Computing Singularity-Free Paths on the Configuration Manifold for Closed-Chain Manipulators

Michael E. Henderson
IBM T.J. Watson Research Center

O. Bohigas, L. Ros, M. Manubens, J. Porta

Institut de Robotica i Informatica Industrial
CSIC-UPC

Use local approximations of surface to solve a global problem

Continuation: Cover a surface

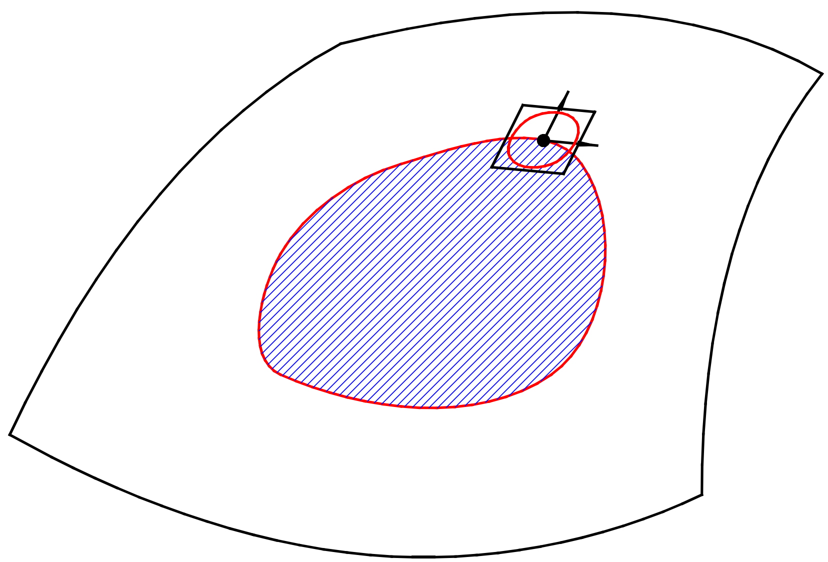
Optimization: Minimize a functional on a surface

Surface is feasible set (trust region methods)

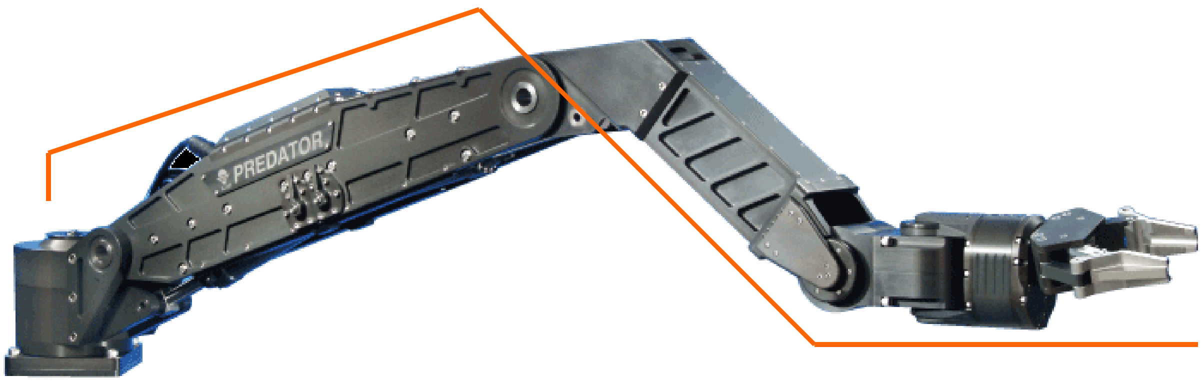
Path planning: Find the shortest path between two points on a surface.

Surface is configuration space.

All can be posed as extending a “known” region from the boundary.

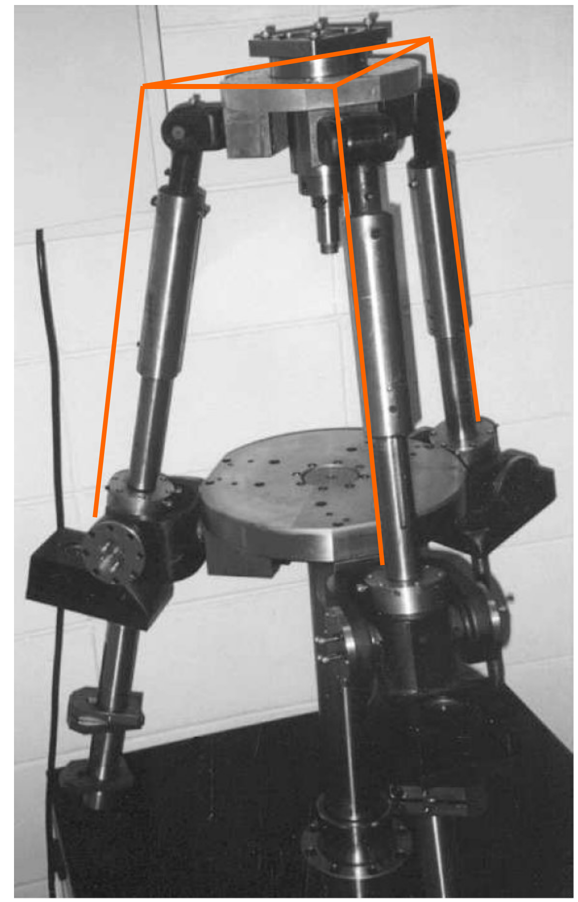


Chain manipulator.



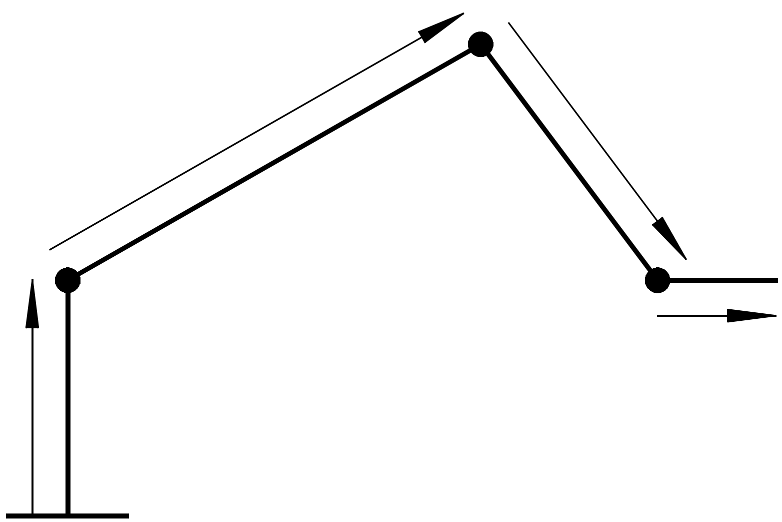
http://krafttelerobotics.com/

Closed Chain manipulator.



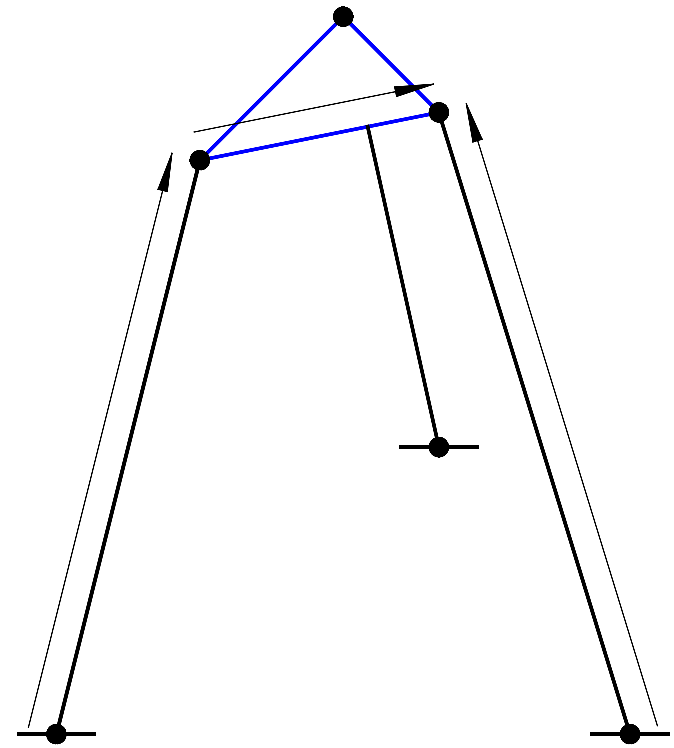
From Zlatanov, Bonev, Gosselin 2002

Open Chain manipulator.



Graph is a tree

Closed Chain manipulator.



Graph has cycles

Configuration Space

$$\Phi(\mathbf{q}_{In}, \mathbf{q}_{Out})=0$$

$$g(\mathbf{q}_{In}, \mathbf{q}_{Out})>0$$

If **x**'s are the position of the joints, $\Phi=0$ has algebraic constraints

$$(x_i - x_j) \cdot (x_i - x_j) = L_{ij}^2$$

Joint restrictions are not algebraic (rotations). $x_j = x_i + L_{ij} R_j(\theta,\phi)$

Describes all possible positions (configuration space).

Configuration Space

$$\Phi(\mathbf{q}_{In}, \mathbf{q}_{Out}) = 0$$

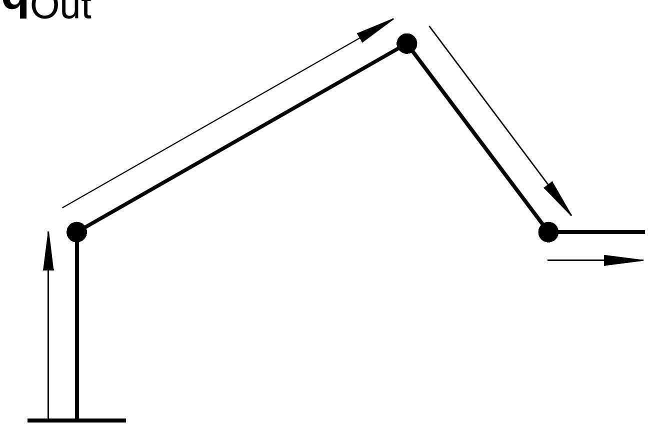
$$g(\mathbf{q}_{In}, \mathbf{q}_{Out}) > 0$$

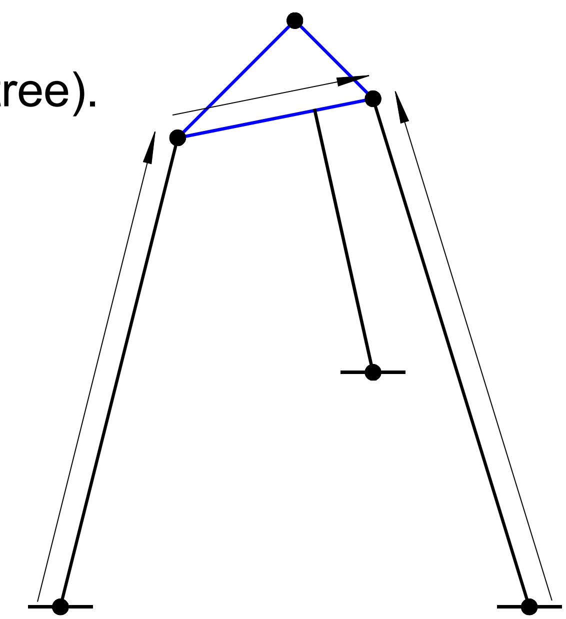
If **x**'s are the position of the joints, Φ=0 has algebraic constraints

$$(x_i - x_j) \cdot (x_i - x_j) = L_{ij}^2$$

If not closed Φ can be put in lower triangular form (walk the tree).

$\mathbf{q}_{In}$ given, find $\mathbf{q}_{Out}$

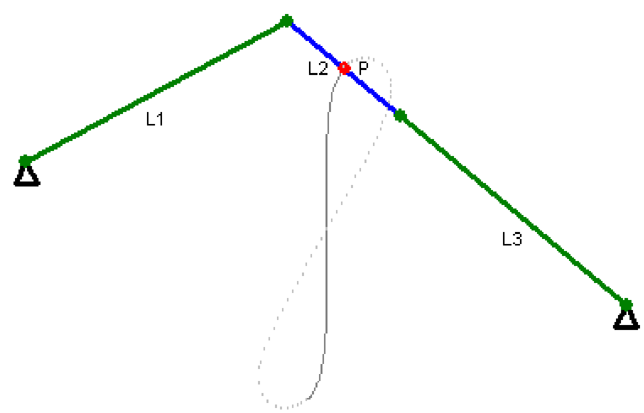




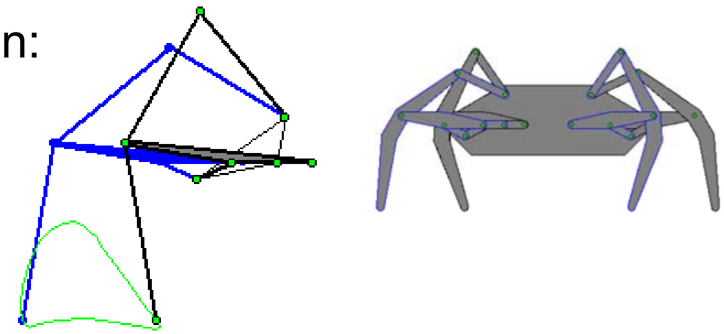
Fairly simple linkages can have complicated configuration spaces

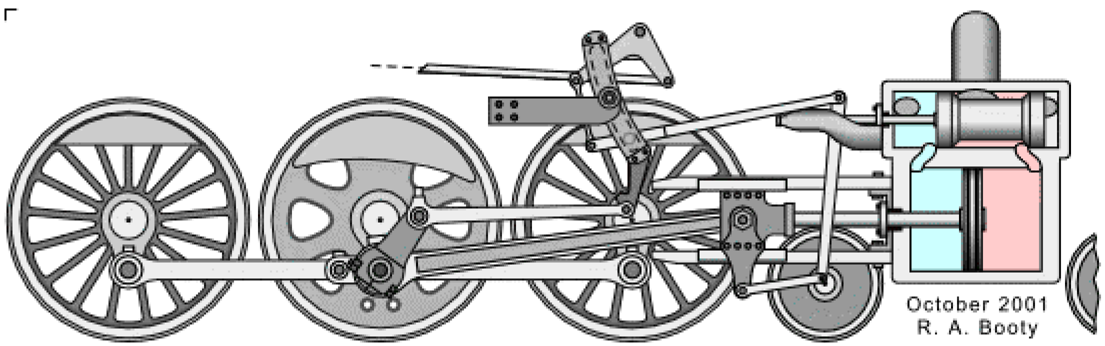
Steam engines/Watt's Linkage

(These are 1 degree of freedom)

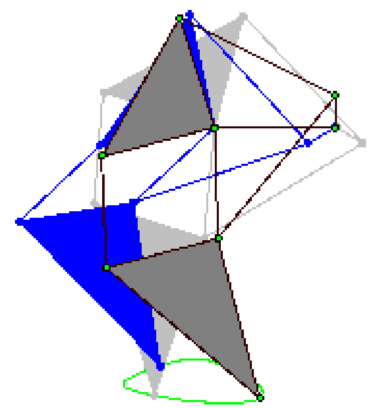


Klann:





