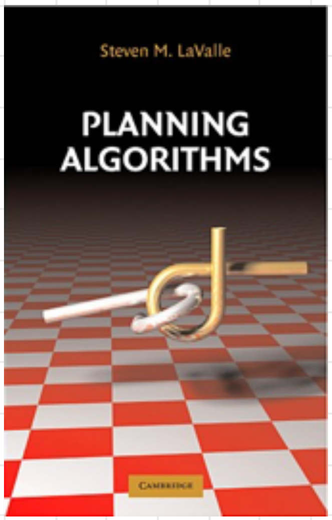
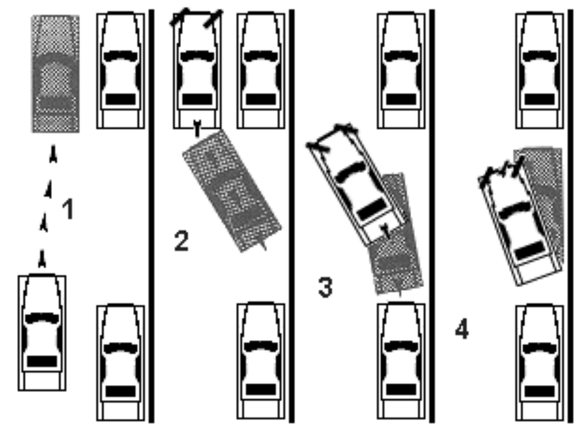
Moving objects in space with obstacles/constraints.

Objects = robots, vehicles, jointed linkages (robot arm), tools (e.g. on automated assembly line), foldable/bendable objects.
Objects need not be physical (e.g. “fly-through” animation).

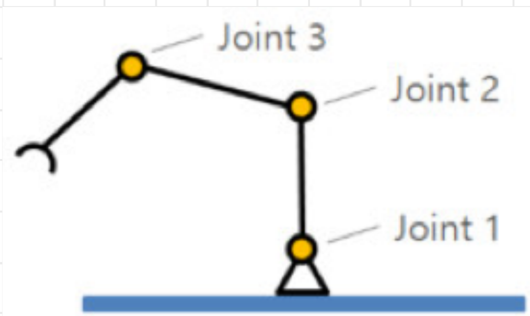
We will concentrate on moving from one position to another, though visiting a sequence of positions is also very interesting.



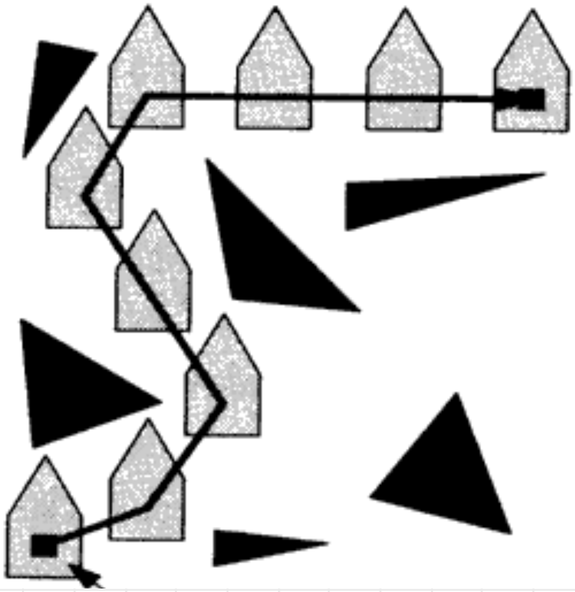


Outline:

- translational motion (one rigid object)
- linkage motion (robot arm)



Translational motion



a polygon translating
among polygonal obstacles.

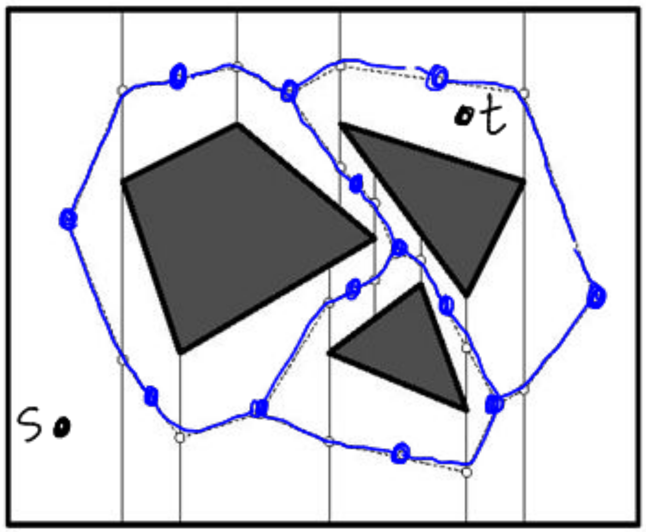
Start with a point moving among polygonal obstacles.

Then we can use the shortest path algorithm from previous lecture.

But we do not really need the shortest path.

A point moving among polygonal obstacles

How to find if there is *some* path from point s to point t among polygonal obstacles.



the blue graph is called a ***roadmap***

- construct trapezoidal map of space outside obstacles

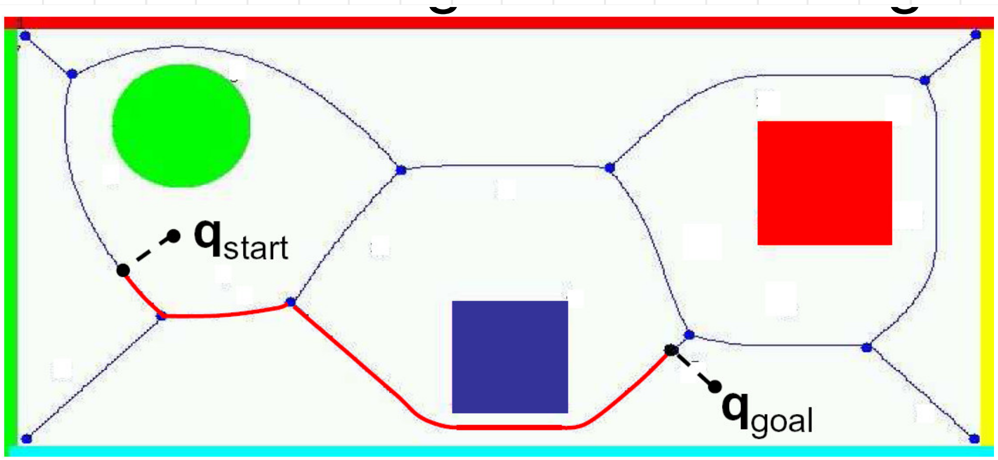
- construct dual graph (in blue above)

- check if Trapezoid(s) and Trapezoid(t) are connected in the dual graph

- time O(n log n)

A point moving among polygonal obstacles

An alternative roadmap: the Voronoi diagram of the obstacles.

