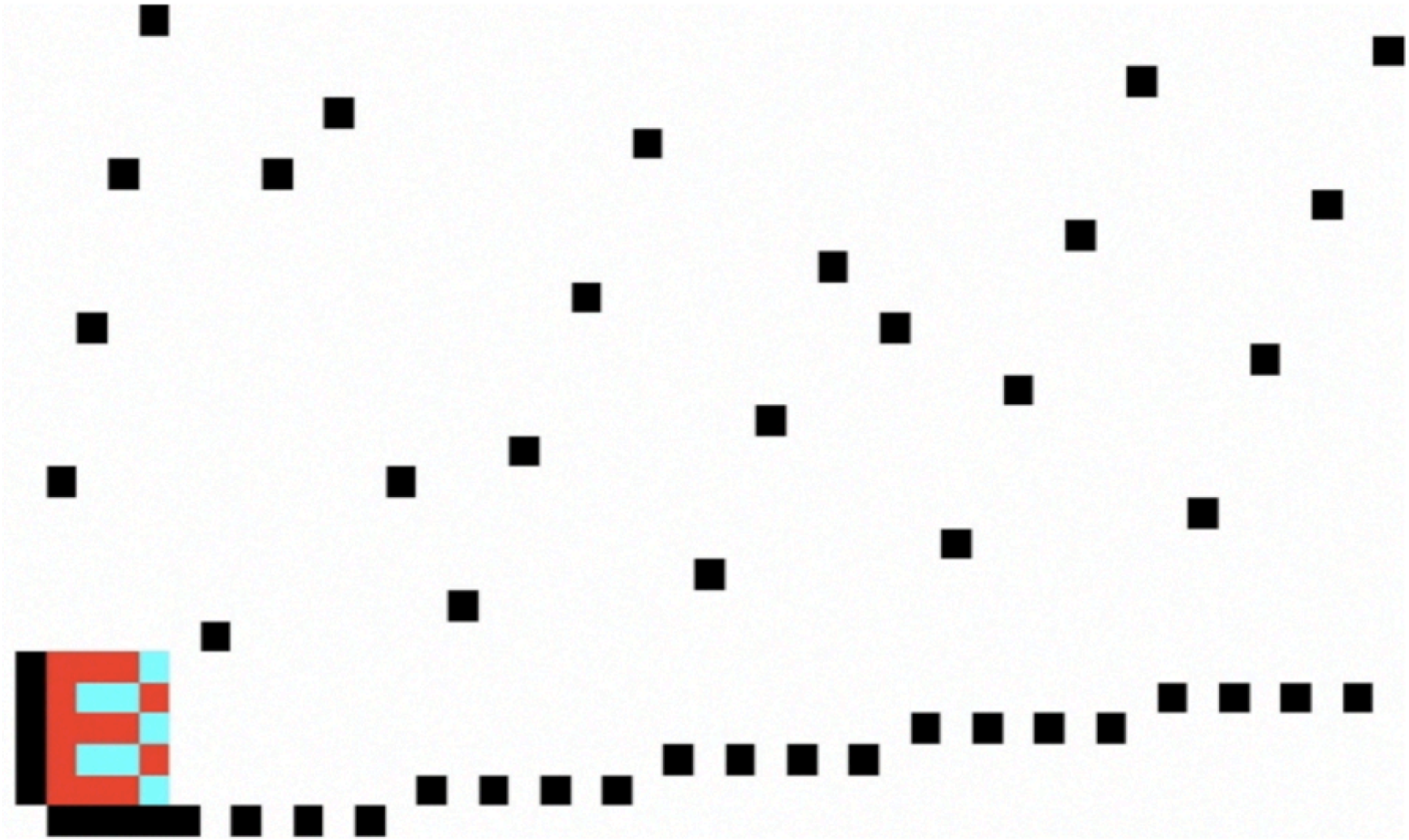


How Obstacles Can Be Helpful