Jansen:



http://home.roadrunner.com/~trumpetb/loco/

Strandbeest: Theo Jansen

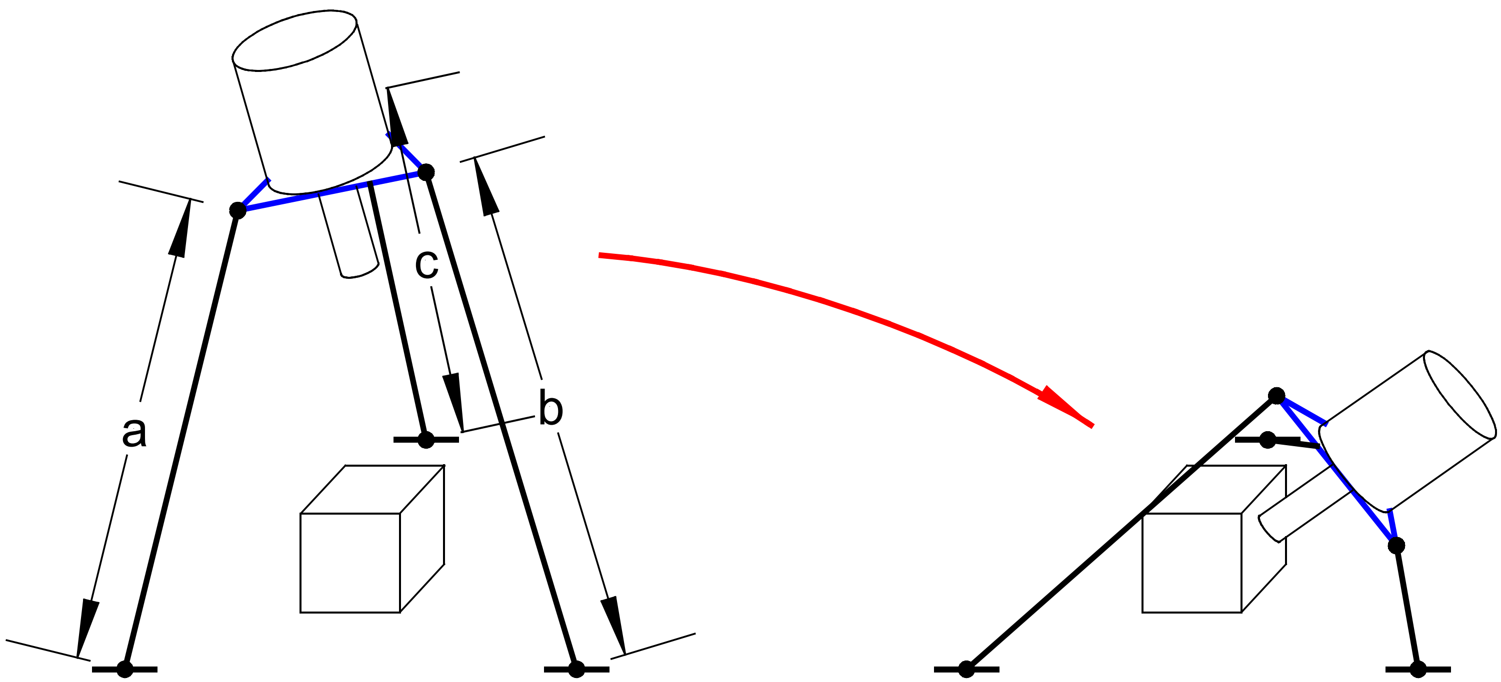
http://shuisman.com/?p=245

Peaucellier–Lipkin linkage

http://kmoddl.library.cornell.edu/model.php?m=reuleaux

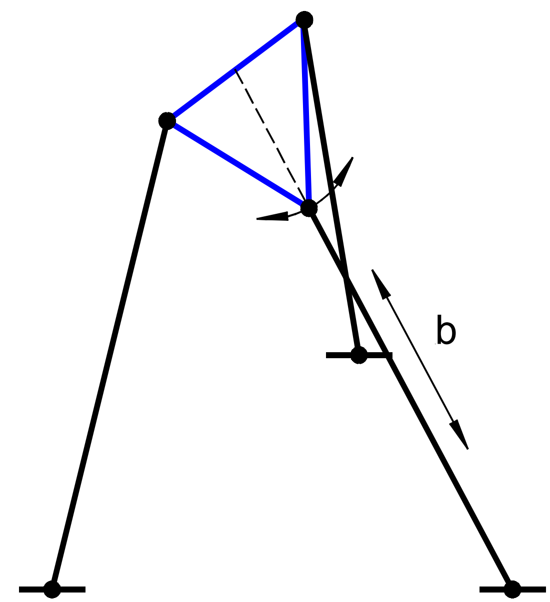
Path Planning

How do you change inputs to move from here to there?



Without “problems”/Singularities

(E.g. when b changes can't control which way the platform moves.)

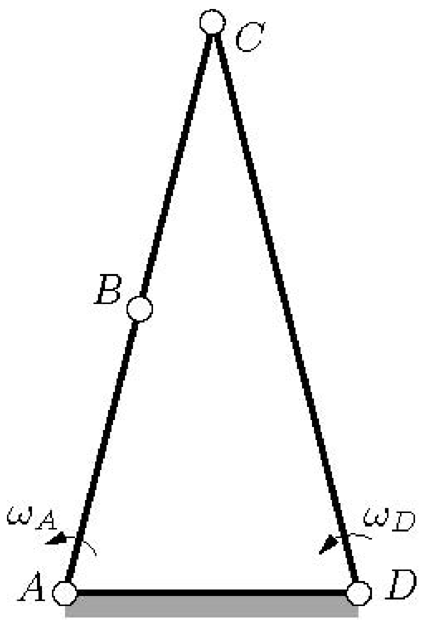


Singular configurations

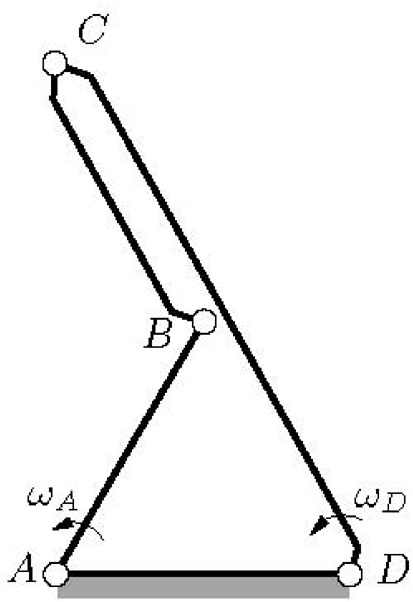
$$\Phi(\mathbf{q}_{In},\mathbf{q}_{Out}) = 0$$

$$\Phi_{In}\Delta\mathbf{q}_{In} + \Phi_{Out}\Delta\mathbf{q}_{\mathbf{Out}} = 0$$

$$\|\Delta\mathbf{q}_{In}\|^2 + \|\Delta\mathbf{q}_{Out}\|^2 = 1$$

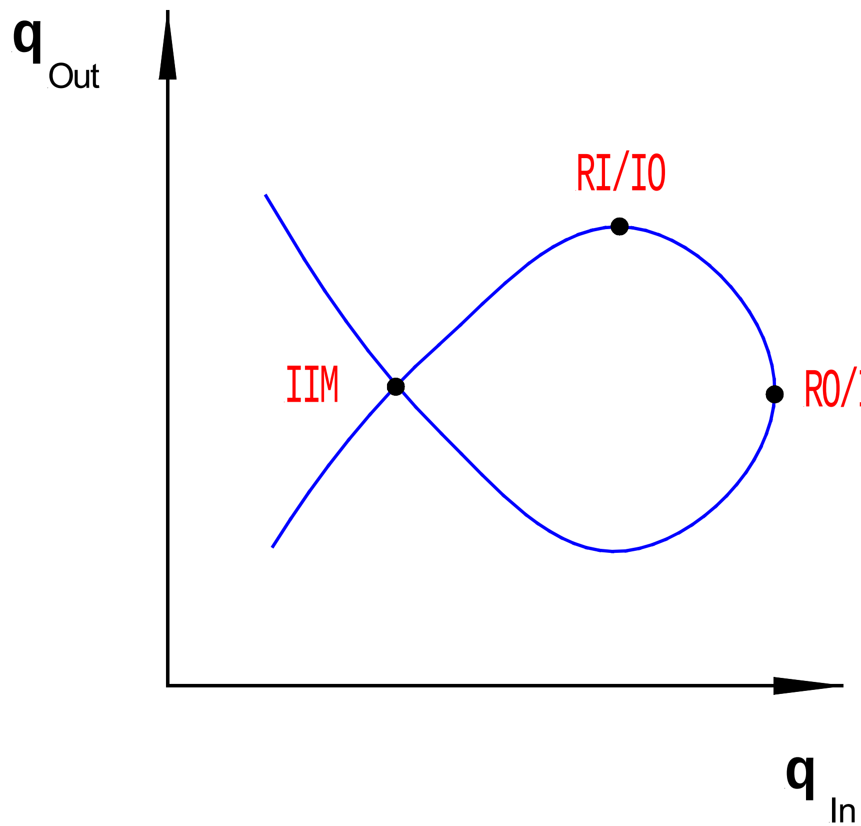


$\Delta\mathbf{q}_{Out} = 0$



$\Delta\mathbf{q}_{In} = 0$

"Stuck"

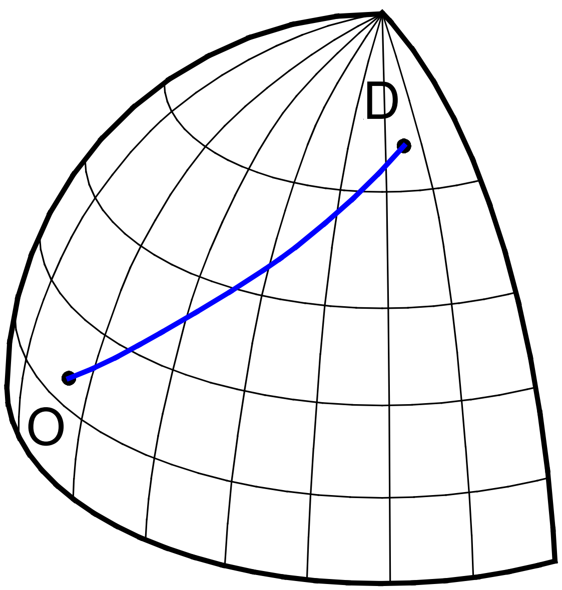


Path Planning:

Given two points in configuration space O and D
find a path through configuration space connecting them.

"Shortest"/"Lowest energy"/"Obstacle avoiding"

Kinematic/Dynamic.



Path Planning:

Given two points in configuration space O and D
find a path through configuration space connecting them

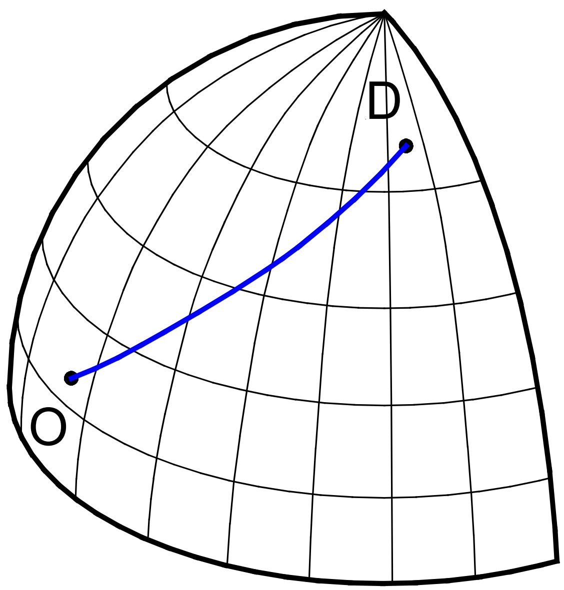
“Shortest”/”Lowest energy”/”Obstacle avoiding”

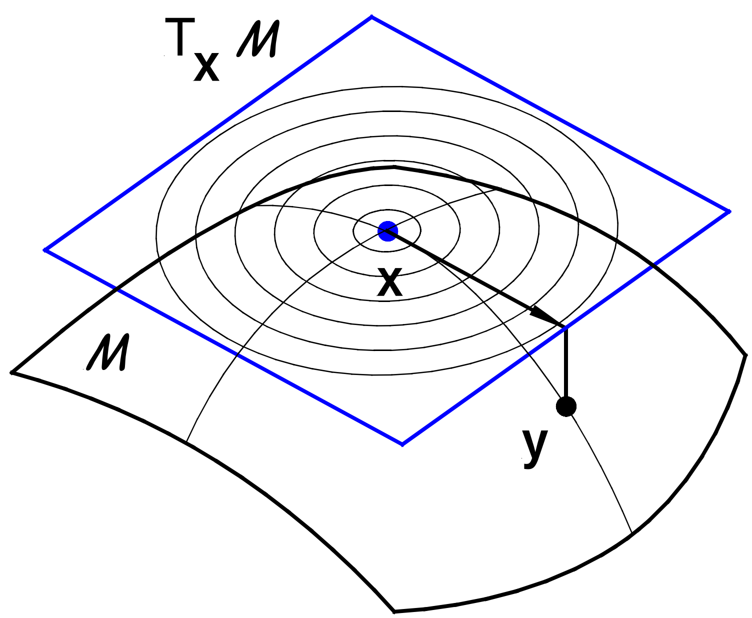
Kinematic/Dynamic.

Distance is local (metric). Length of a path is not.