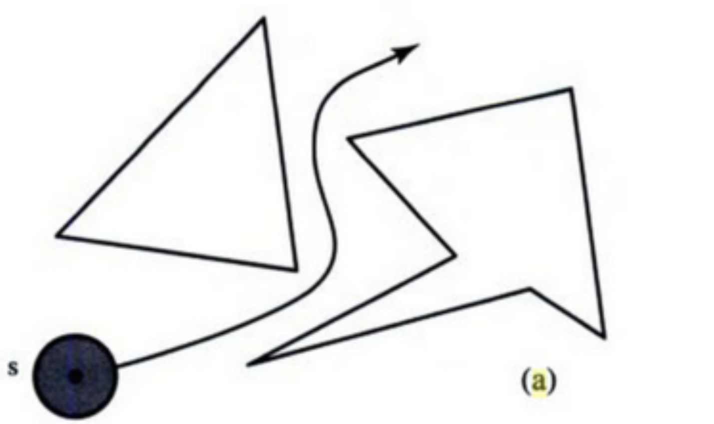


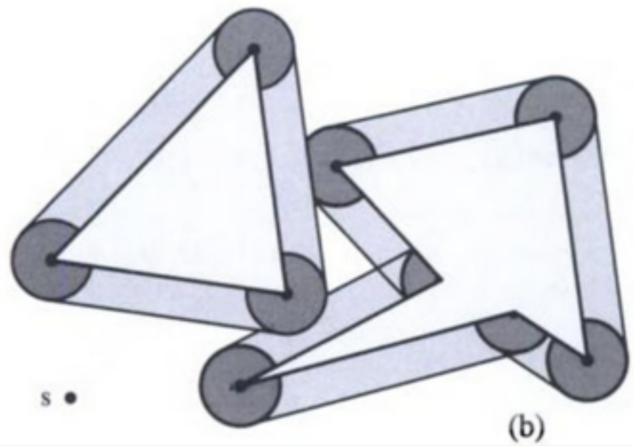
Then, for a given route, the point stays as far as possible from the obstacles.

A disc moving among polygonal obstacles.

Model as a point (the center of the disc) moving among enlarged obstacles.

disc radius = r
the center of the disc must stay distance $\geq r$ from the obstacles





O'Rourke

the disc cannot follow this path

"Enlarging the obstacles" is captured more formally via ***Minkowski sum***, $A \oplus B$

Minkowski sum

Let A and B be sets of points in the plane.

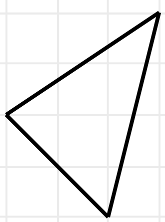
Definition. The **Minkowski sum** of A and B is

$$A \oplus B = \{ x + y : x \in A, y \in B \}$$

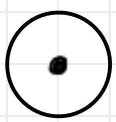
as vector addition of points

$x_0 \oplus B = \{ x_0 + y : y \in B \}$ = translate B by vector x_0
so $A \oplus B$ = translate B by all possible points in A

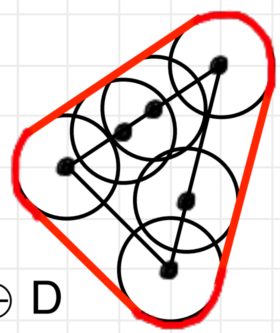
Let P = polygon, D = disc centered at (0,0)
Then $P \oplus D$ = union of copies of D placed at each point of P



P



D



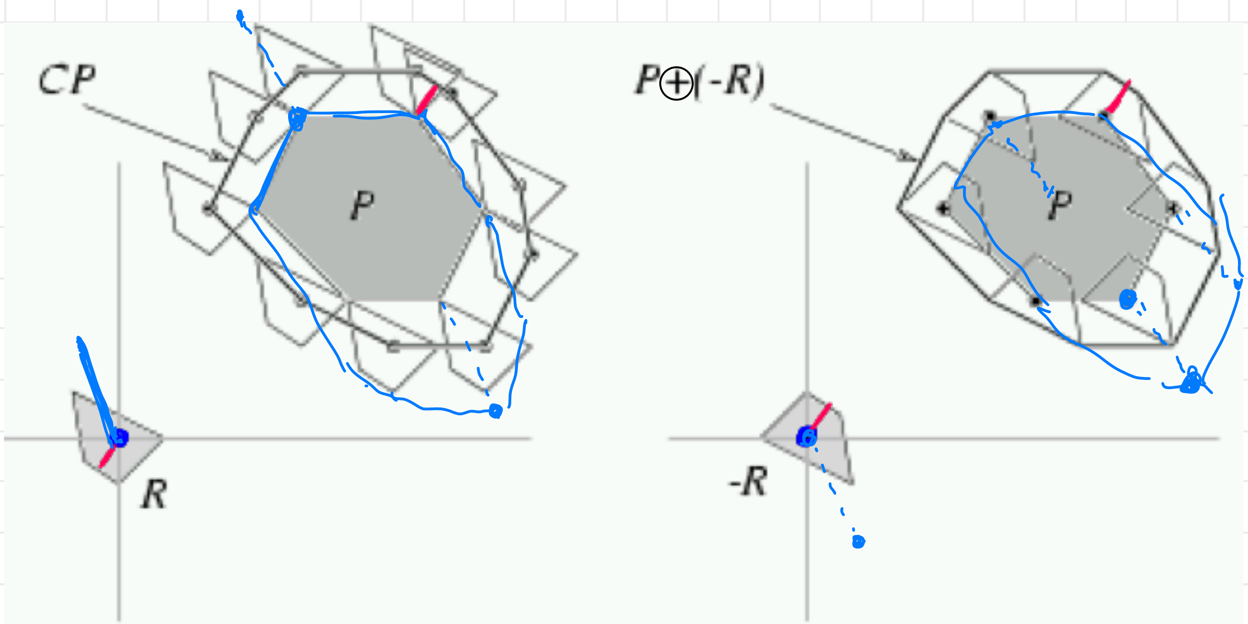
$P \oplus D$

A convex polygon moving among polygons — by translation only

R = moving polygon
P = obstacle polygon

Replace R by a ***reference point*** at the origin. We will move the reference point.