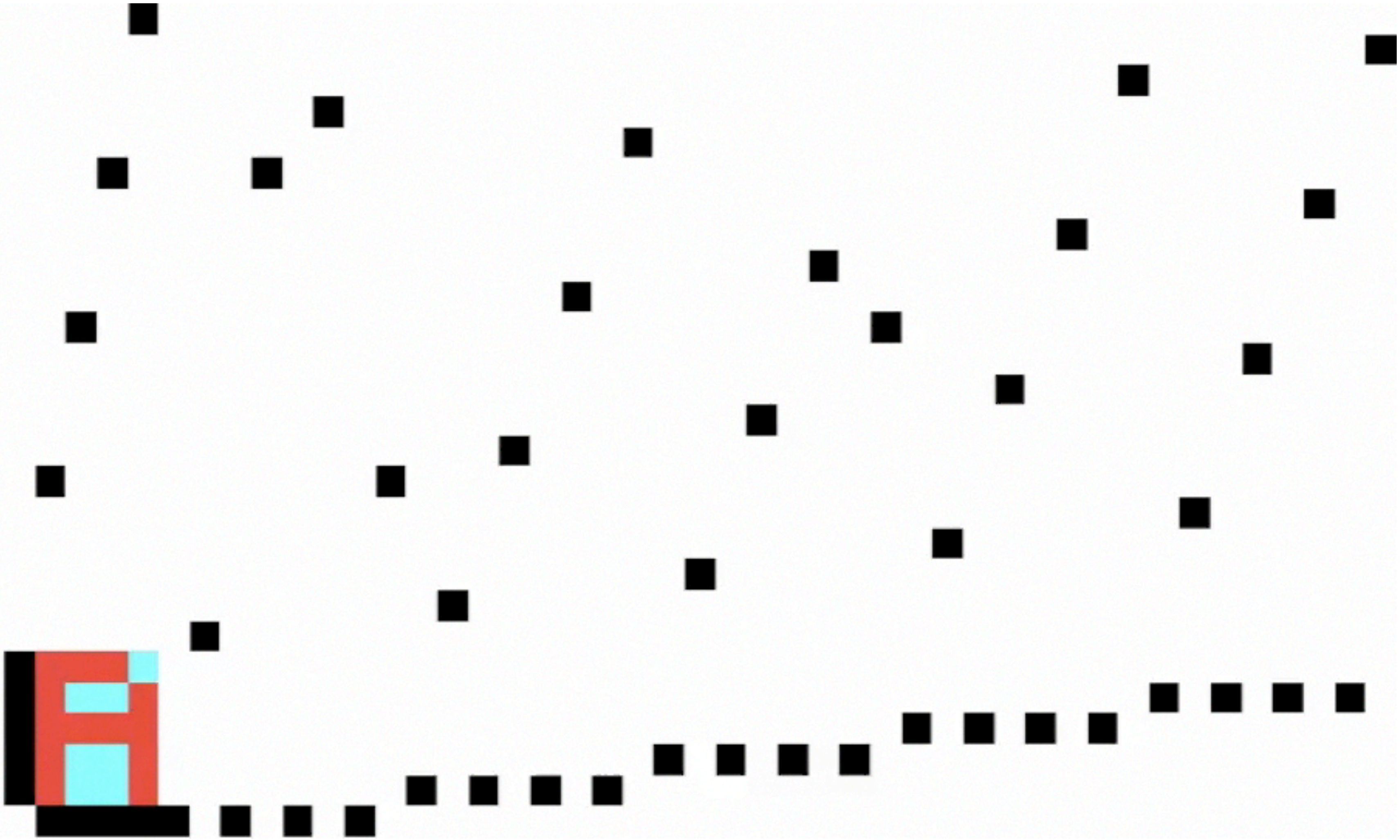




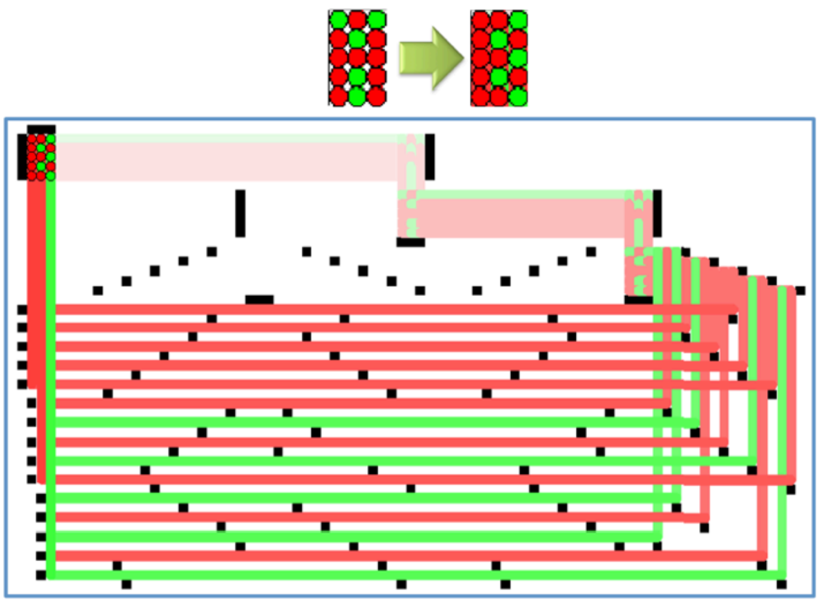