Path which

minimizes $\int [g].\mathbf{d}s$





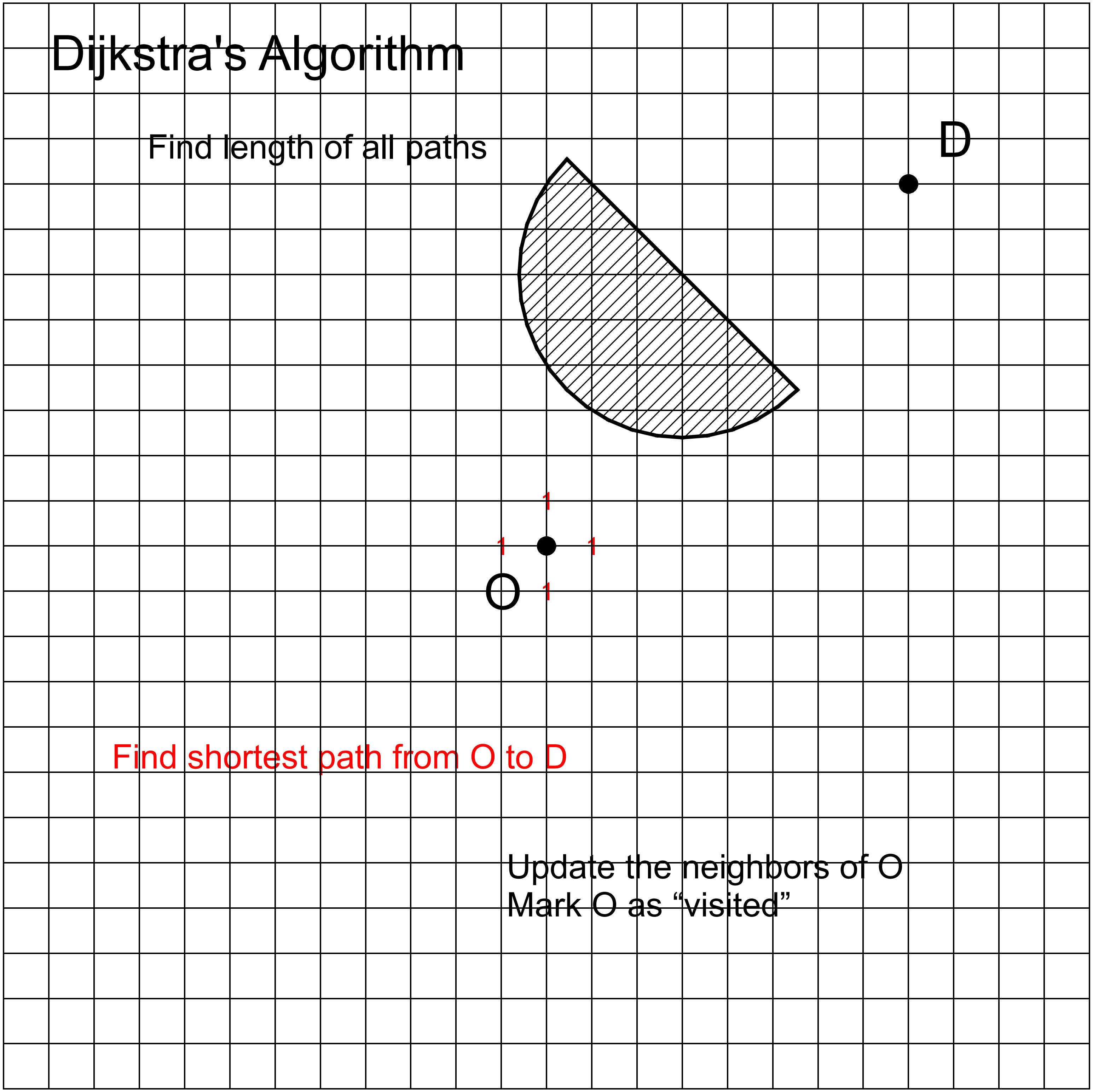
If dist(O,x) is known

dist(O,y) can be found

y near x

Dijkstra's Algorithm

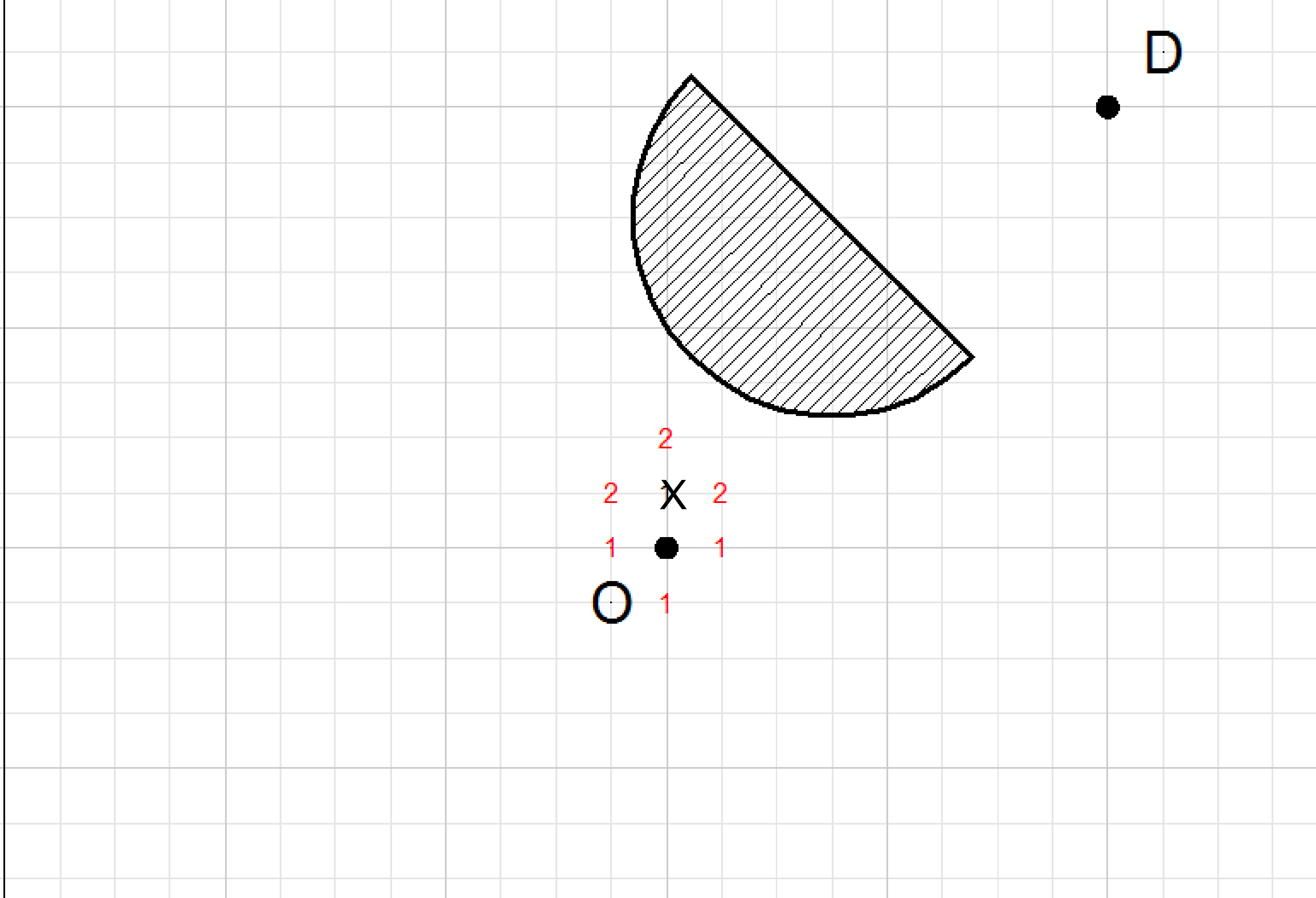
Find length of all paths



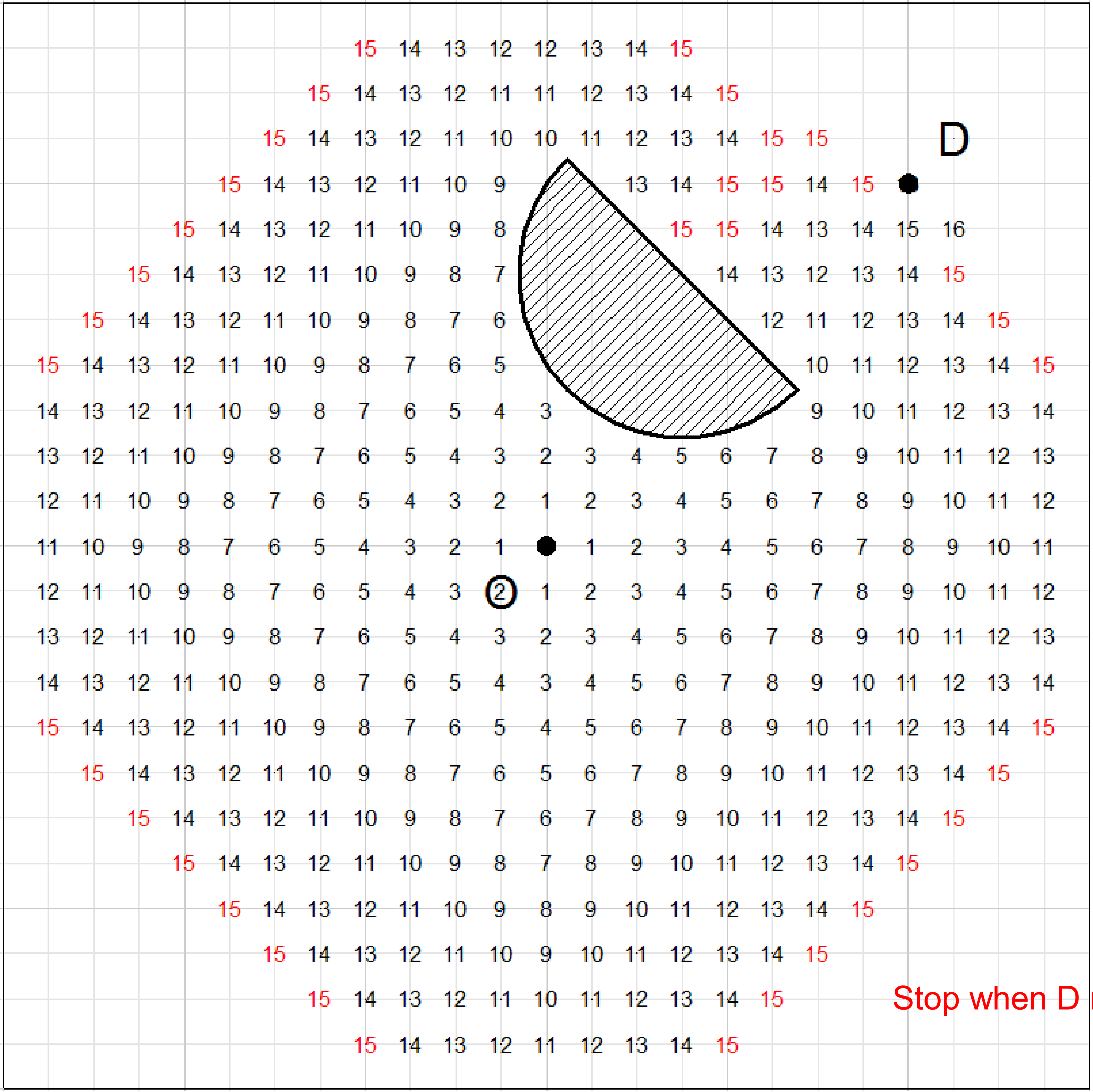
Find shortest path from O to D

Update the neighbors of O
Mark O as “visited”

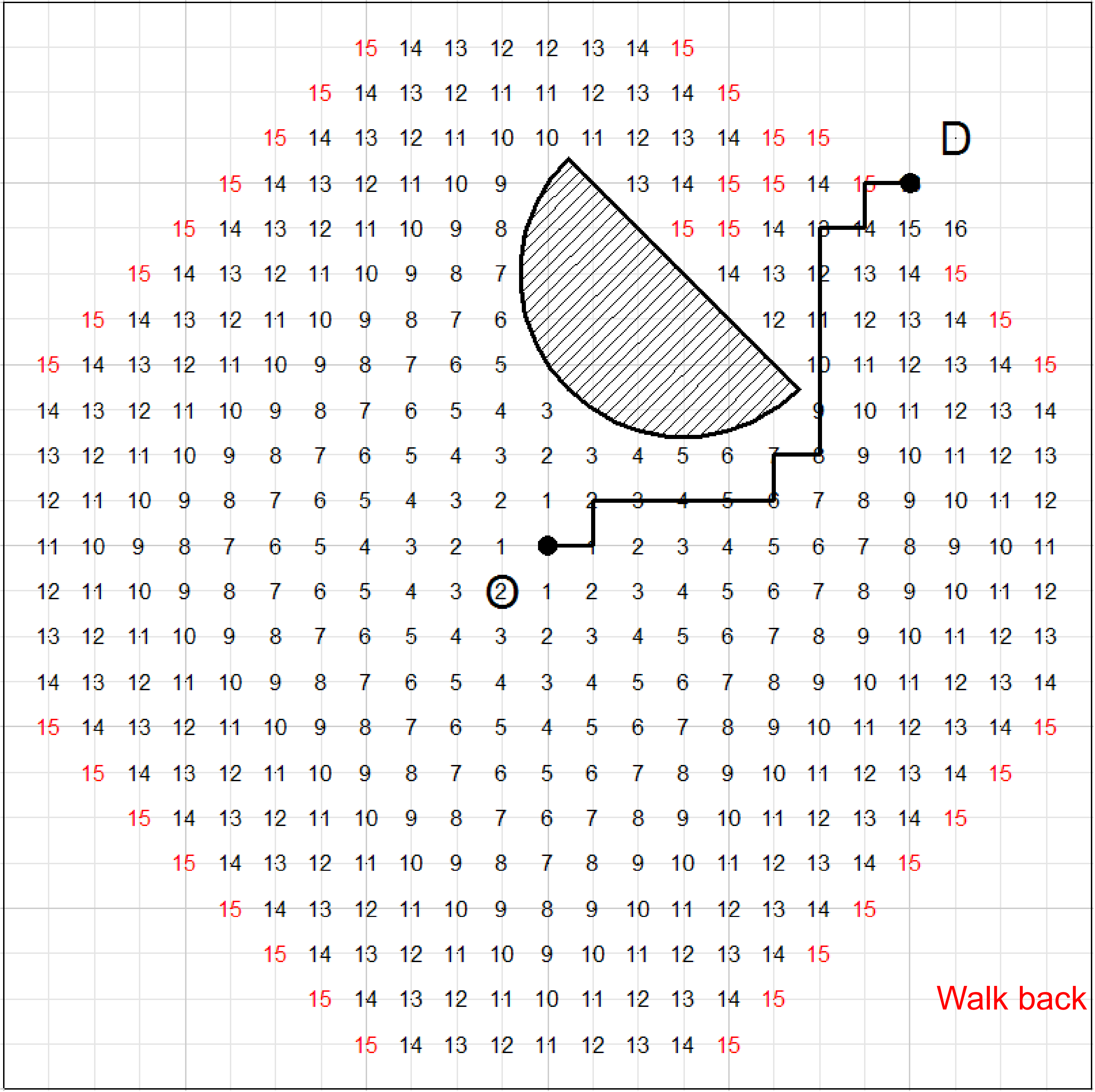
Dijkstra's Algorithm



Pick closest unvisited vertex X
  Update the unvisited neighbors of X
  Mark X as visited

