Autonomous Mobile Robots

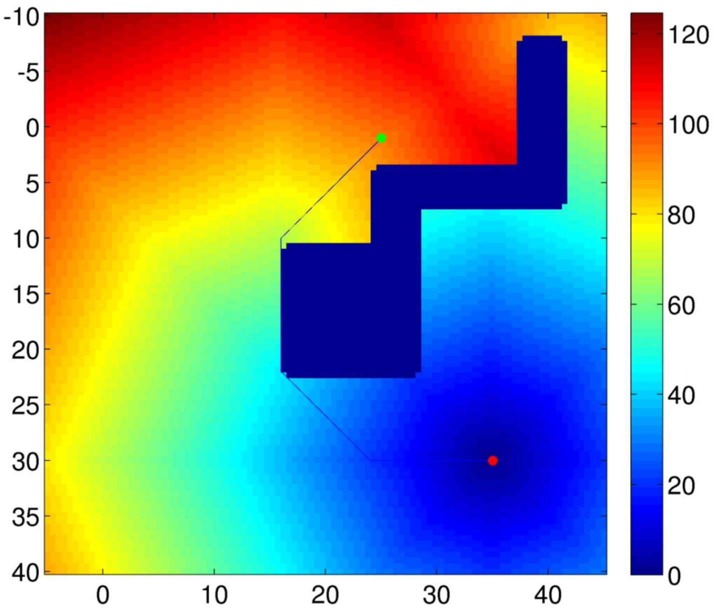
Exercise 6 : Dijkstra's Algorithm and the Dynamic Window Approach for Motion Planning

Daniel Dugas, Rik Bähnemann

Planning Algorithm

Input: Obstacle Map, Start Pose, Goal Pose
Output: Feasible Robot Path
Algorithm:

1. Create distance field with Dijkstra
2. While (not at goal):
   1. Follow gradient with DWA



Recap Dijkstra

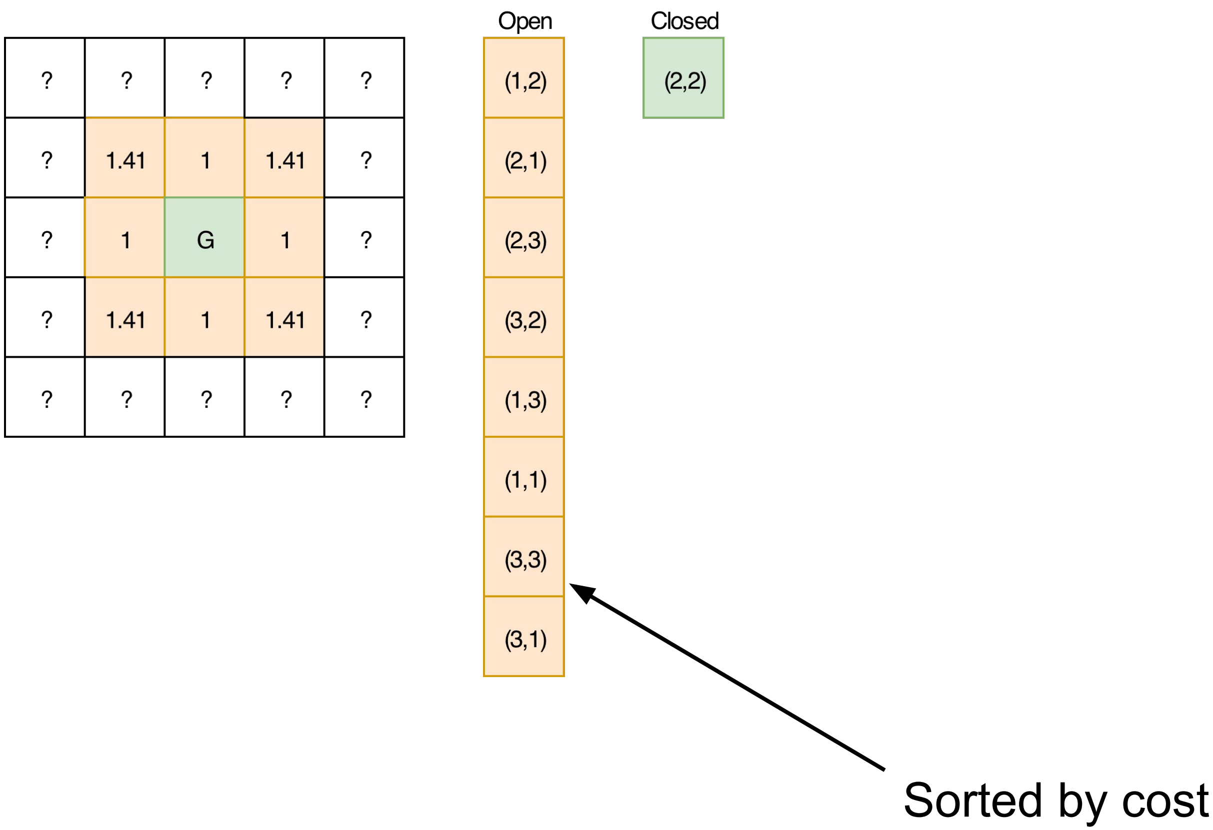
- While queue is not empty and not at goal…

(0,0)	(0,1)	(0,2)	(0,3)	(0,4)
(1,0)	(1,1)	(1,2)	(1,3)	(1,4)
(2,0)	(2,1)	(2,2)	(2,3)	(2,4)
(3,0)	(3,1)	(3,2)	(3,3)	(3,4)
(4,0)	(4,1)	(4,2)	(4,3)	(4,4)

Open	Closed
(2,2)	

Recap Dijkstra

- Pop front node from queue
- Expand and add new nodes to queue
- Resolve double insertions



Recap Dijkstra

…

?	2.41	2	2.41	?
?	1.41	1	1.41	?
?	1	G	1	?
?	1.41	1	1.41	?
?	?	?	?	?