More Obstacle Action!

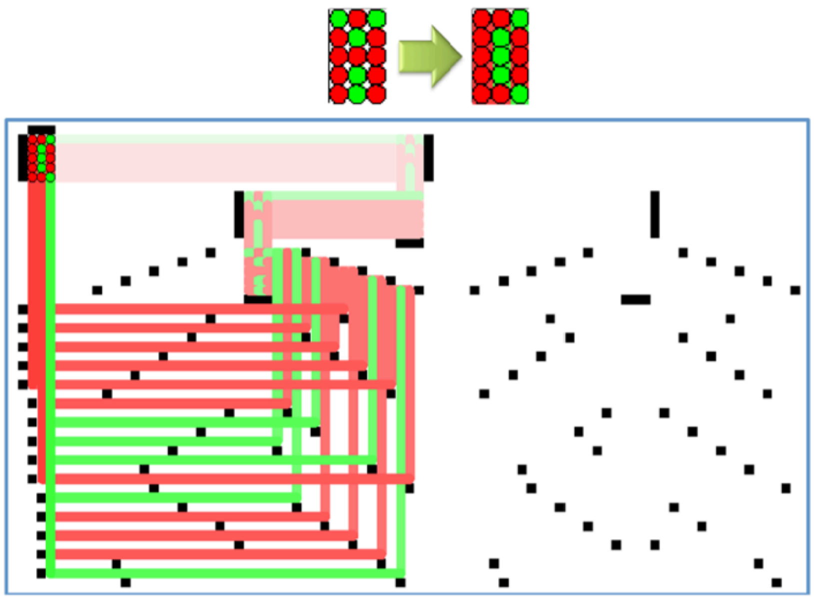


More Obstacle Action!