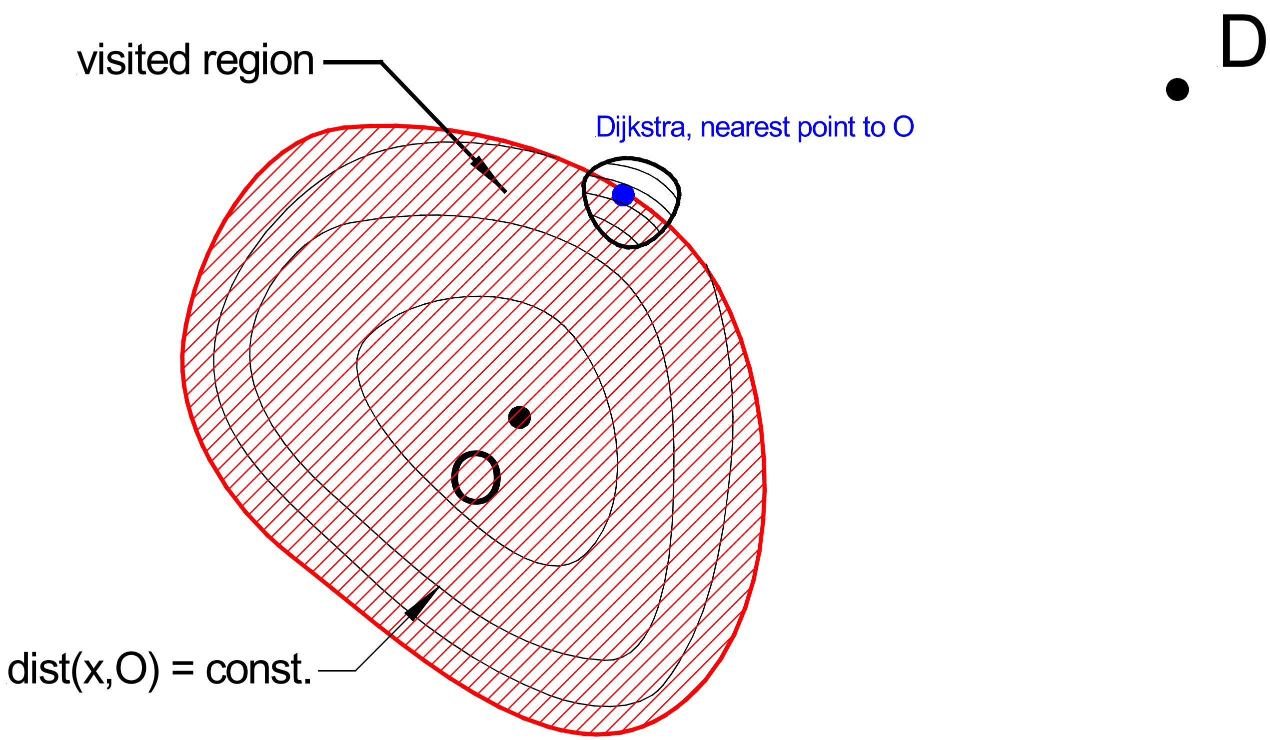


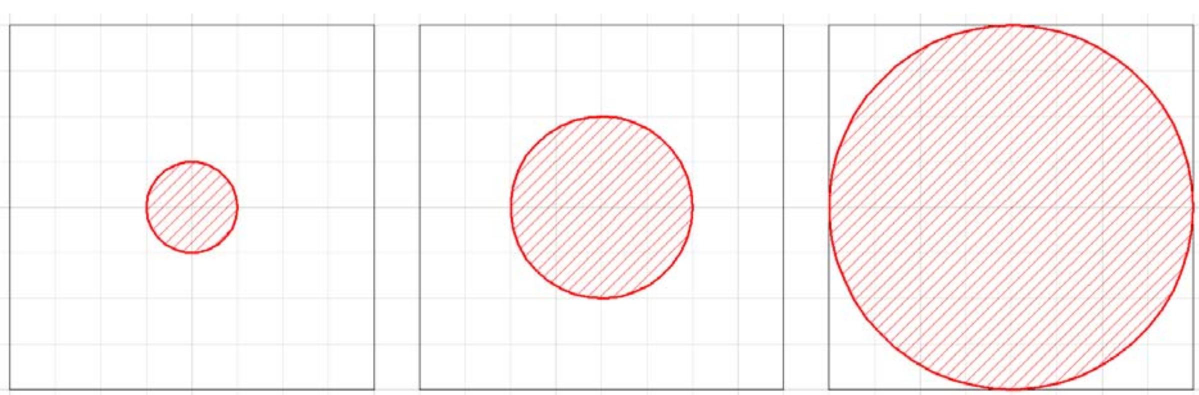
Stop when D reached



Walk back to O

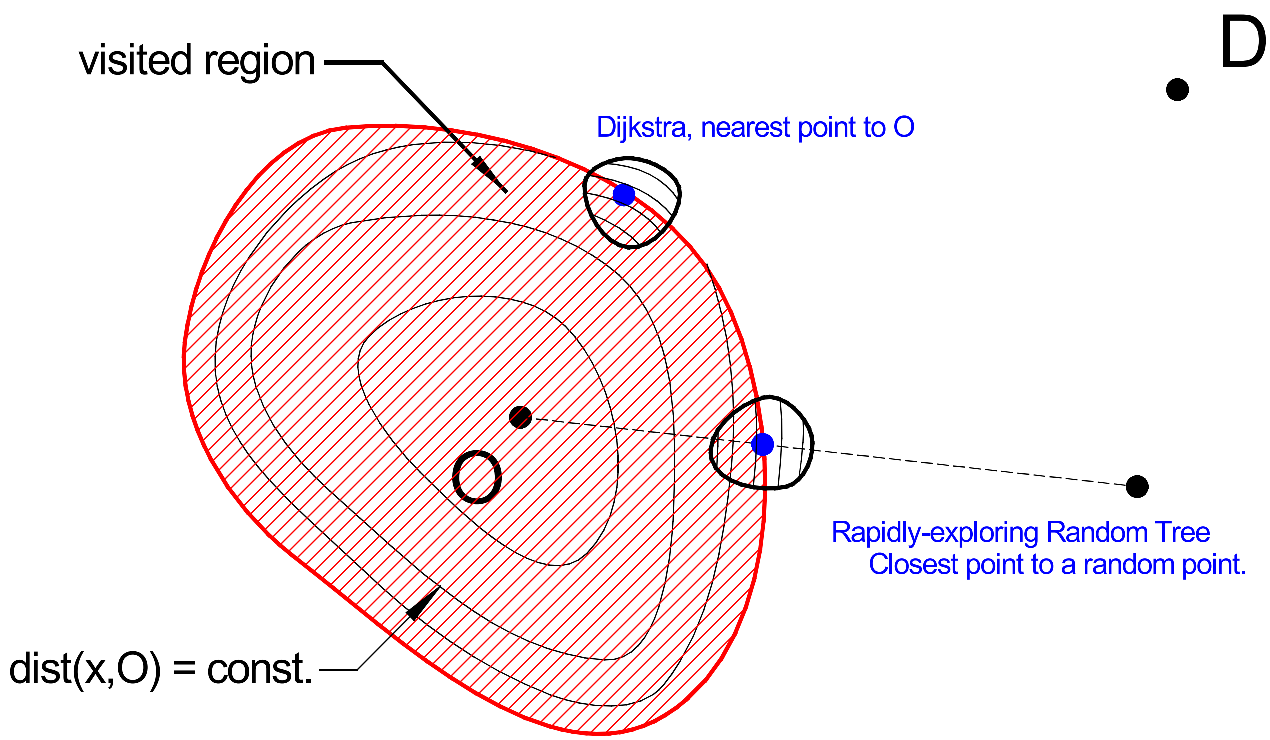
Abstract version

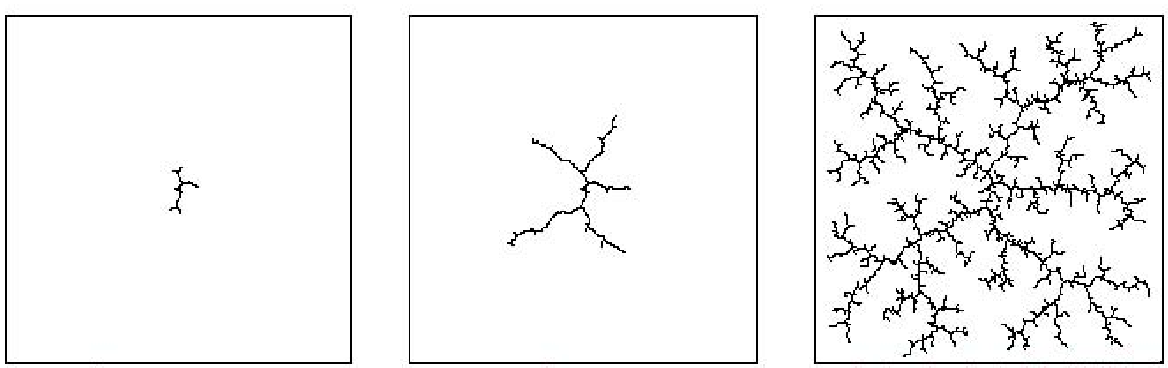




Abstract version

RRT: Rapidly Exploring Random Tree



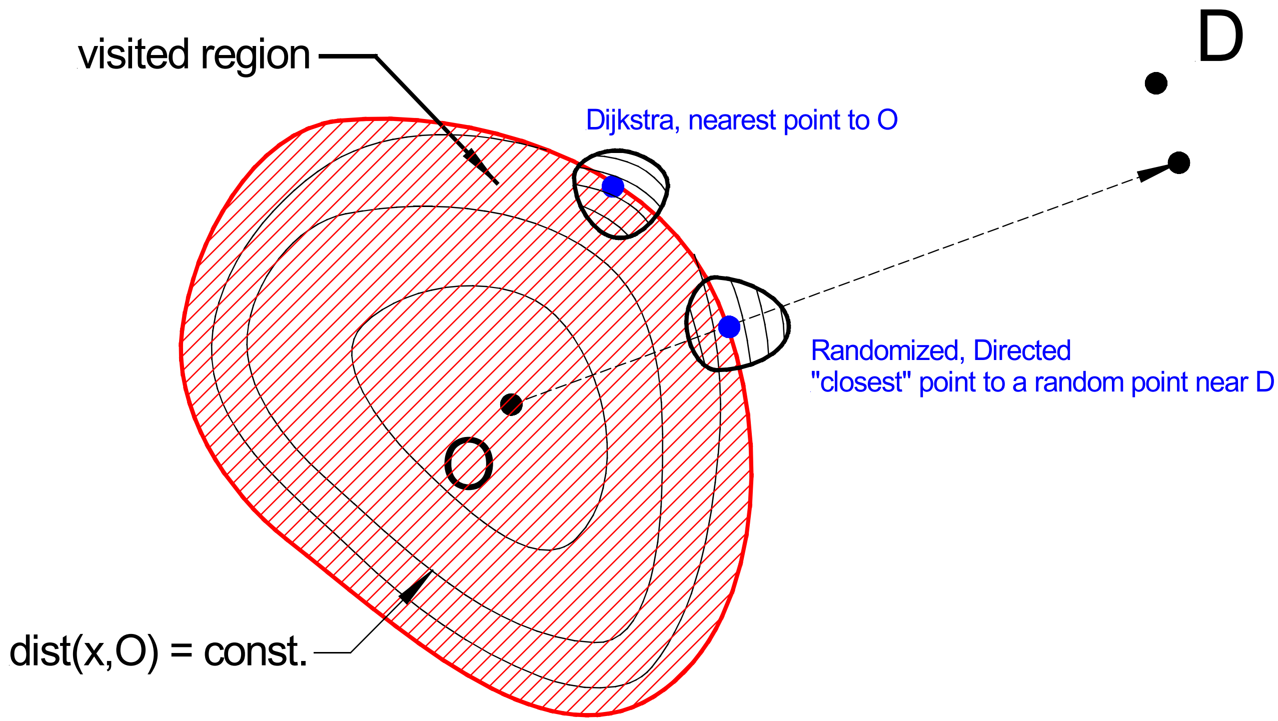


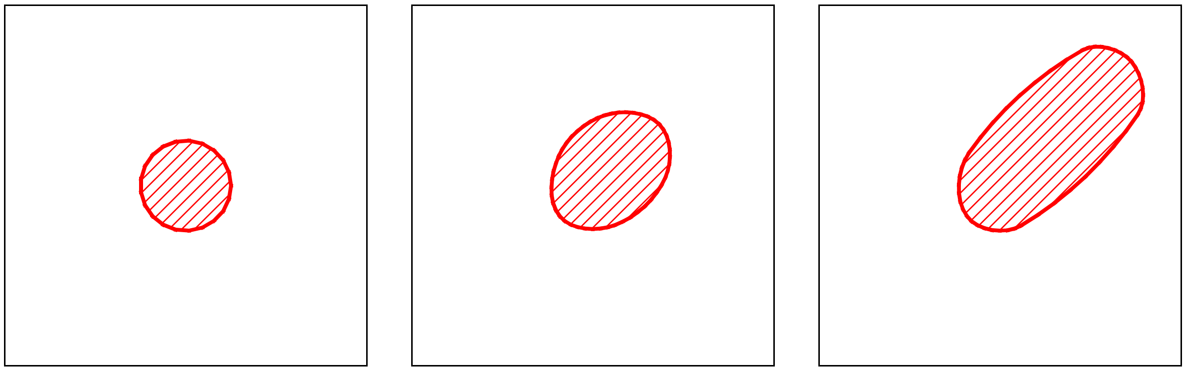
LaSalle

(needs some distance estimate in ambient space)

Abstract version

α-RRT (Randomized, directed)



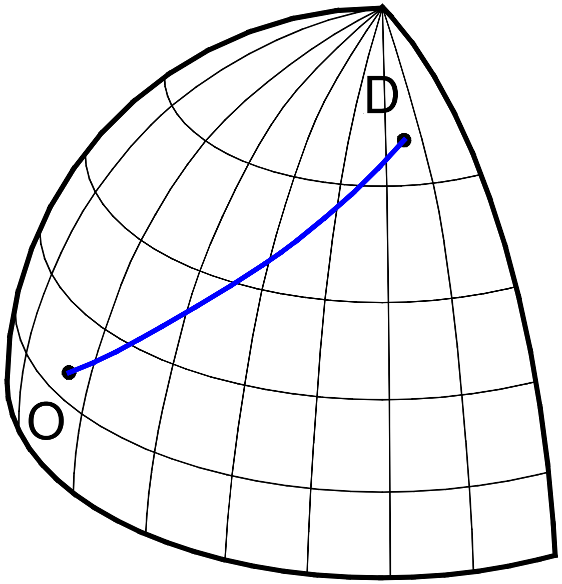


Porta, Jaillet

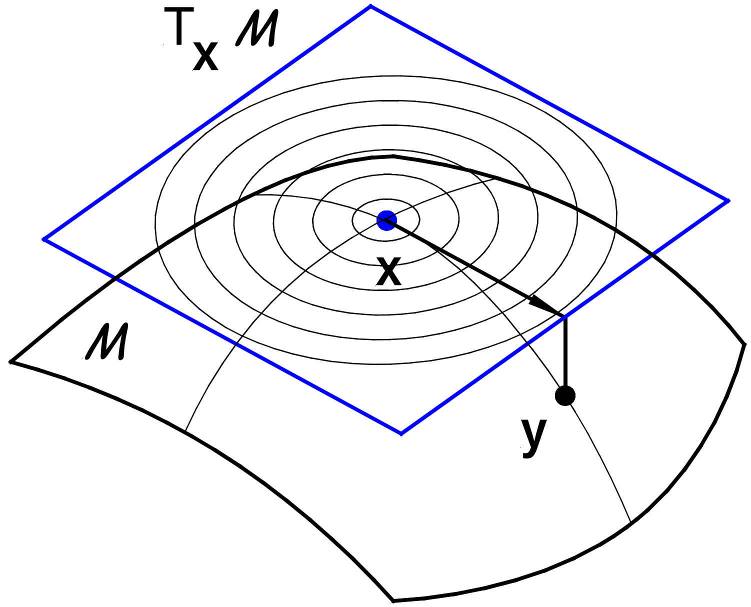
(needs some distance estimate in ambient space)

d(O,*) on configuration space

Find a path on a surface

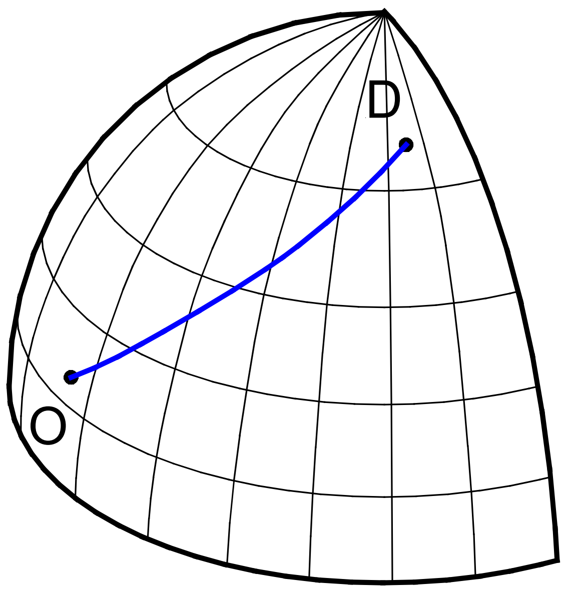


Distance defied locally.