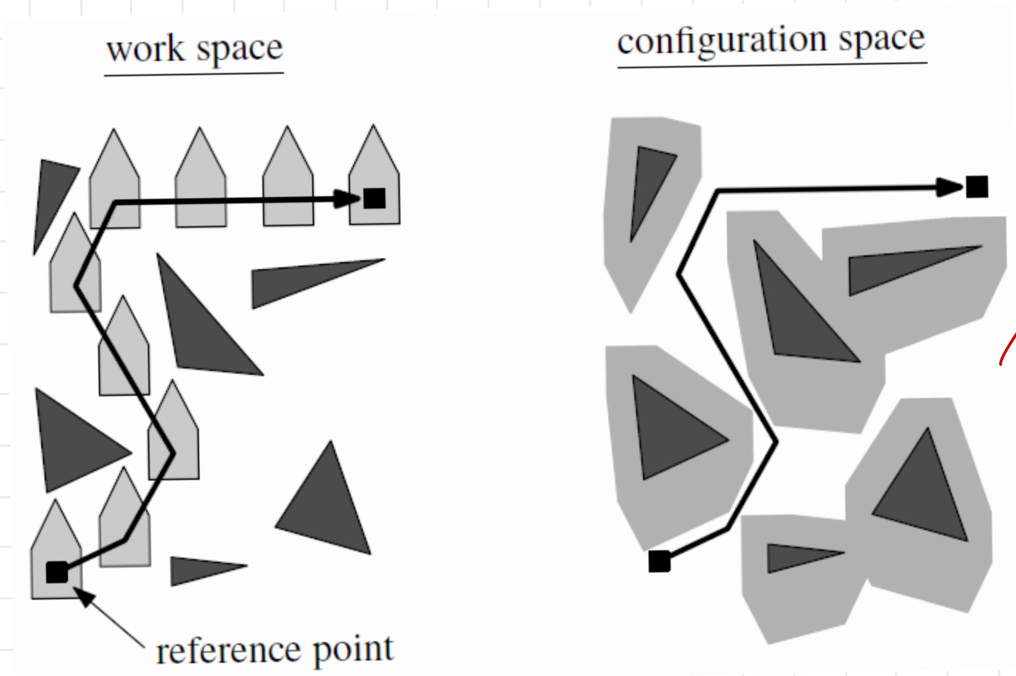
Enlarge P to compensate — we need $P \oplus (-R) = \{x - y : x \in P, y \in R\}$



where can the reference point be when R touches P?

$= P \oplus (-R)$

Can polygon R move (via translations) from initial to final position among polygonal obstacles?



[CGAA]

free space

High level idea

1. compute the Minkowski sum P $\oplus$ (–R) for each obstacle P

2. take the union, to obtain new polygonal obstacles

3. test if a point (the reference point) can move from initial to final position among the new enlarged obstacles
   in the free space

Can polygon R move (via translations) from initial to final position among polygonal obstacles?

High level idea

1. compute the Minkowski sum $P \oplus (-R)$ for each obstacle P
2. take the union, to obtain new polygonal obstacles
3. test if a point (the reference point) can move from initial to final position among the new enlarged obstacles

What we will cover:

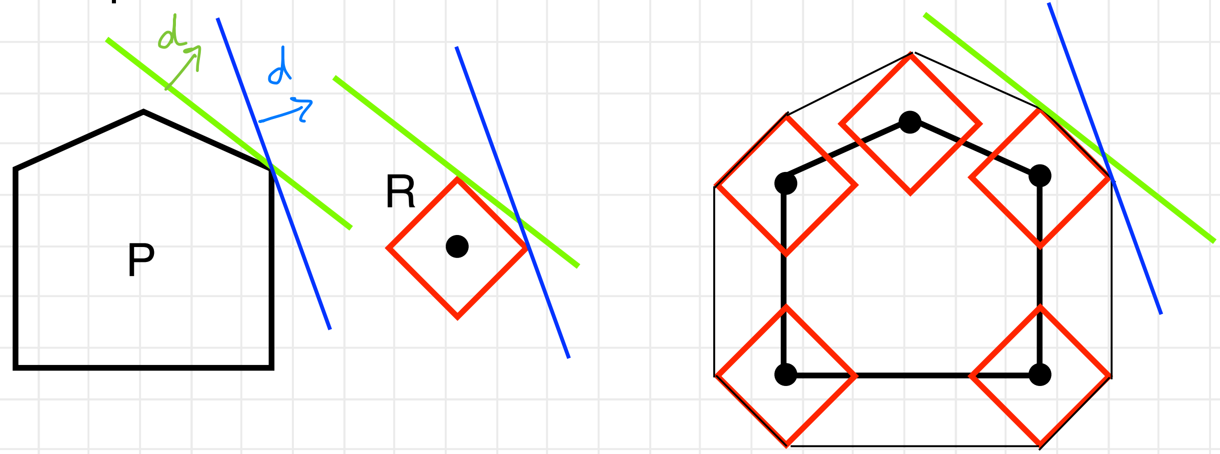
- the case where obstacles and R are convex
    - computing the Minkowski sum of two convex polygons
    - computing the union of convex Minkowski sums
- the idea of handling non-convex polygons

The Minkowski sum of two convex polygons

Theorem. If P and R are convex polygons with n and m edges, respectively, then $P \oplus R$ is convex with at most n+m edges and can be found in O(n+m) time.

Proof Let P have vertices $p_1, p_2, \ldots p_n$. Let R have vertices $r_1, r_2, \ldots, r_m$.

Claim. Vertices of $P \oplus R$ have the form $p_i + r_j$.
Stronger Claim. The vertex (extreme point) of $P \oplus R$ in direction d is the sum of the extreme points of P and R in direction d.



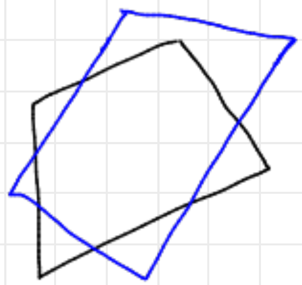
How to find $P \oplus R$

rotate direction d
Each time the extreme vertex (of P or R) changes,
output corresponding vertex of $P \oplus R$.