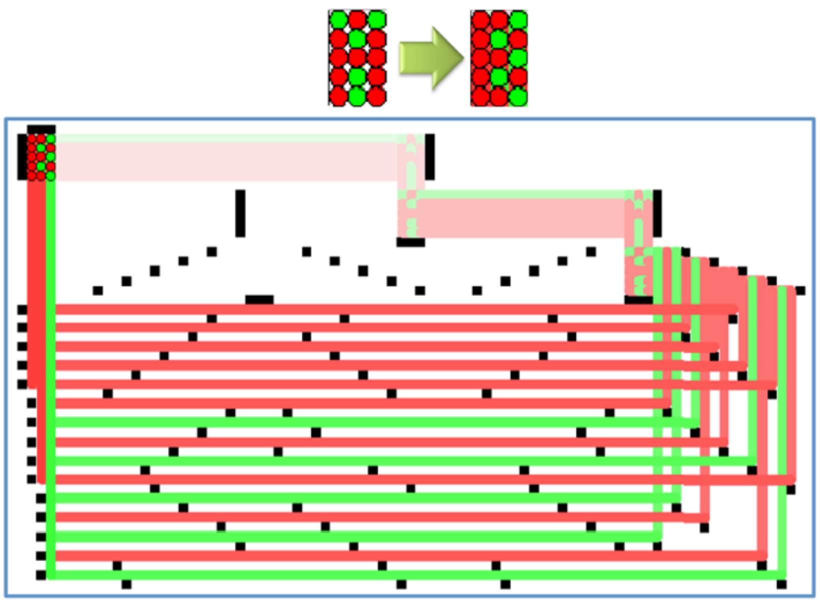
Multiple Permutations

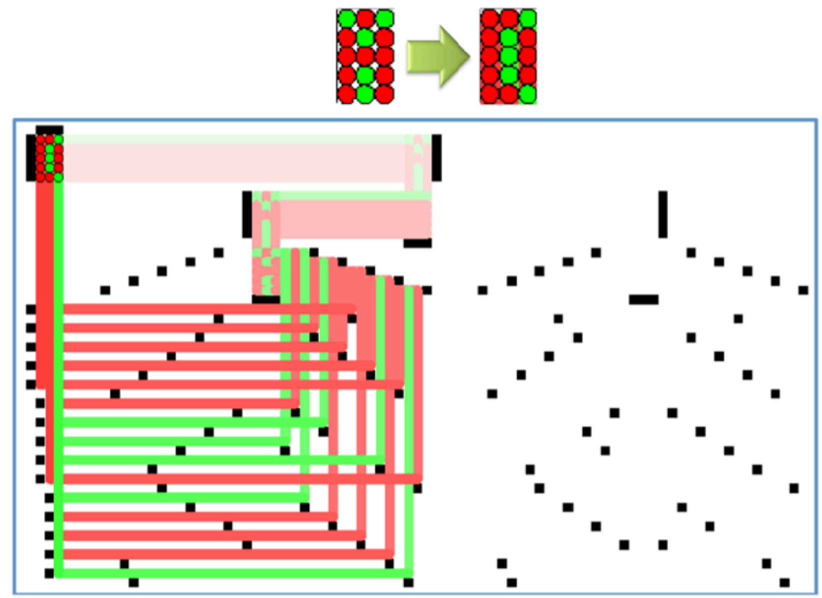




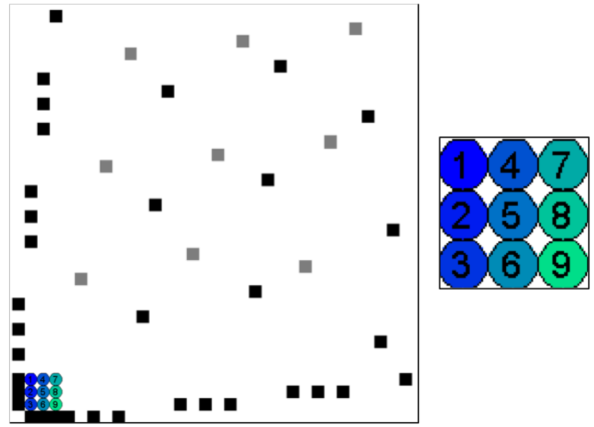
Multiple Permutations

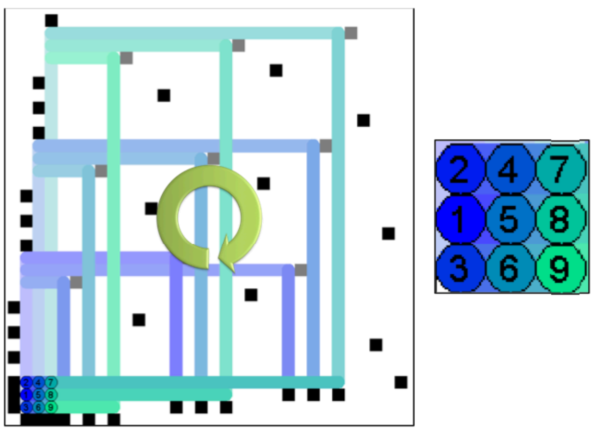
Theorem 3. *For any set of k fixed, but arbitrary, permutations of $n \times n$ pixels, we can construct a set of $O(kN)$ obstacles, such that we can switch from a start arrangement into any of the k permutations using at most $O(\log k)$ force-field moves.*