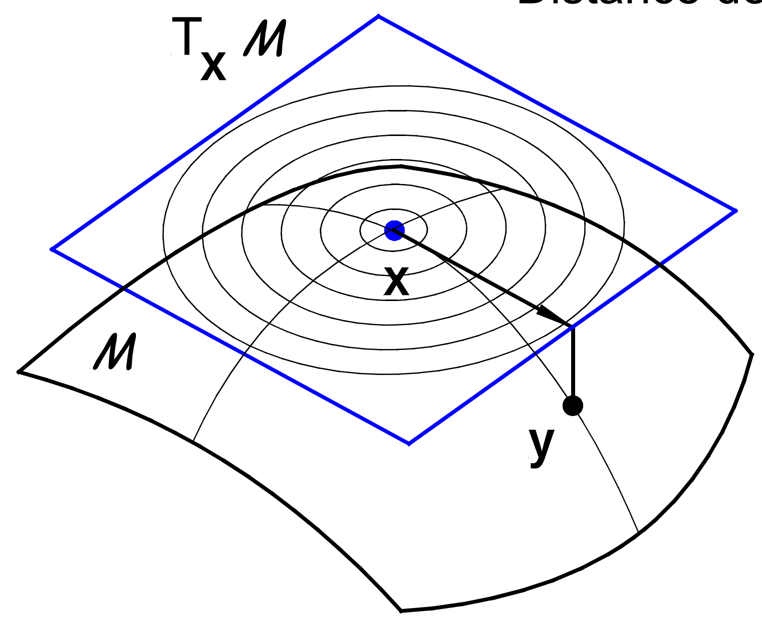


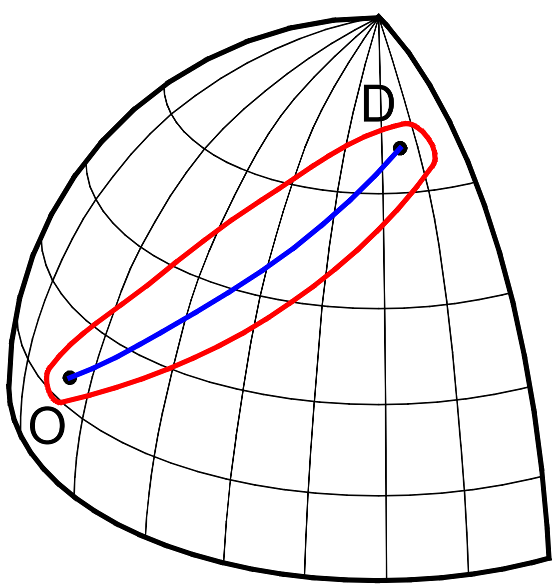
d(O,*) on configuration space

Find a path on a surface

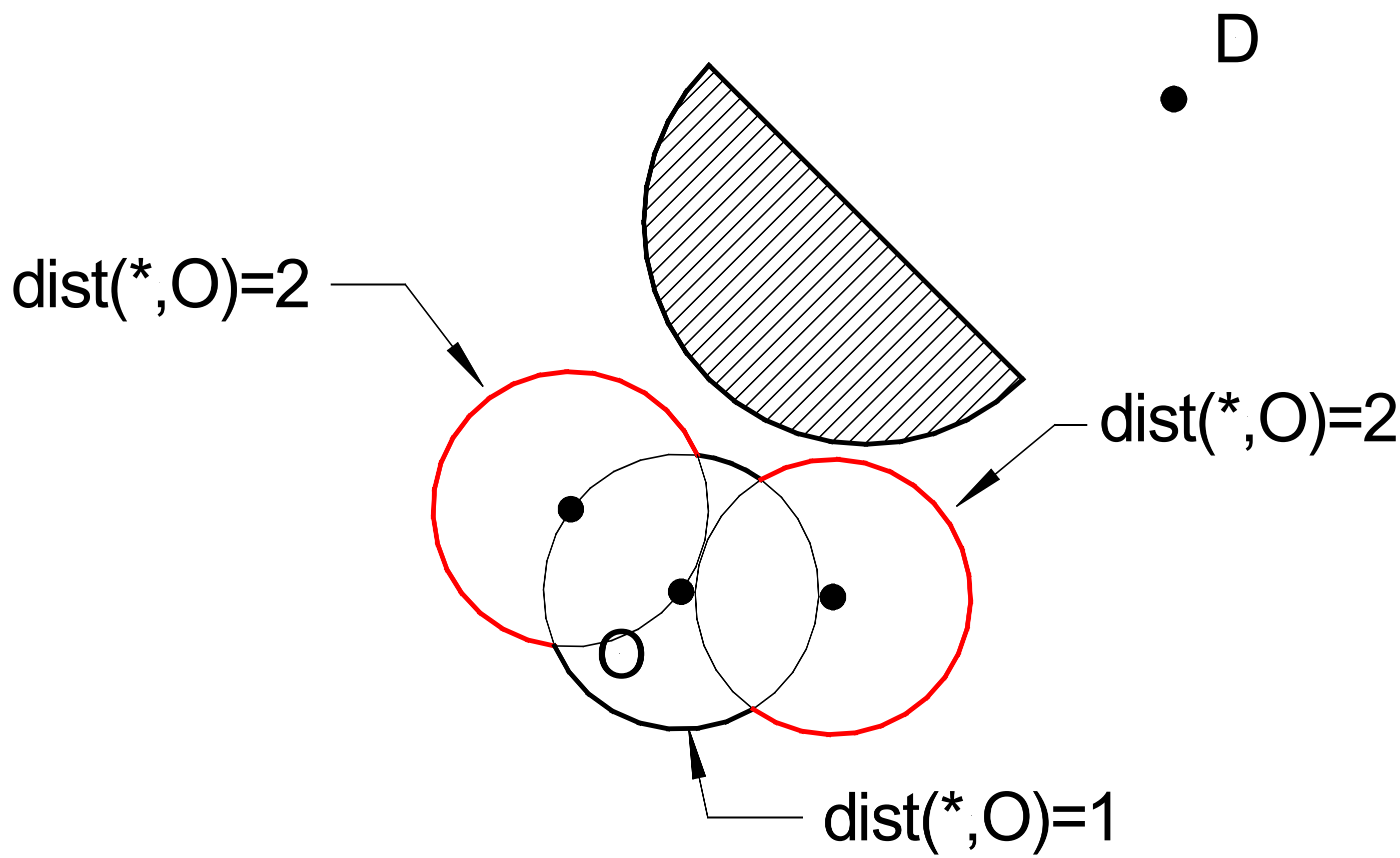


Distance defied locally.



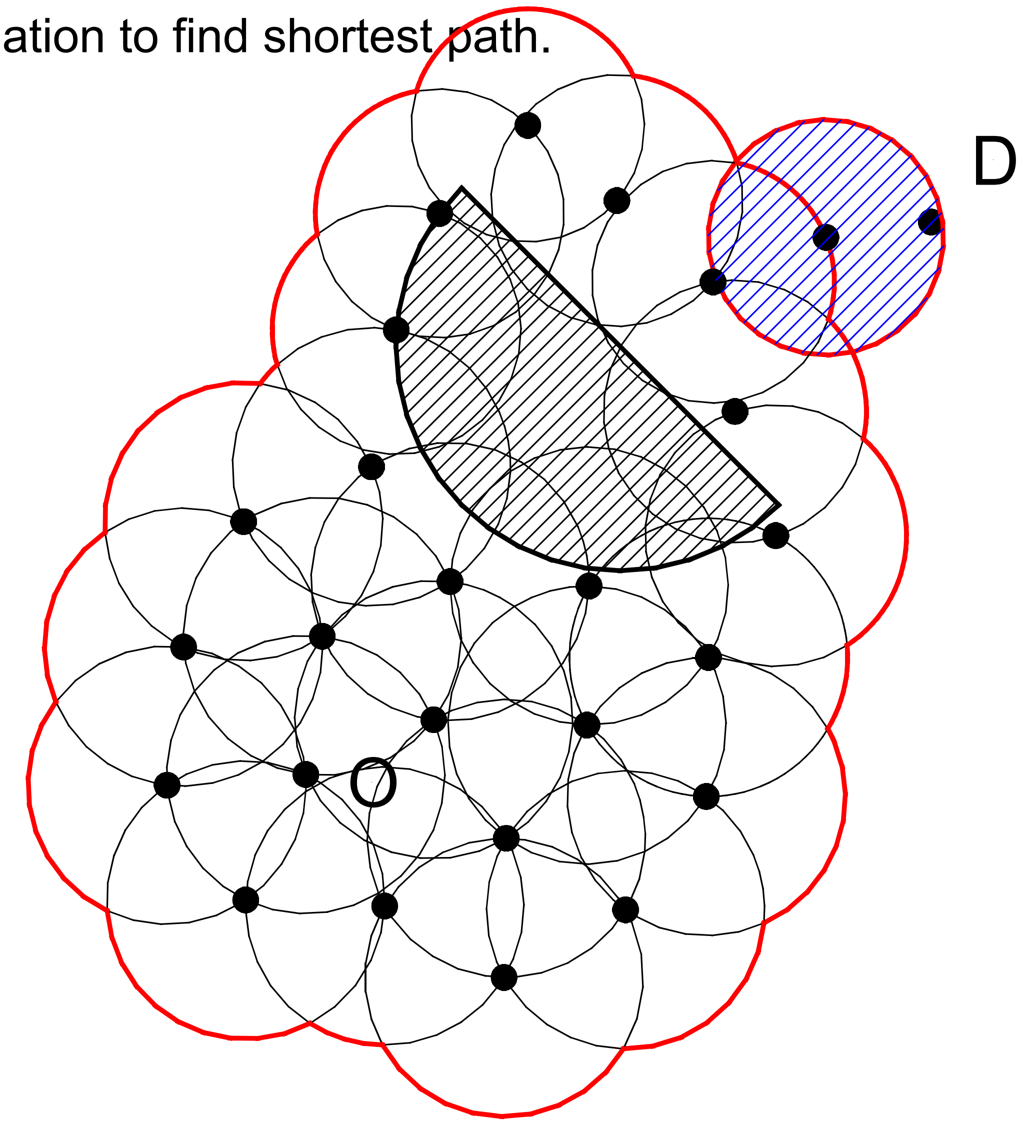


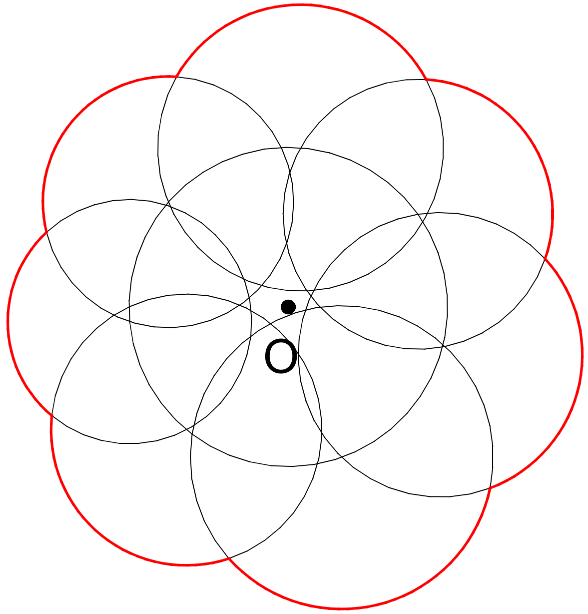
Using continuation to find shortest path.



Inside union know dist(O,*)

Using continuation to find shortest path.

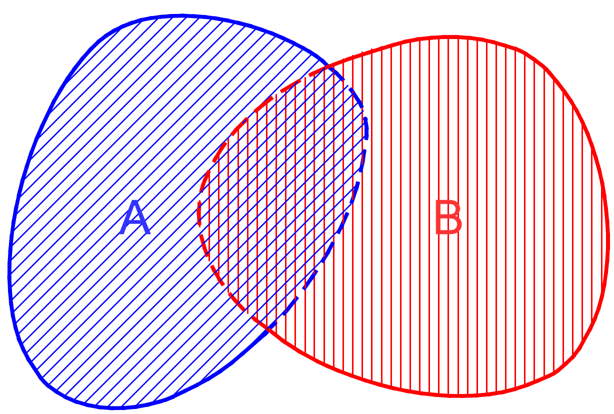




Visited region is the union
of spherical neighborhoods.

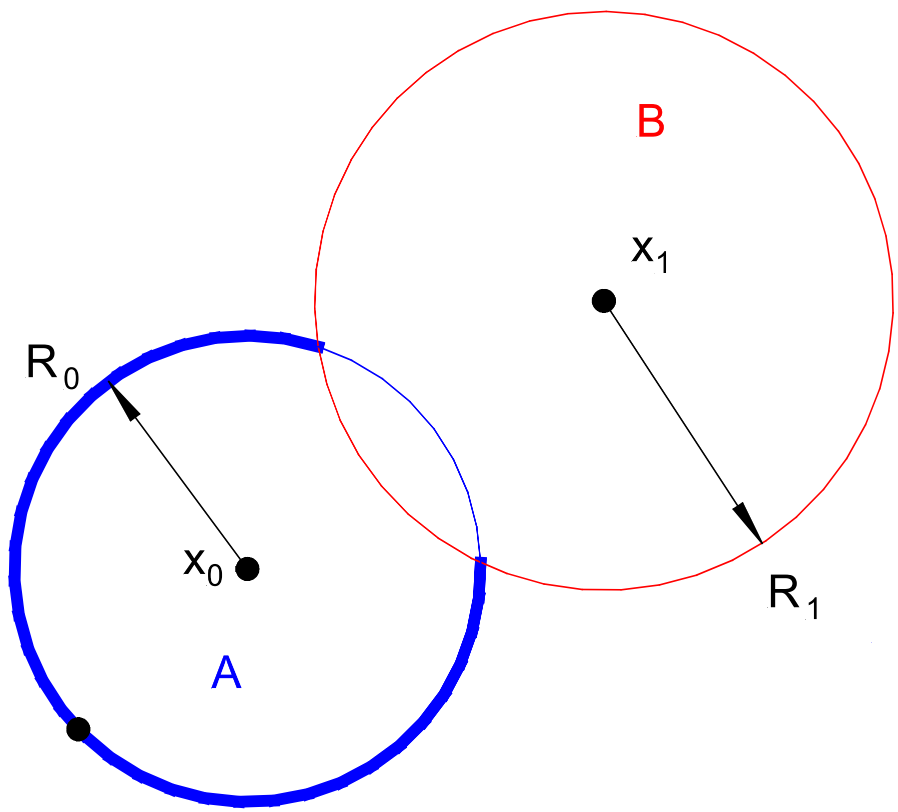
Need to find a point on the boundary.