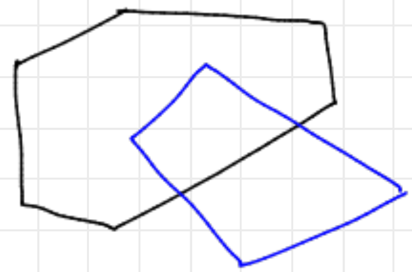
Computing the union of convex Minkowski sums

The complexity of the union is reduced due to the following:

two convex polygons' boundaries can intersect many times



they form ***pseudodiscs*** if their boundaries intersect at most twice



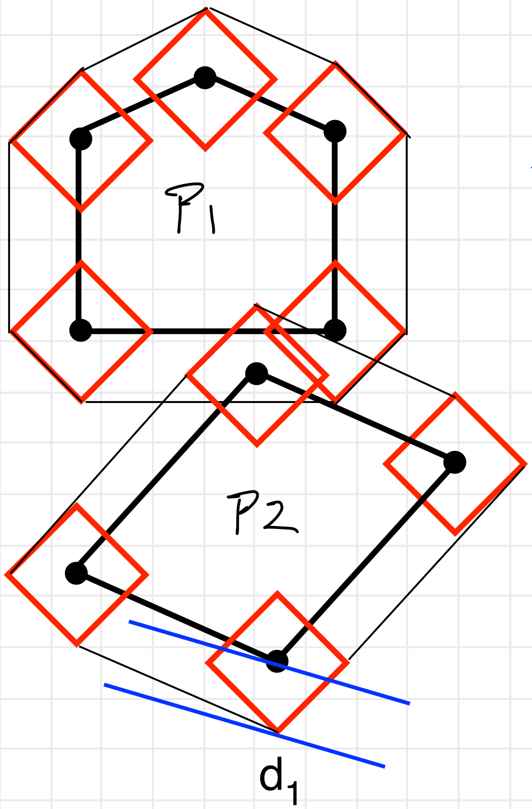
Note: need a more careful definition in case of shared boundary segments

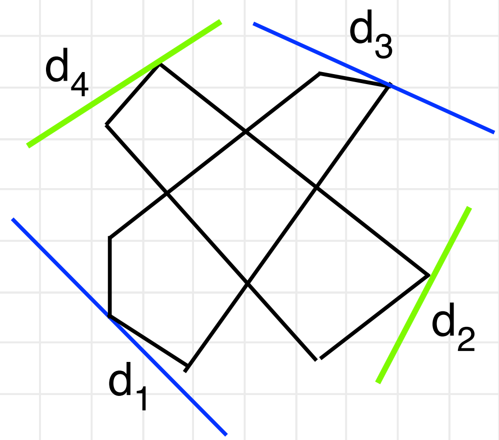
Theorem 1: Let P1 and P2 be disjoint convex polygons, and let R be convex. Then P1 $\oplus$ R and P2 $\oplus$ R form pseudodiscs.

Theorem 2: If Q1. . . Qk are pairwise pseudodiscs of total size n then their union has size O(n).

Theorem 1: Let P1 and P2 be disjoint convex polygons, and let R be convex.
Then P1 ⊕ R and P2 ⊕ R form pseudodiscs.

Proof.

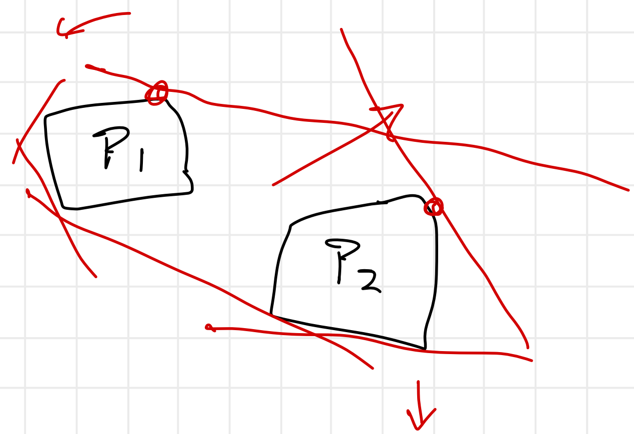




Suppose P1 ⊕ R and P2 ⊕ R are not pseudodiscs.
Then around the union we can find tangent lines d_1, d_2, d_3, d_4, at extreme points alternating between the two.

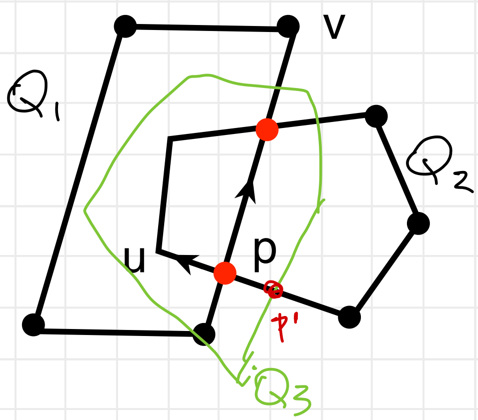
These extreme points correspond to extreme points of P1 and P2, respectively. (by the Stronger Claim.)

But that's impossible for two disjoint convex polygons.



Theorem 2: If Q1. . . Qk are pairwise pseudodiscs of total size n then their union has size O(n).

Proof.



Vertices of the union are of two types:

1. vertices of the Qi's ●

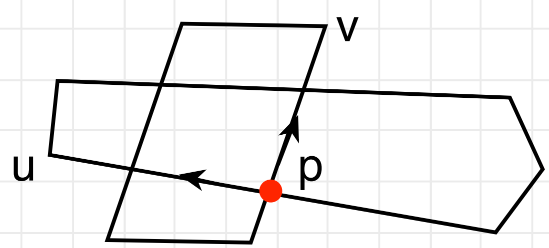
2. intersections of edges ●

We just need to bound the number of type 2 vertices.

From a type 2 vertex p, follow the two edges into the interior to endpoints u, v.

Claim. At least one of u, v is interior to the union.

otherwise



then they are not pseudodiscs!

Charge p to that vertex.
Observe: Each interior vertex is charged at most 2 times (from its two edges).
Thus there are at most 2n vertices of type 2.