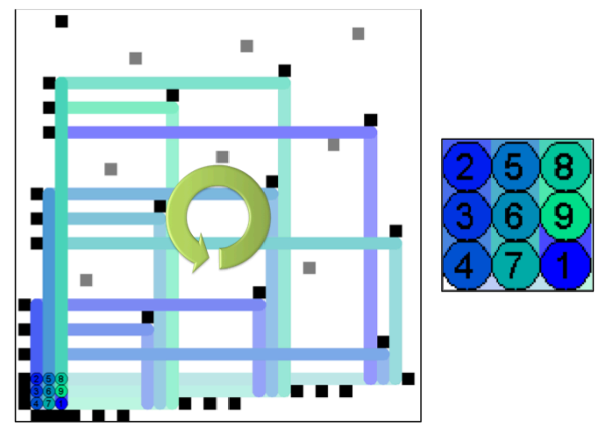


Designing Obstacles





CW: (12)

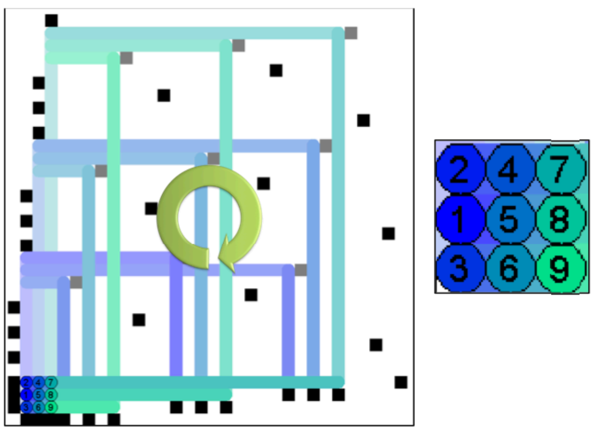


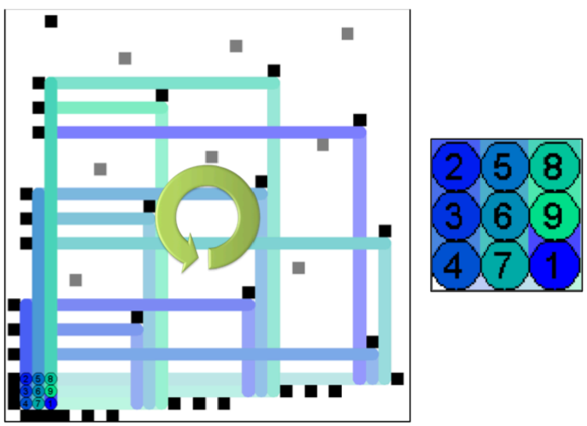
CCW: (123456789)

Designing Obstacles

Lemma 5. *Any permutation of N objects can be generated by the two base permutations $p = (12)$ and $q = (12 \cdots N)$. Moreover, any permutation can be generated by a sequence of length at most N^2 that consists of p and q.*

Theorem 6. *We can construct a set of $O(N)$ obstacles such that any $n \times n$ arrangement of N pixels can be rearranged into any other $n \times n$ arrangement π of the same pixels, using at most $O(N^2)$ force-field moves.*





CW: (12)

CCW: (123456789)

Designing Obstacles

Lemma 7. *Any permutation of N objects can be generated by the N base permutations $p_1 = (12), p_2 = (13), \ldots, p_{N-1} = (1(N-1))$ and $q = (12 \cdots N)$. Moreover, any permutation can be generated by a sequence of length at most N that consists of the p_i and q.*

Theorem 8. *We can construct a set of $O(N^2)$ obstacles such that any $n \times n$ arrangement of N pixels can be rearranged into any other $n \times n$ arrangement π of the same pixels, using at most $O(N \log N)$ force-field moves.*

Theorem 9. *Suppose we have a set of obstacles such that any permutation of an $n \times n$ arrangement of pixels can be achieved by at most M force-field moves. Then M is at least $\Omega(N \log N)$.*

Proof. Each permutation must be achieved by a sequence of force-field moves. Because each decision for a force-field move $\{u, d, l, r\}$ partitions the remaining set of possible permutations into at most four different subsets, we need at least $\Omega(\log(N!)) = \Omega(N \log N)$ such moves. □

More on Complexity!