$$\delta (A \cup B) = (\delta A \cap !B) \cup (\delta B \cap !A)$$



The boundary of the union of spheres

$$\delta\,(A \cup B) = (\delta A \cap\, !B) \cup (\delta B \cap\, !A)$$



$\delta A \quad |x - x_0|^2 = R_0^2$

$!B \quad |x - x_1|^2 > R_1^2$

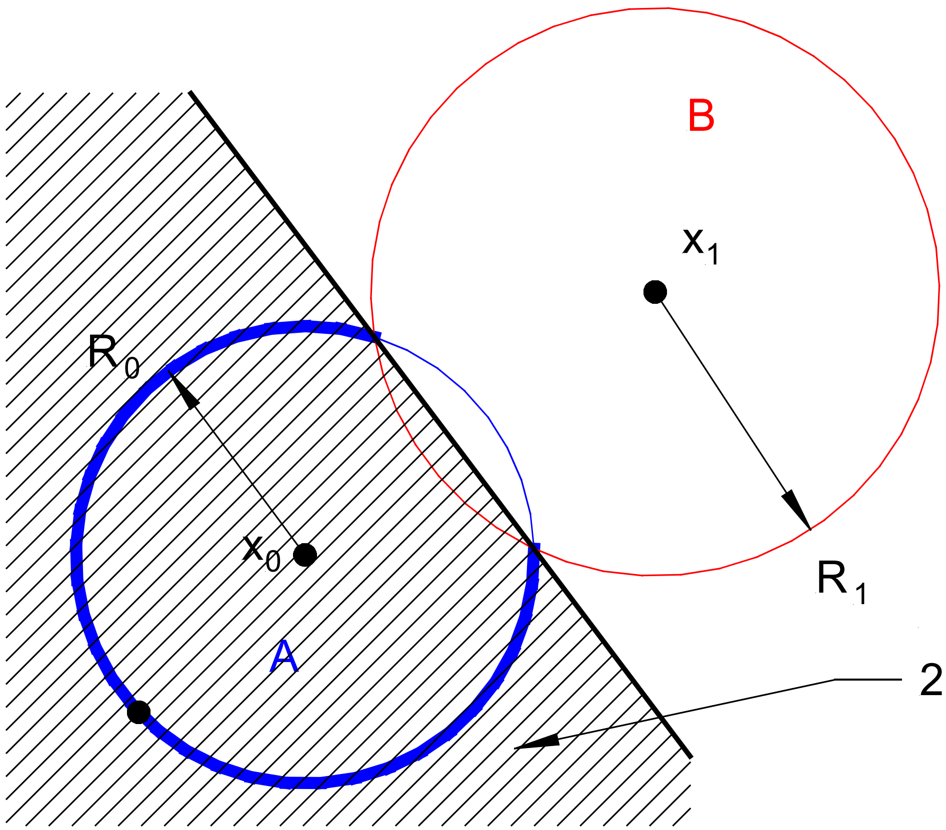
$\delta A \quad |x|^2 - 2\,x.x_1 > R_1^2 - |x_1|^2$

$!B \quad |x|^2 - 2\,x.x_0 = R_0^2 - |x_0|^2$

$\delta A \cap\, !B$

The boundary of the union of spheres

$$\delta(A \cup B) = (\delta A \cap !B) \cup (\delta B \cap !A)$$



$\delta A \quad |x - x_0|^2 = R_0^2$

$!B \quad |x - x_1|^2 > R_1^2$

$\delta A \quad |x|^2 - 2\,x.x_1 > R_1^2 - |x_1|^2$

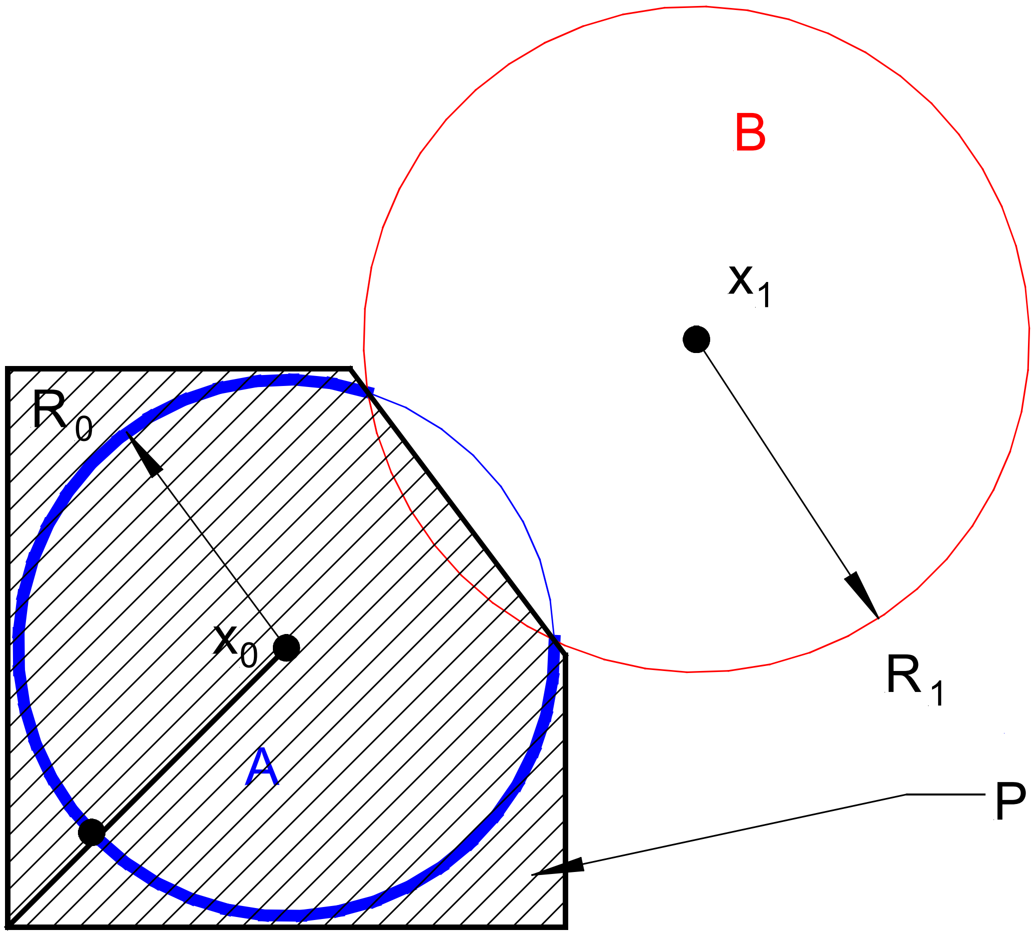
$!B \quad |x|^2 - 2\,x.x_0 = R_0^2 - |x_0|^2$

$2\,x.(x_1 - x_0) < R_0^2 - R_1^2 - |x_0|^2 + |x_1|^2$

$\delta A \cap !B$

The boundary of the union of spheres

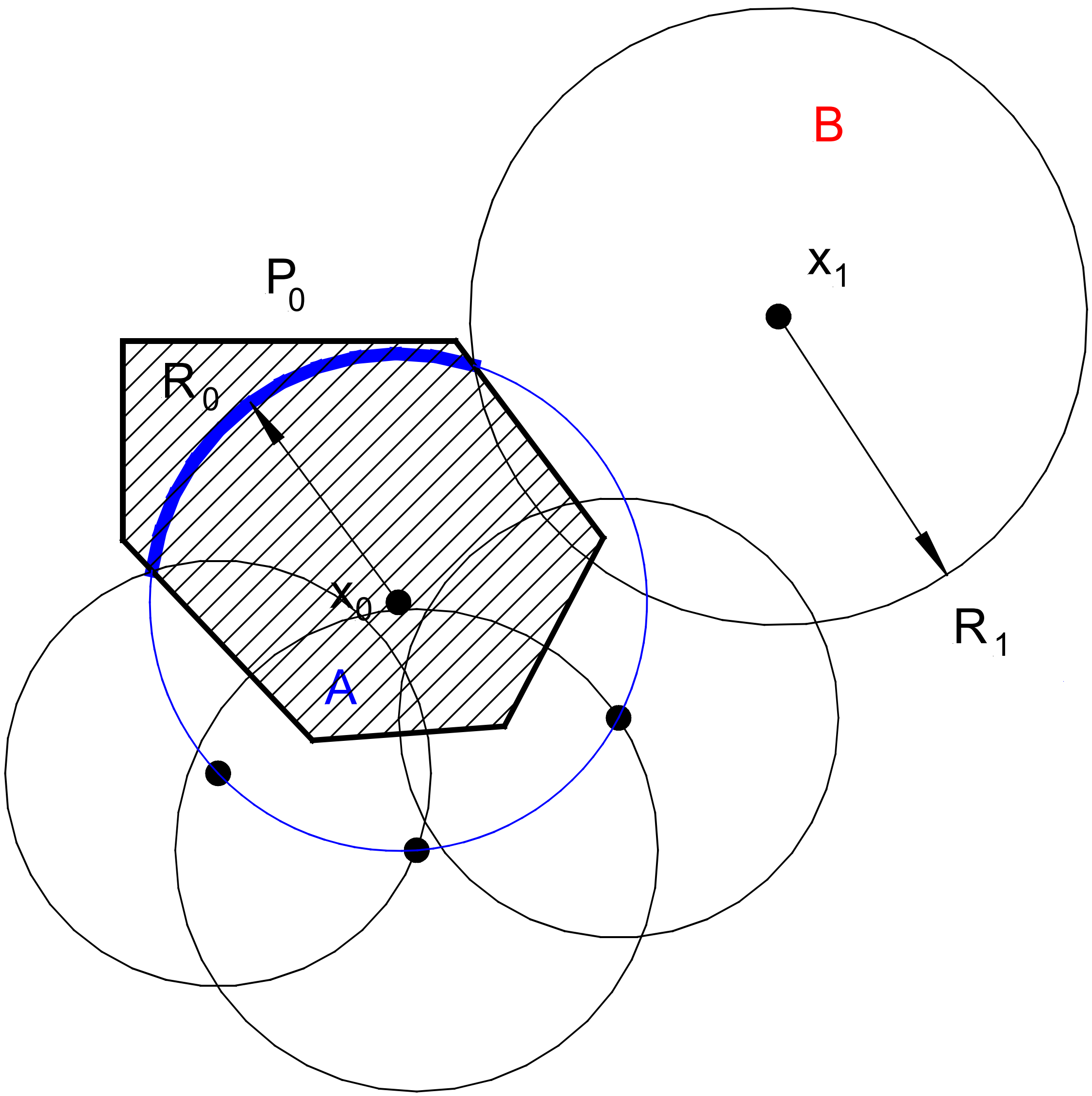
$$\delta (A \cup B) = (\delta A \cap !B) \cup (\delta B \cap !A)$$



$\delta A \cap !B$

The boundary of the union of spheres

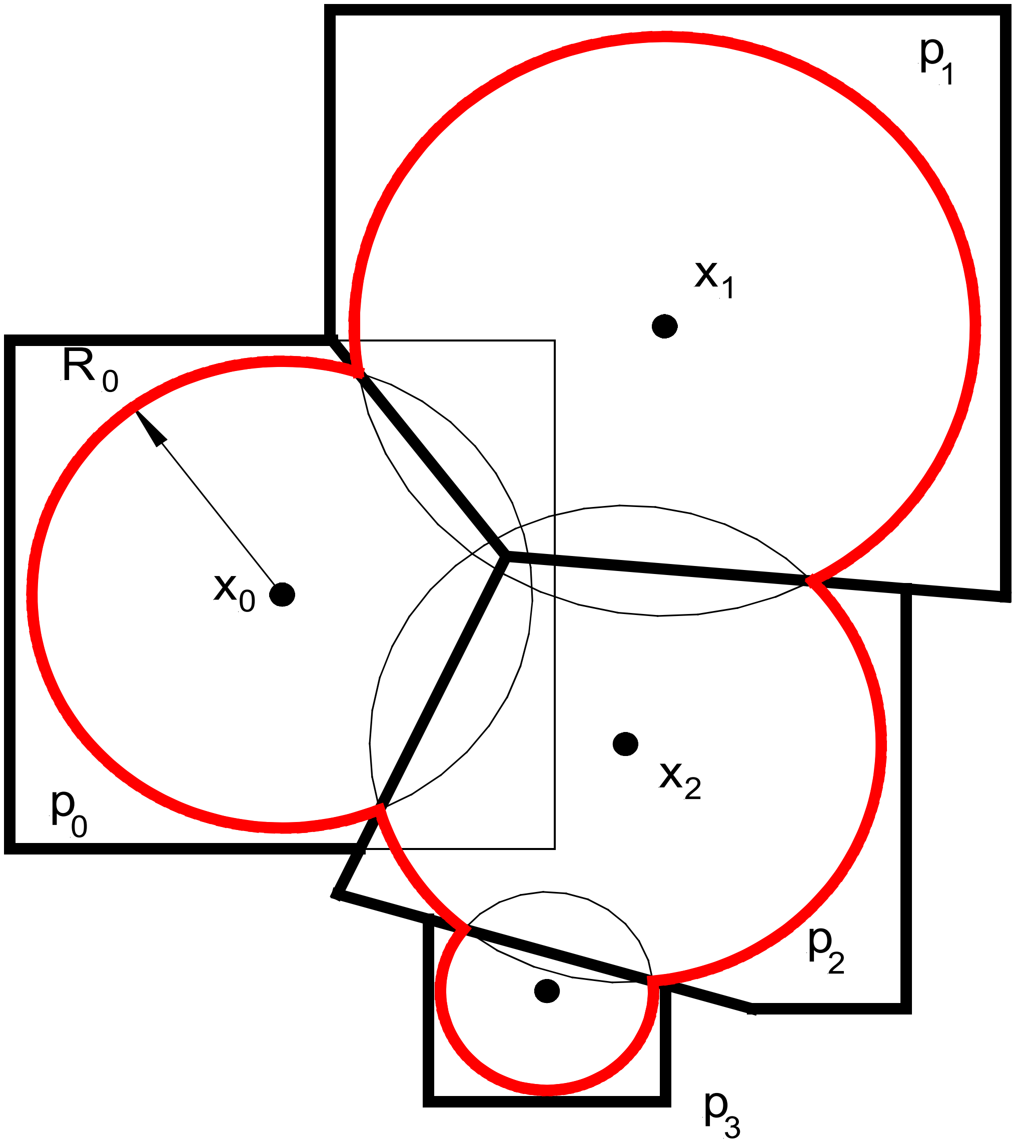
$$\delta (A \cup B) = (\delta A \cap !B) \cup (\delta B \cap !A)$$