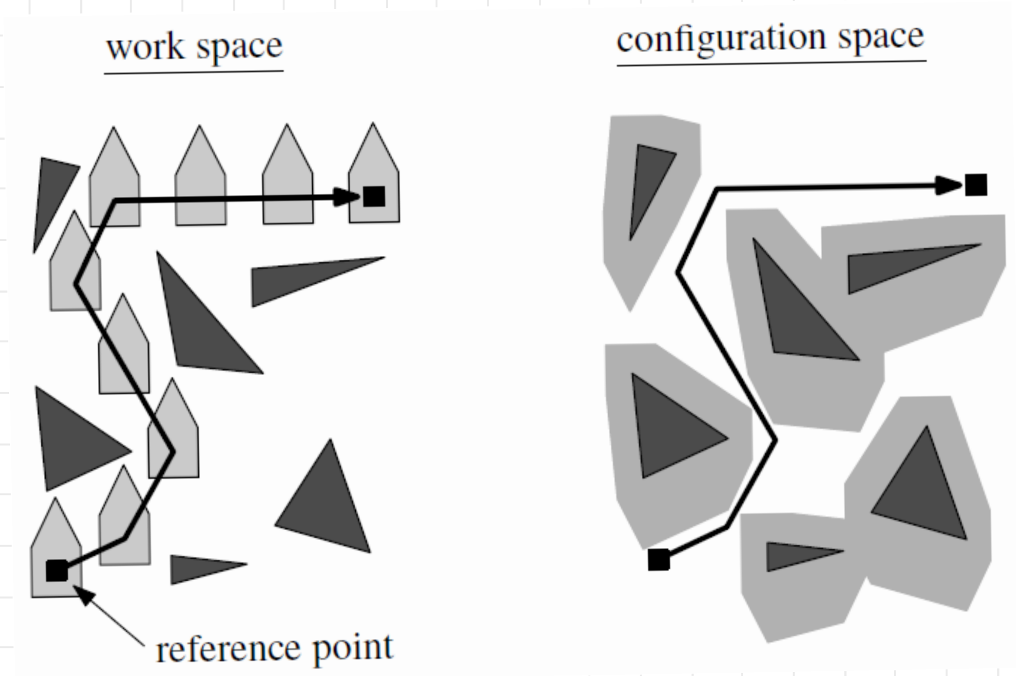
why?

Aha: then p is not on the union (boundary).

How do we show the green Qi doesn't charge u twice from same edge?

Recall

Can polygon R move (via translations) from initial to final position among polygonal obstacles?



[CGAA]

High level idea

1. compute the Minkowski sum P ⊕ (–R) for each obstacle P
2. take the union, to obtain new polygonal obstacles

   we didn't give details on computing union.

3. test if a point (the reference point) can move from initial to final position among the new enlarged obstacles

How to deal with non-convex obstacles

Cut them into triangles. (We assume R is convex.)

$P \oplus R = \text{Union}\{ T_P \oplus R : T_P \text{ a triangle of } P \}$

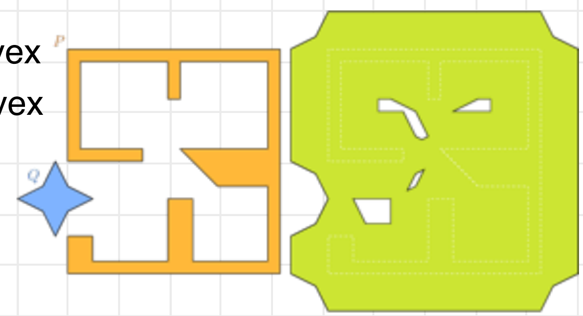
$T_P \oplus R$ is a convex Minkowski sum, and we know how to take their union.

Examples of more complicated Minkowski sums:

P non-convex
R convex

Figure 83: Minkowski sum of O(nm) complexity.

P non-convex
R non-convex



Completing the plan.

Suppose obstacles have total size n and the robot is convex of fixed size.

Forbidden space = union of enlarged convex polygons = $\bigcup \{ P \oplus (-R)$: P an obstacle $\}$
Free space = complement of forbidden space.

Forbidden space has size O(n) by Theorems 1 and 2.

FACT: Forbidden space can be computed in O(n log n) time (complicated),
or via a simpler O(n $\log^2$ n) time divide and conquer algorithm.
Details in [CGAA].

Then the problem is reduced to finding a path for a point in a polygonal region of size O(n).

Translational motion planning in higher dimensions

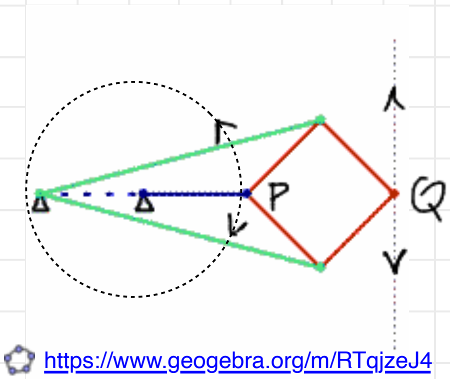
in 2D $O(n \log n)$ by above method (for convex robot of fixed size)

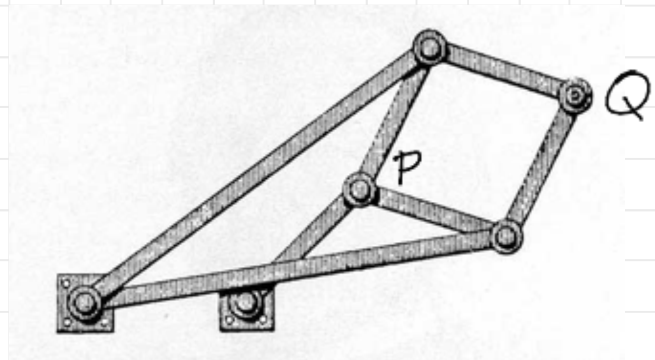
in 3D $O(n^2 \log^2 n)$ by similar method (for convex robot of fixed size)
(Note that this finds a path, not necessarily a shortest path.)

General road map algorithm of Canny $O(n^d \log n)$ where d is the number of degrees of freedom — this applies to rotational motion as well.

Robot Arm Motion (Linkages)

The study of linkages is old, e.g. Peaucellier linkage to convert rotary motion to linear motion





as P moves on a circle, Q moves on a line

We will just look at a chain (not a general graph, which gets into "rigidity theory"). Input is a polygonal chain where the segments ("links") have fixed lengths and the angles between successive links may change.