Theoretical Computer Science 36 (1985) 265–289
North-Holland

THE COMPLEXITY OF FINDING MINIMUM-LENGTH GENERATOR SEQUENCES

Mark R. JERRUM

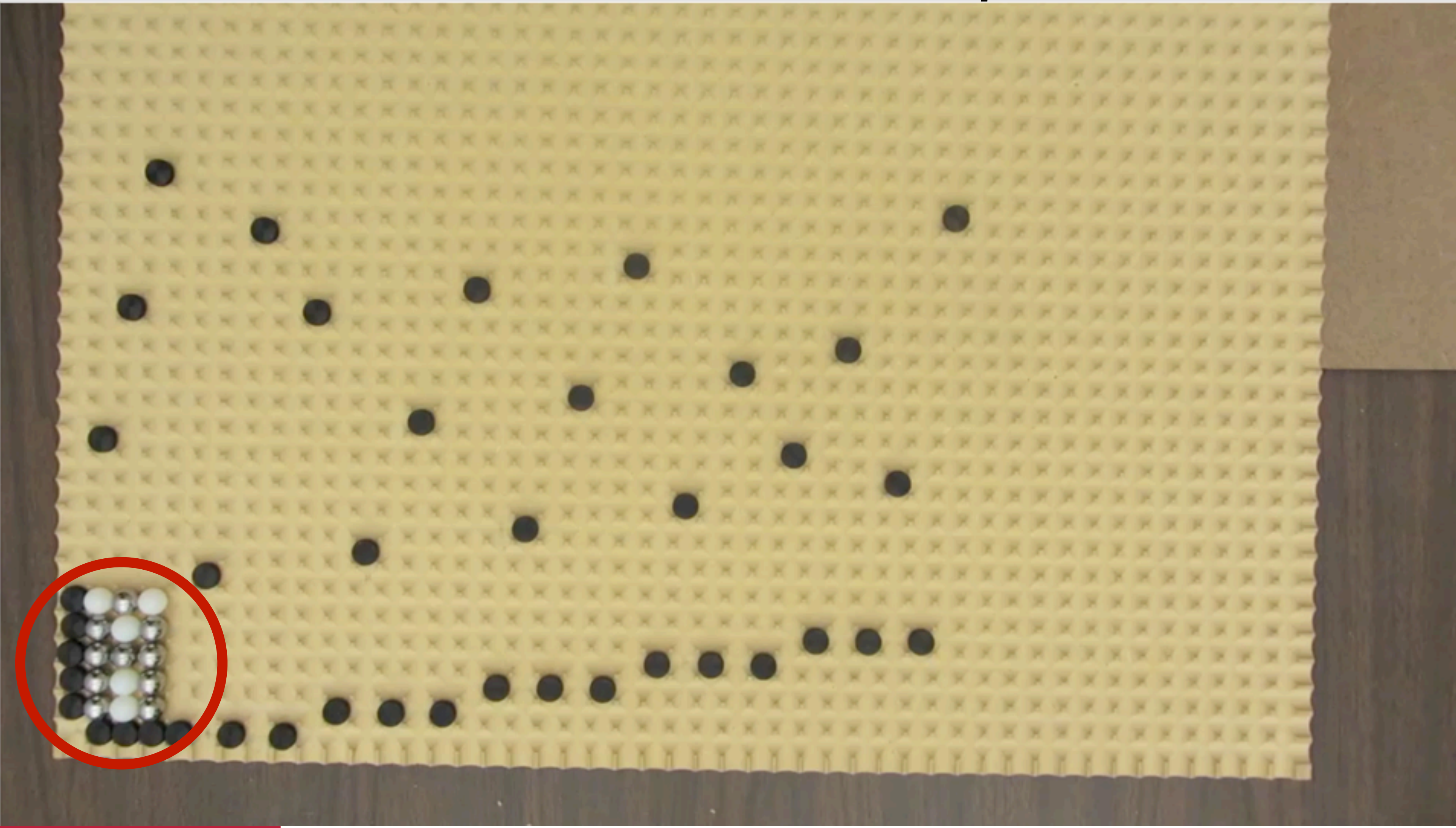
Department of Computer Science, University of Edinburgh, Edinburgh EH9 3JZ, Scotland (United Kingdom)