$$P_0 = C \cap H_{01} \cap H_{02} \cap H_{03} \cap H_{04}$$

Finite, convex

The boundary of the union of spheres



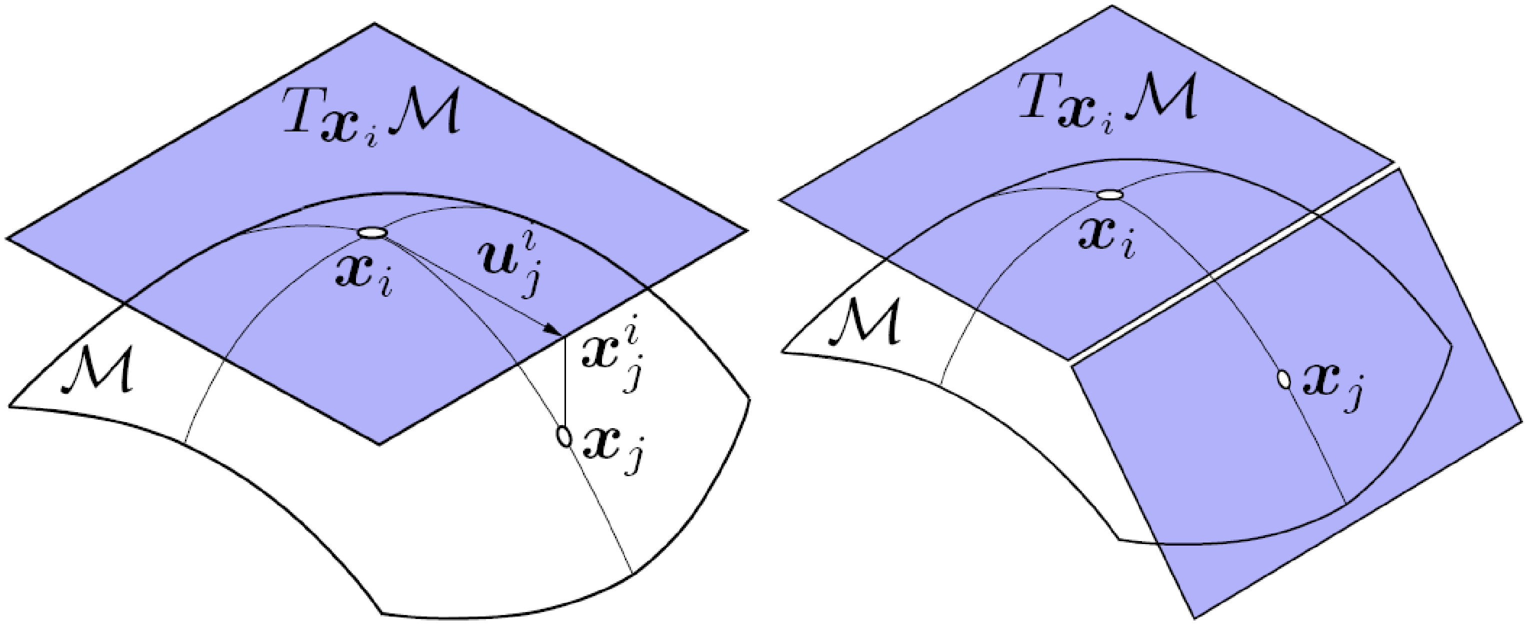
On boundary ↔ on sphere and in P

Keep a simple list of x,R,P

Choose one with a vertex of P
|v-x|>R

Restricted Laguerre-Voronoi diagram

On a curved configuration space.



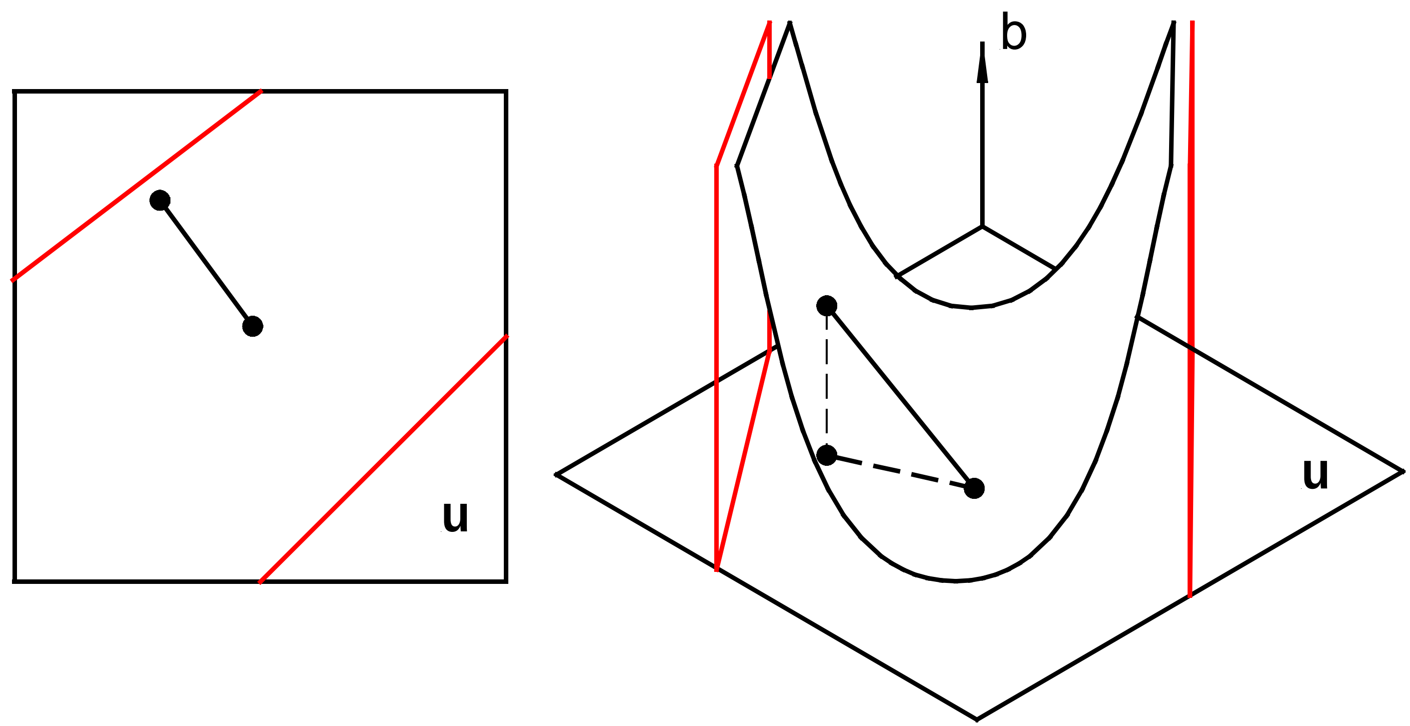
Avoiding singularities

$F(u)=0$

$b = 1/\mu(u)$

indicator function $\mu(u)$ e.g $\det(F_u(u))$

In (u,b) space singularities get mapped to infinity



Example: “3-RRR” planar manipulator.

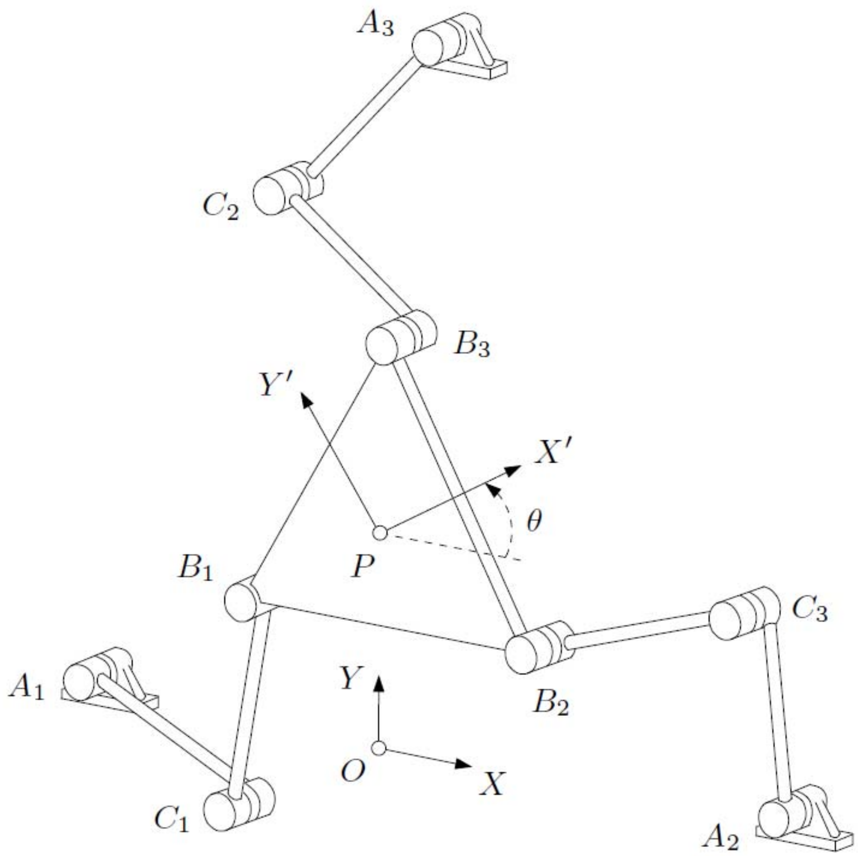
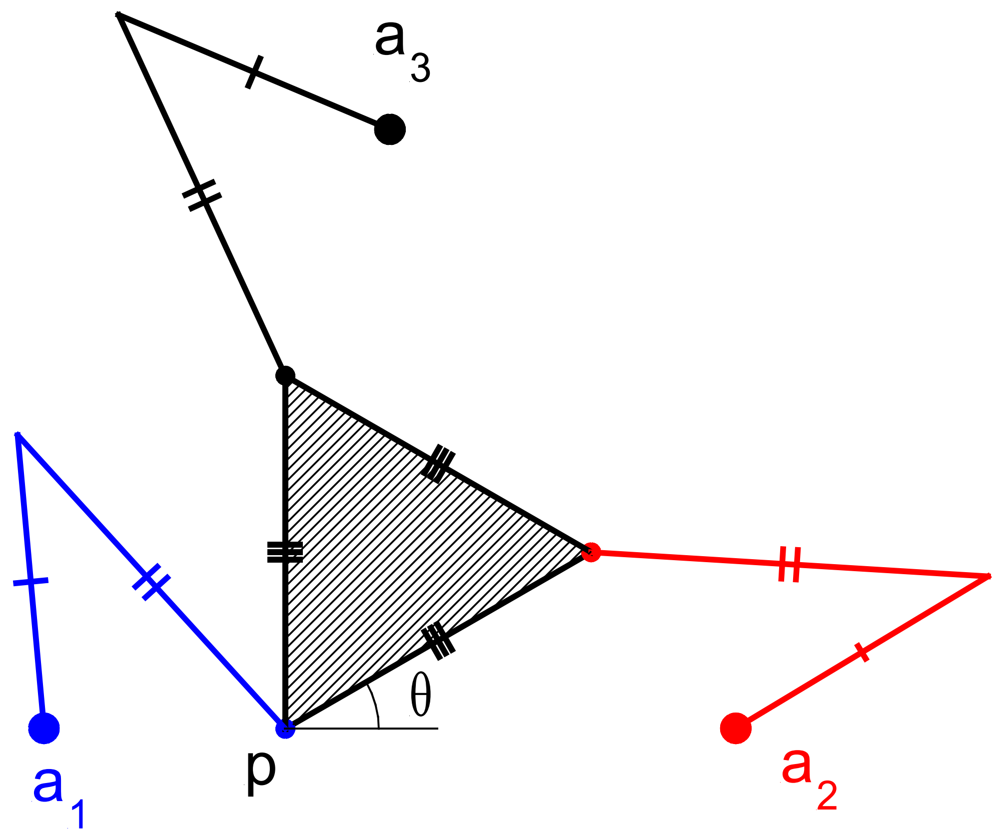
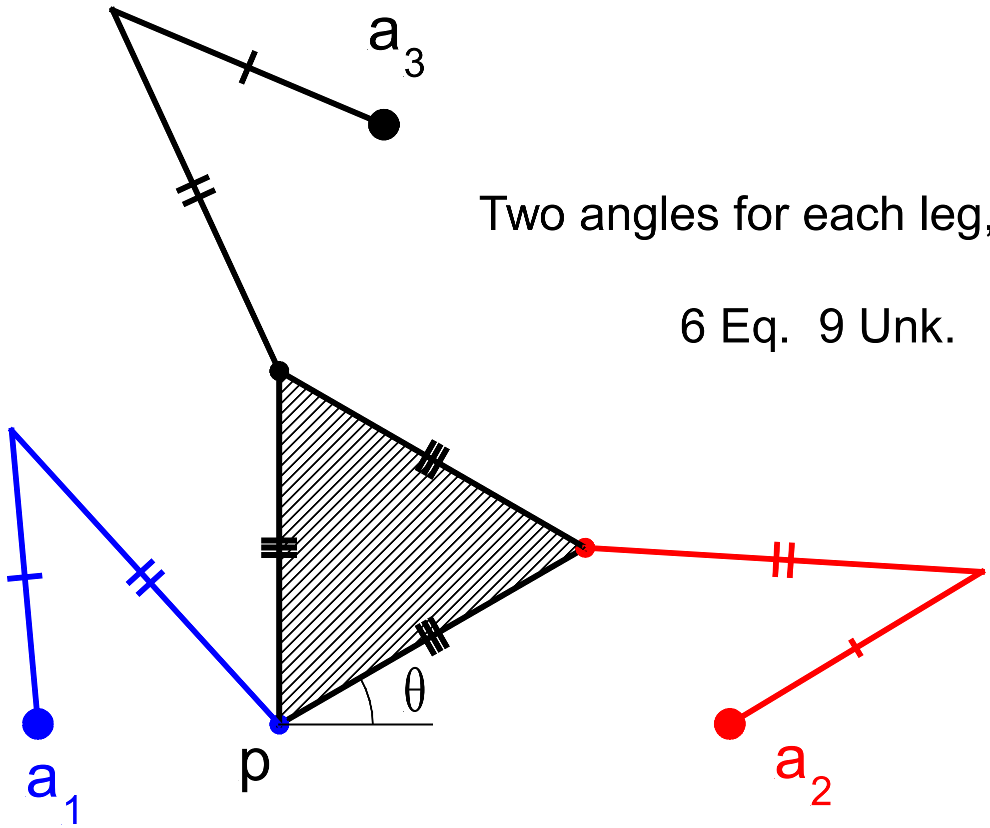


Fig. 9. A 3-RRR planar manipulator. Points A_1, A_2, and A_3 are fixed to the ground.