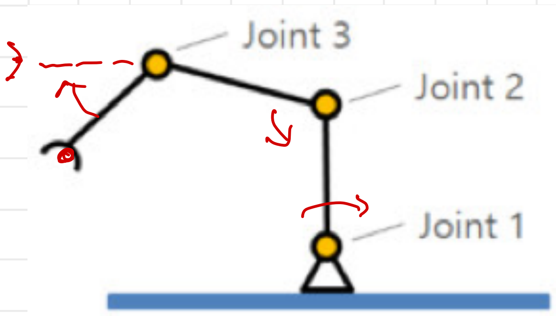
Two models:

- intersection of links allowed, e.g. above, where linkage is essentially planar, but each link is slightly higher (in 3rd dimension) than previous

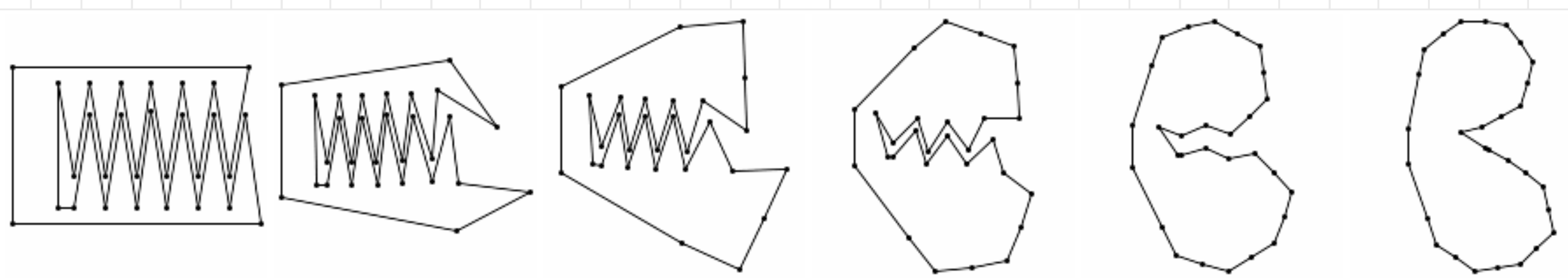
- intersections forbidden, e.g. protein folding, robot arm in 3D

We will study two problems:

1. Given a chain with one endpoint fixed, where can the other endpoint reach?
   Allow intersections.



2. Given a chain, can we go from any configuration to any other?
   Forbid intersections.

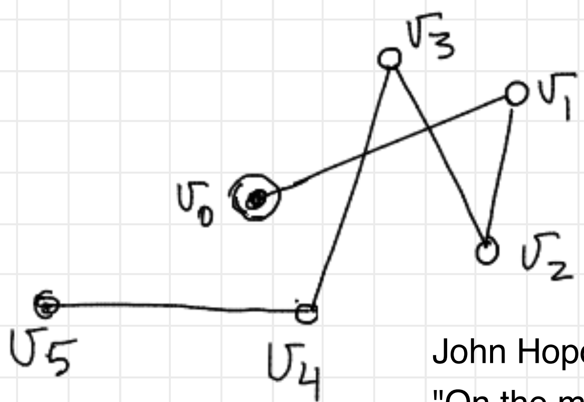


different views not to scale

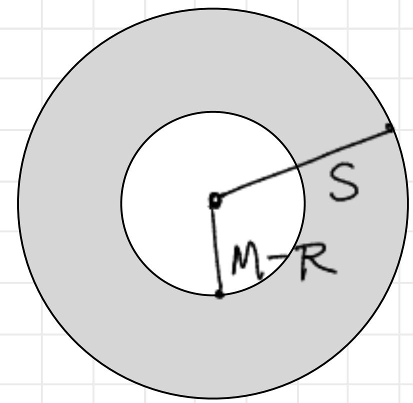
Theorem. Given a chain $v_0, \ldots, v_n$ with link lenghts $L_1, \ldots, L_n$ and with v_0 pinned in the plane, the reachability region of v_n is an annulus with

outer radius = $S = \sum L_i$

inner radius = $\begin{cases} M - R, \text{ where } M = \max L_i,\ R = S - M \\ 0 \text{ if } R > M \end{cases}$



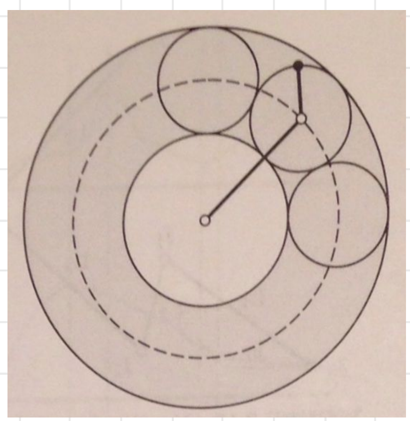
S = sum
M = max, R = the rest

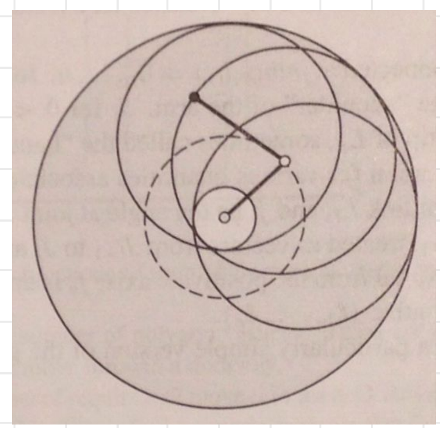


John Hopcroft, Deborah Joseph, and Sue Whitesides.
"On the movement of robot arms in 2-dimensional bounded regions." 1985

Idea of proof

- trivial for n = 1
- for n = 2





Theorem. Given a chain $v_0, \ldots, v_n$ with link lengths $L_1, \ldots, L_n$ and with v_0 pinned in the plane, the reachability region of v_n is an annulus with