Communicated by M.S. Paterson
Received July 1983
Revised May 1984

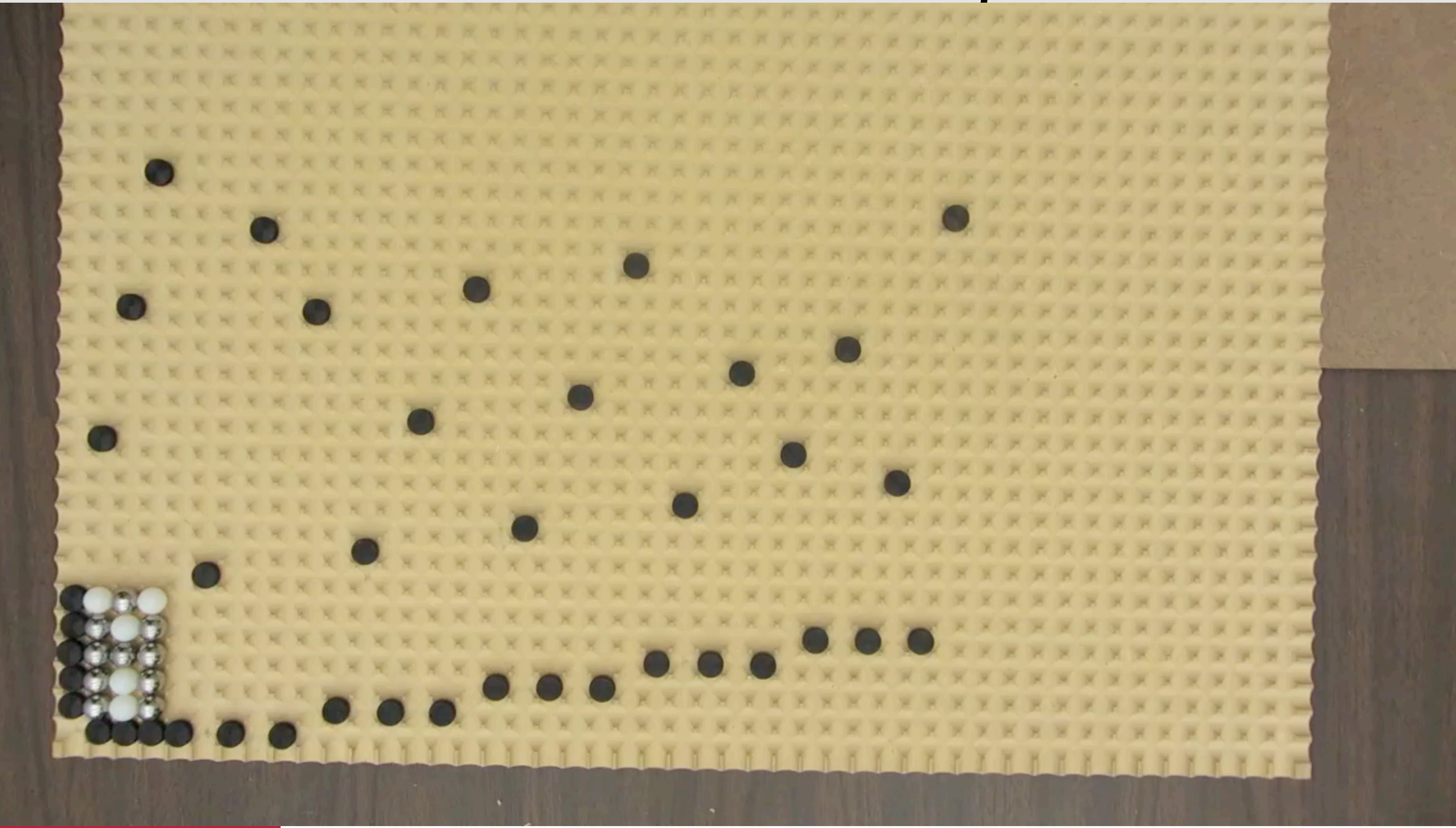
Abstract. The computational complexity of the following problem is investigated: Given a permutation group specified as a set of generators, and a single target permutation which is a member of the group, what is the shortest expression for the target permutation in terms of the generators? The general problem is demonstrated to be PSPACE-complete and, indeed, is shown to remain so even when the generator set is restricted to contain only two permutations. The restriction on generator set cardinality is the best possible, as the problem becomes soluble in polynomial time if the generator set contains only one permutation. An interesting feature of this problem is that it does not fall under the headings of 'two person games' or 'formal languages' which cover the great majority of known PSPACE-complete problems. Some restricted versions of the problem.