The pivots lie on an equilateral triangle with side 2.35
Each leg is two links, lengths 1 and 1.35
Platform is an equilateral triangle with side 1.2



Two angles for each leg, + p and θ

6 Eq. 9 Unk.

$$\mathbf{a}_1 = (0,0)$$

$$\mathbf{a}_2 = 2.35(1,0)$$

$$\mathbf{a}_3 = 2.35(\frac{1}{2}, \frac{\sqrt{3}}{2})$$

$$\mathbf{b}_1 = (0,0)$$

$$\mathbf{b}_2 = 1.2(1,0)$$

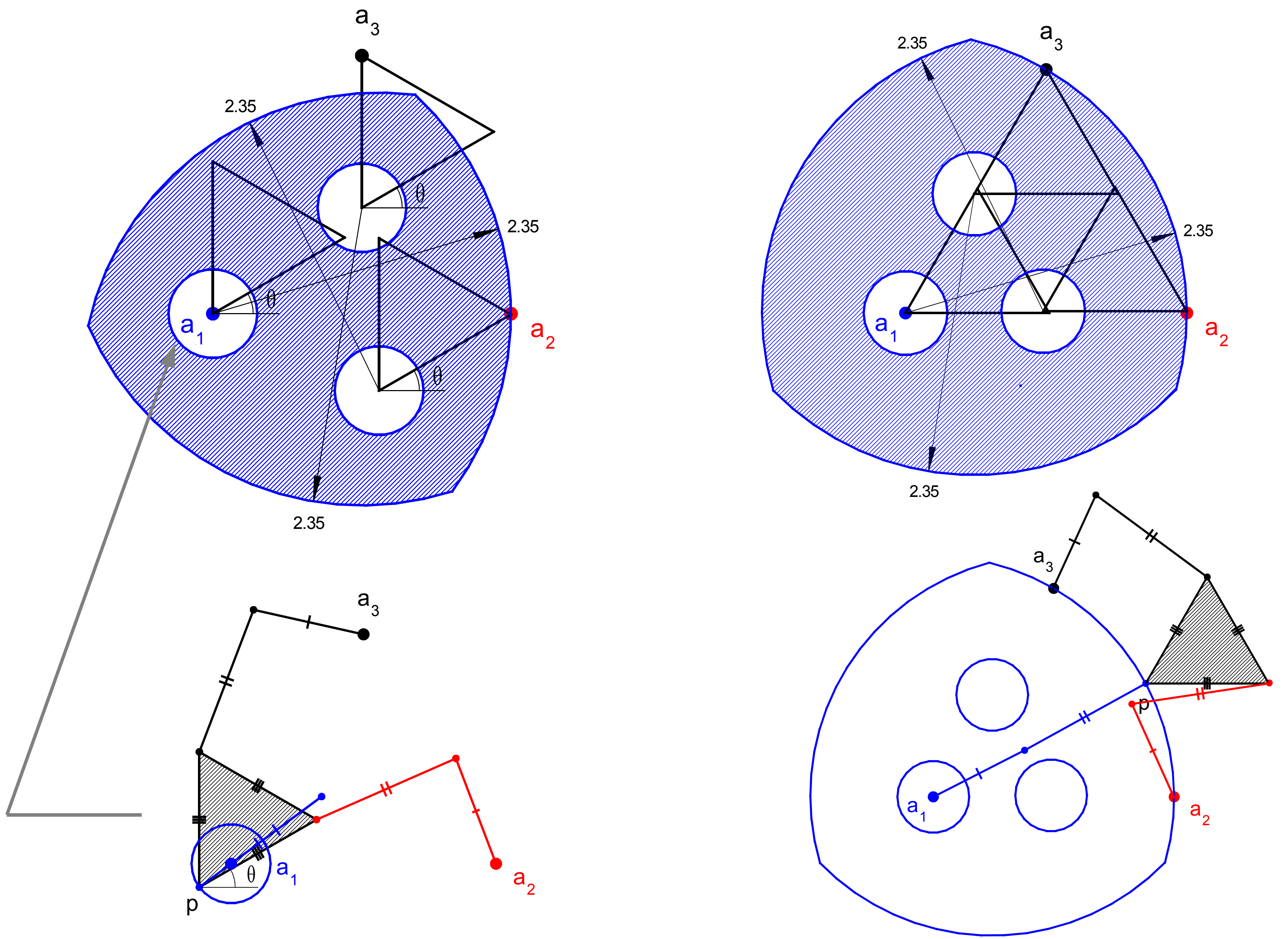
$$\mathbf{b}_3 = 1.2(\frac{1}{2}, \frac{\sqrt{3}}{2})$$

$$\mathbf{a}_1 + (\cos\phi_{11}, \sin\phi_{11}) + 1.35(\cos\phi_{12}, \sin\phi_{12}) - \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \mathbf{b}_1 = \mathbf{p}$$

$$\mathbf{a}_2 + (\cos\phi_{21}, \sin\phi_{21}) + 1.35(\cos\phi_{22}, \sin\phi_{22}) - \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \mathbf{b}_2 = \mathbf{p}$$

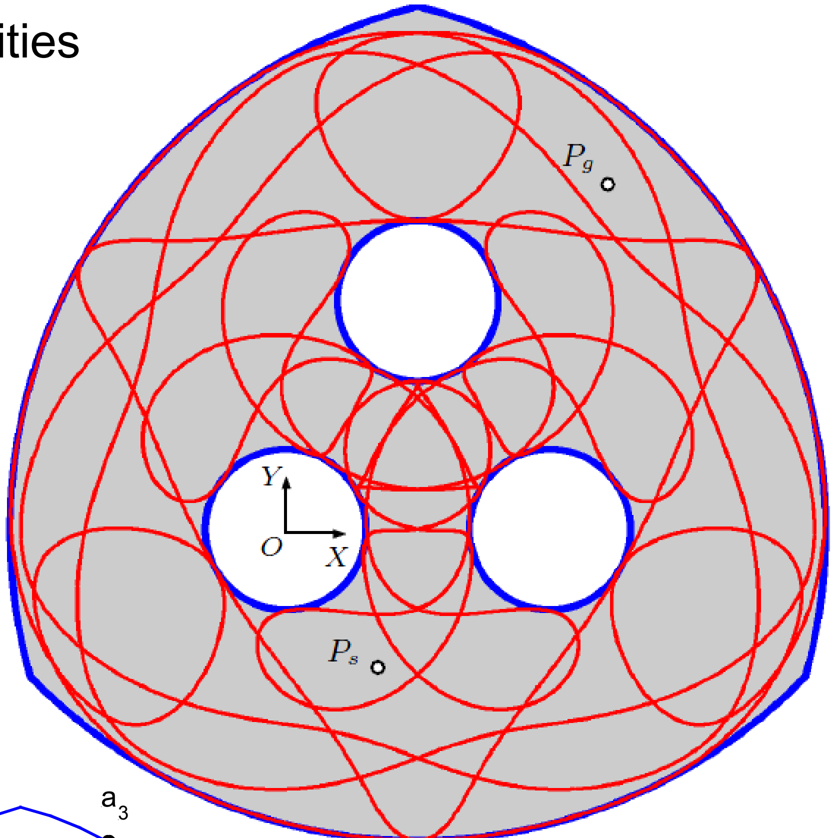
$$\mathbf{a}_3 + (\cos\phi_{31}, \sin\phi_{31}) + 1.35(\cos\phi_{32}, \sin\phi_{32}) - \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \mathbf{b}_3 = \mathbf{p}$$

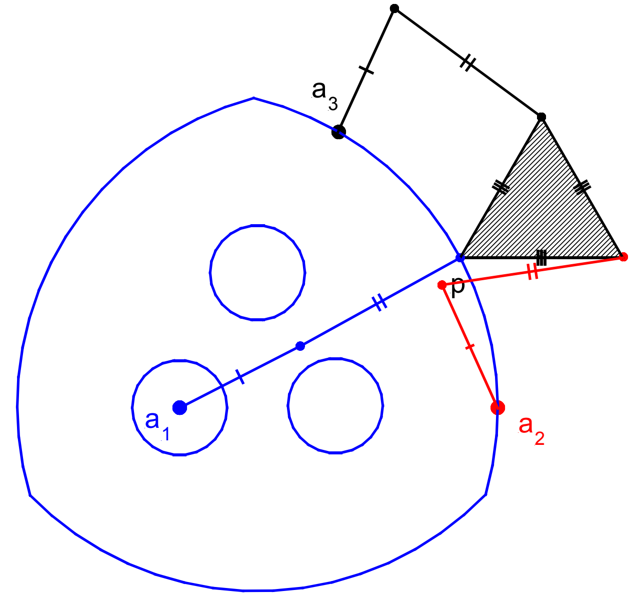
A projection of configuration space for 3-RRR



Singularities

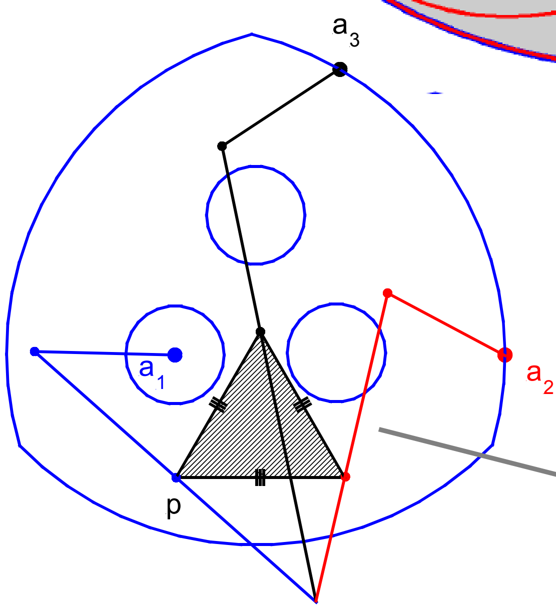
$\theta=0$



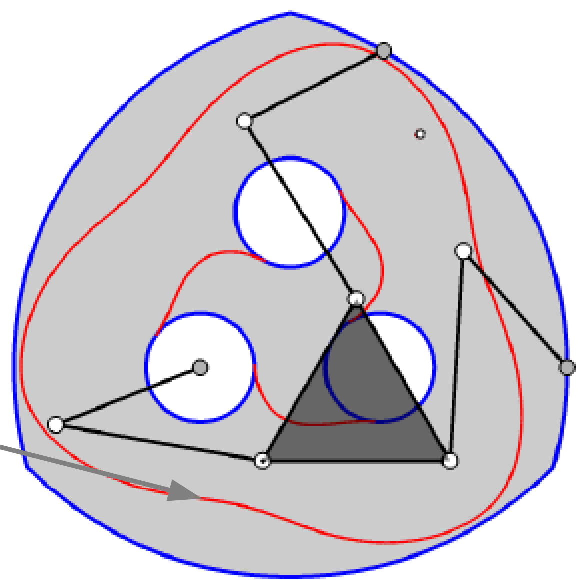


Fold when one leg is straight.

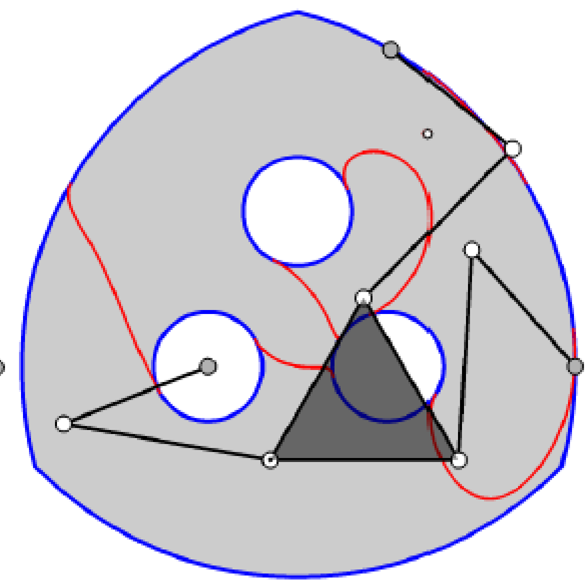
Eight different sets/way legs bend.



Singular when the three legs cross at a point



$(+,+,+)$



$(+,+,-)$

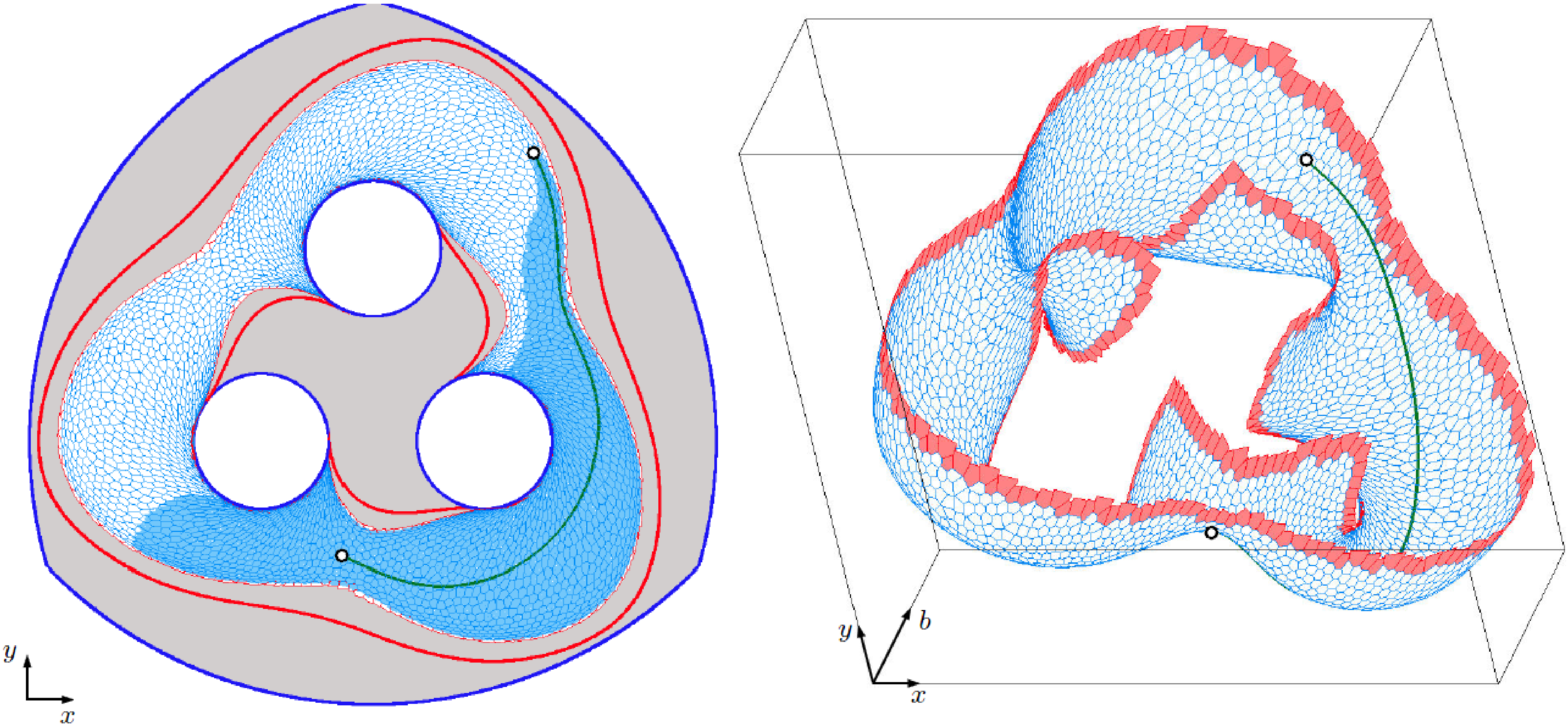


Fig. 11. Results of the method on computing a singularity-free path to connect points P_s and P_g indicated in Fig. 10, assuming the platform orientation fixed to $\theta = 0$. All results are shown projected onto the (x, y)-plane (left) and onto the (x, y, b)-space (right). The obtained path is shown in green overlaid onto the atlas of the singularity-free component of $\mathcal{C}$ attainable from the start configuration. The atlas charts are shown colored in white, with blue or red edges depending on whether they lie inside or outside of the domain $\mathcal{D}$. The part actually explored by the algorithm to connect the two configurations is shown shadowed in blue on the left figure. An animation of this figure is available in the supplementary downloadable material associated with this paper.

Summary

To find a non-singular path from one configuration to another.

Work with (u,b) with b=1/det.

Use a directed version of Dijkstra's algorithm on a surface

Visited region is a union of spherical balls in the tangent space.

Store u, tangent space T, radius R, and polyhedron P

Choose a point on the boundary “nearest” the destination.

Point on sphere, inside polyhedron.

Update polyhedra by subtracting half spaces.

Bohigas, H, Ros, Manubens, Porta

Planning Singularity-free Paths on Closed-Chain Manipulators

To appear, IEEE Transactions on Robotics