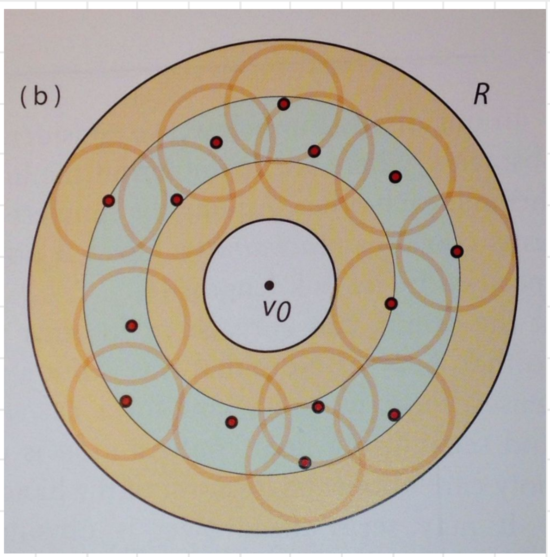
outer radius = $S = \sum L_i$

inner radius = $\begin{cases} M - R, \text{ where } M = \max L_i,\ R = S - M \\ 0 \text{ if } R > M \end{cases}$

S = sum
M = max, R = the rest

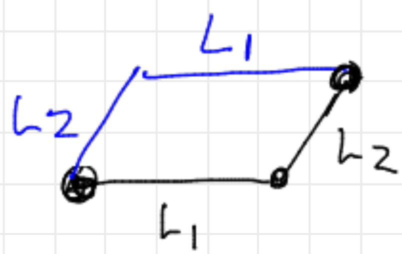
Idea of proof

General case by induction on n.
The first n-1 links yield an annulus. Adding the last link, gives the Minkowski sum of the annulus and a disc — which is an annulus.

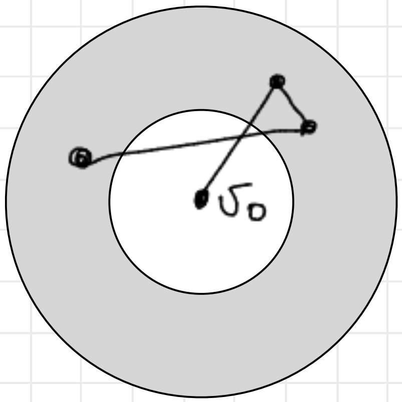


Devadoss and O'Rourke

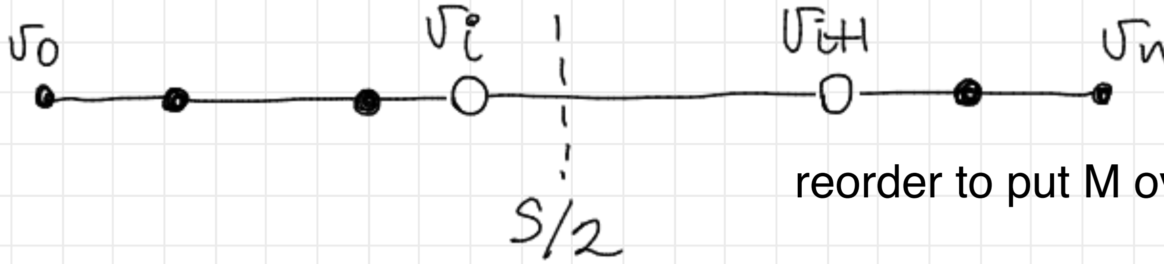
The formulas for outer and inner radius are clear if L_1 is the longest link, but note that the order does not matter!



The theorem tells us WHICH points can be reached.
It is also possible to find HOW to reach any point in the annulus.



In fact, it is possible to reach any point using only two of the joints and locking the others (just ensure that S, M, R are the same)



reorder to put M over the midpoint

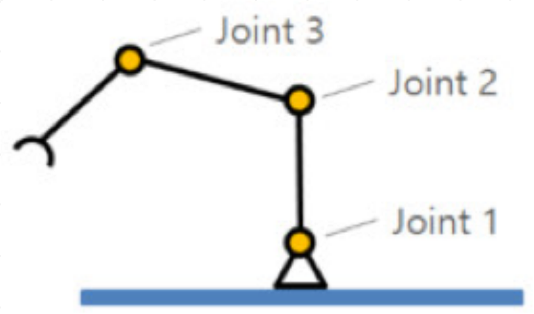
So it suffices to find out how to reach a point with 3 links.

(Details omitted)

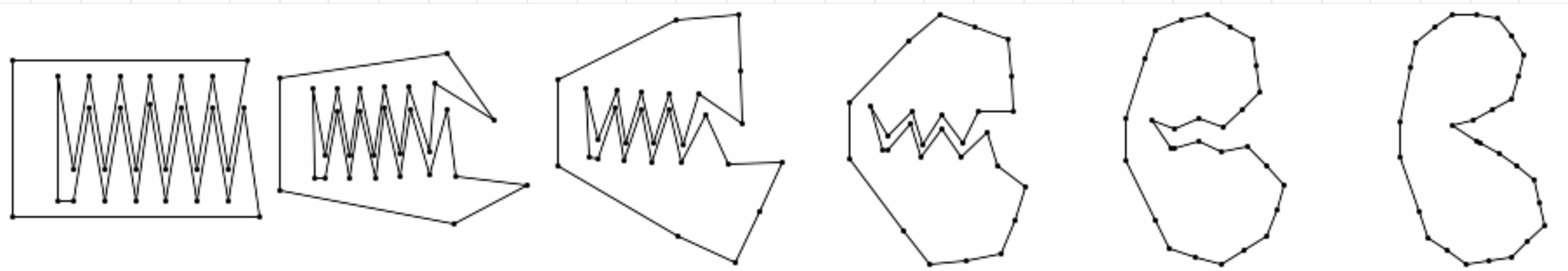
Recall

We will study two problems:

1. Given a chain with one endpoint fixed, where can the other endpoint reach?
   Allow intersections.

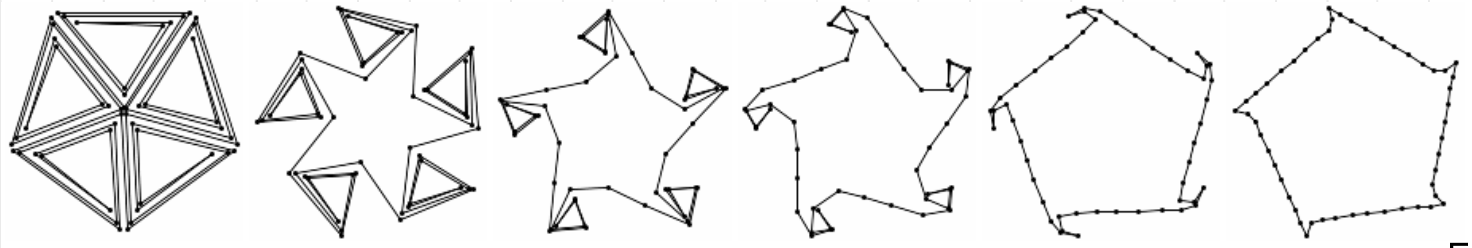


2. Given a chain, can we go from any configuration to any other?
   Forbid intersections.



different views not to scale

2. Given a linkage, can we go from any configuration to any other?
Forbid intersections.



not to scale

Erik Demaine

In 3D, the answer is “not always”.