Part 4.3: A Real-World Demo!

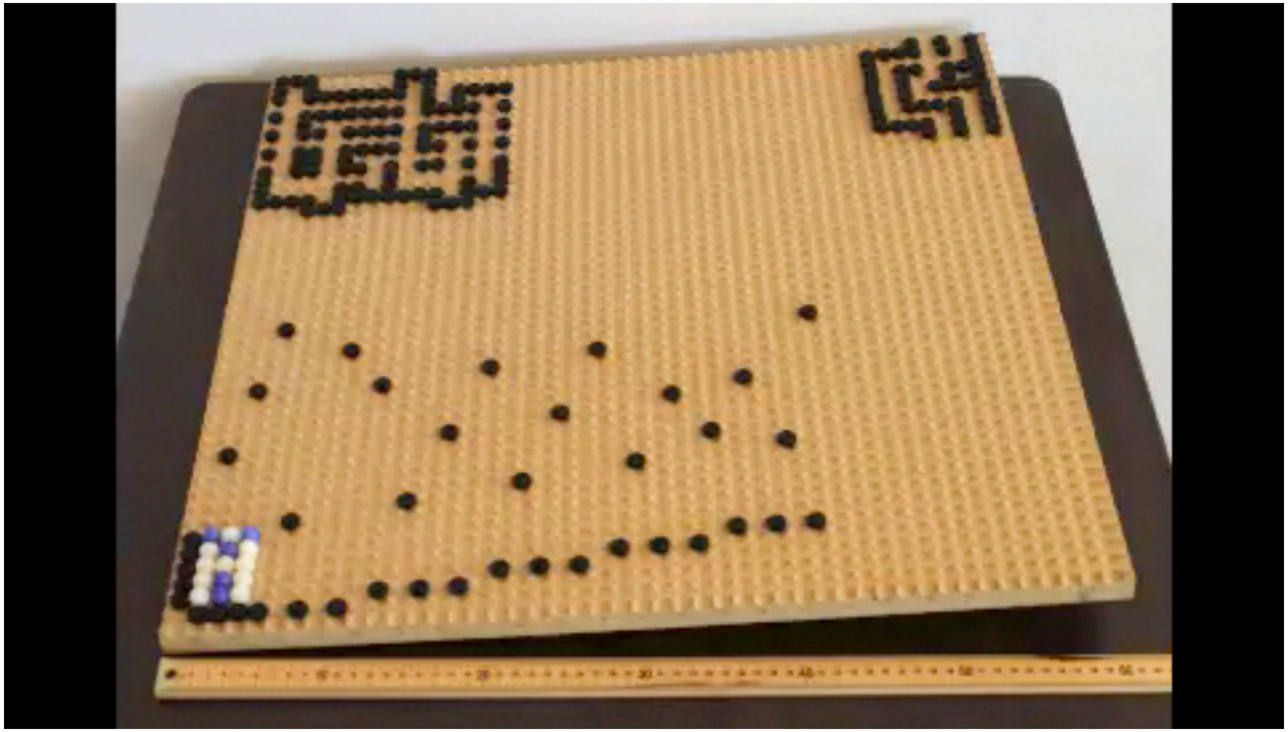
Demo with Real Objects



Demo with Real Objects



Demo II



Conclusions

- More work in theory and practice!

Thank you!