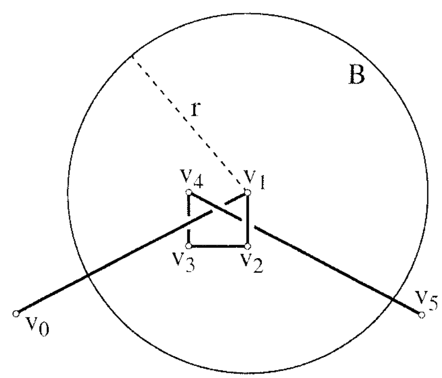


Fig. 1. A locked, open chain K with long “knitting needles” at the ends.

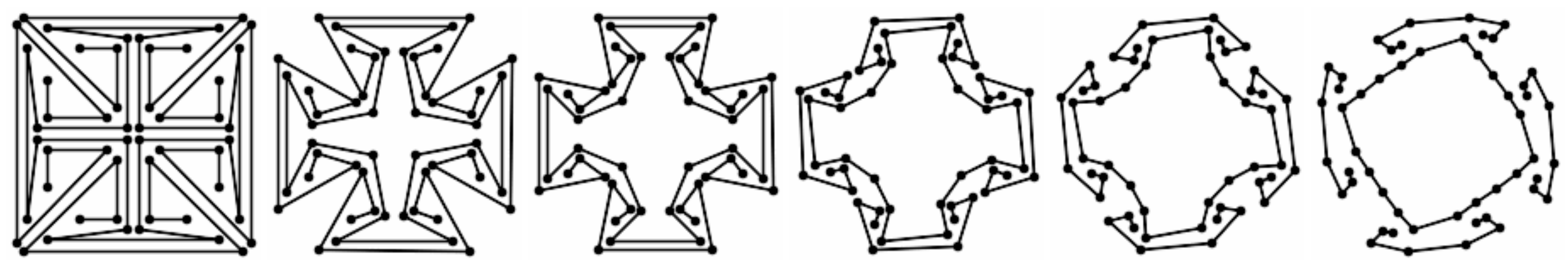
Biedl, T., Demaine, E., Demaine, M., Lazard, S., Lubiw, A., O'Rourke, J., Overmars, M., Robbins, S., Streinu, I., Toussaint, G. and Whitesides, S., Locked and unlocked polygonal chains in three dimensions. 2001

OPEN. Can a chain of unit length links be locked? relevant for protein chains.

OPEN. Find a polynomial time algorithm to test if a 3D polygonal chain is locked.
(It is PSPACE-hard to test if we can get from one configuration to another.)

Theorem. In 2D, any chain can be straightened. Any closed chain can be made convex.

Robert Connelly, Erik D. Demaine, and Günter Rote.
"Straightening Polygonal Arcs and Convexifying Polygonal Cycles." 2003
(Erik Demain's PhD thesis work)



This implies that a linkage can go from any configuration to any other.

initial config. $\longrightarrow$ straight config. $\longleftarrow$ final config.
$\longrightarrow$

idea of proof: they show that it suffices to use *expansive motions* — the distance between any two vertices never decreases.

A better way to go from initial to final configuration — avoid going through intermediate straight/convex chain.

Hayley Iben, James F. O'Brien, and Erik D. Demaine.
"Refolding planar polygons." 2009.

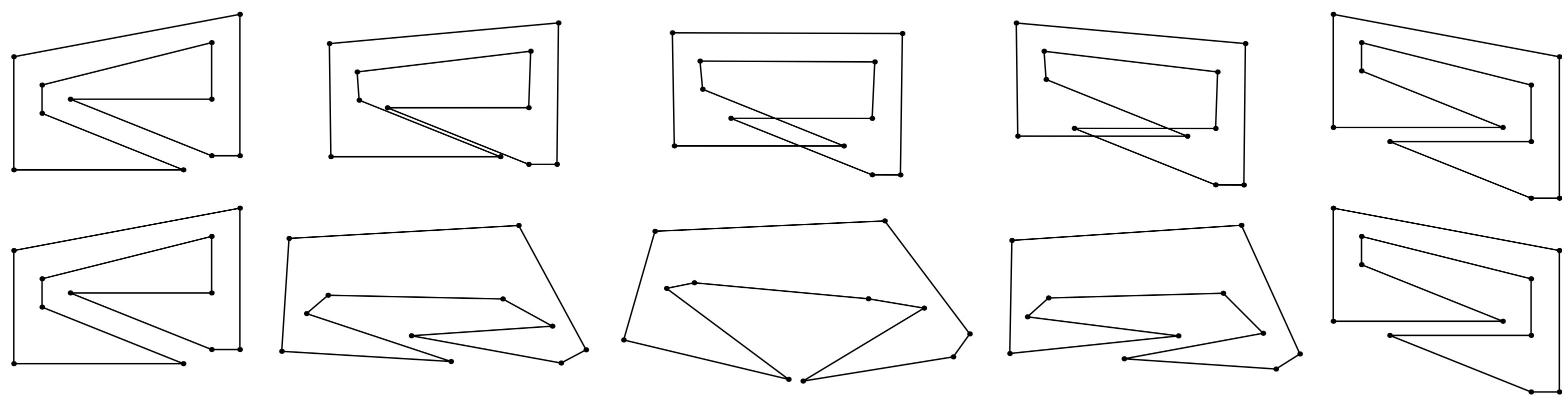


Fig. 2 The *top row* demonstrates how using the vertex-position metric alone will, as expected, generate a sequence with self intersections. The *bottom row* illustrates how the collision-avoidance machinery alters the vertex motions to avoid self intersection. Computation times were less than one second

Summary

- Motion planning
    - convex robot translating among 2D obstacles
    - linkages

References

- [CGAA] Chapter 13
- [Zurich notes] Appendix D