Open
(2,1)
(2,3)
(3,2)
(1,3)
(1,1)
(3,3)
(3,1)
(0,2)
(0,1)
(0,3)

Closed
(2,2)
(1,2)

Recap Dijkstra

…

?	2.41	2	2.41	?
2.41	1.41	1	1.41	?
2	1	G	1	?
2.41	1.41	1	1.41	?
?	?	?	?	?

Open	Closed
(2,3)	(2,2)
(3,2)	(1,2)
(1,3)	(2,1)
(1,1)	
(3,3)	
(3,1)	
(0,2)	
(2,0)	
(0,1)	
(0,3)	
(1,0)	
(3,0)	

Recap Dijkstra

- Optimal solution in case of positive edge costs
- $O(|V|\log(|V|) + |E|)$
- Speed up with heuristic (A*)

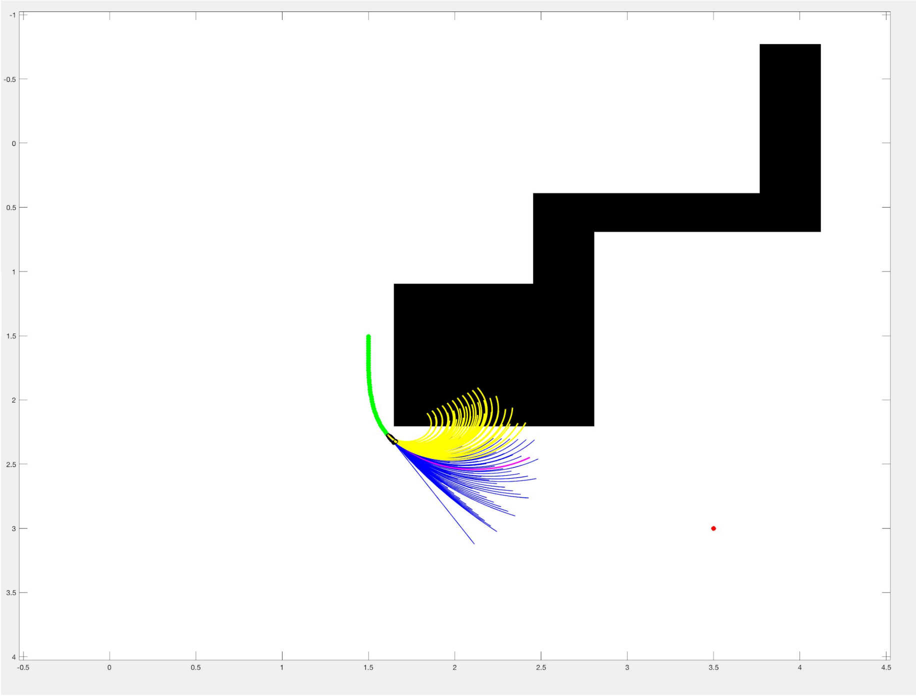
Recap Dynamic Window Approach

Input: Obstacle Map, Current State, Goal Pose
Output: Next Control Input
Algorithm:

1. Sample feasible inputs
2. For all feasible inputs:
    1. Compute trajectory over horizon
    2. Score trajectory

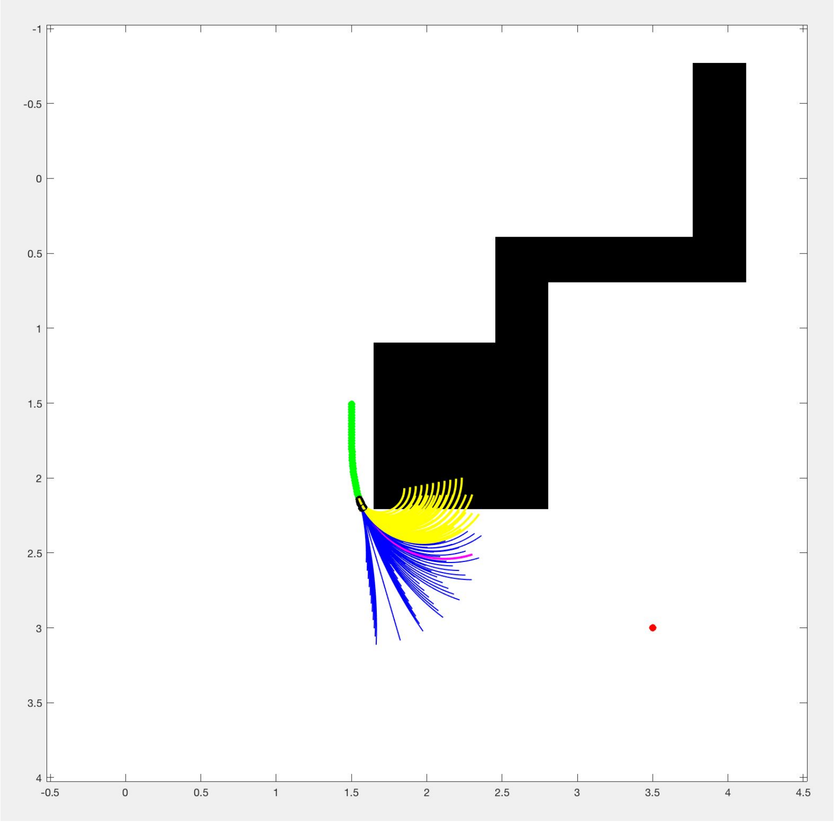
    $$G(v,\omega) = \alpha\,\text{heading}(v,\omega) + \beta\,\text{dist}(v,\omega) + \gamma\,\text{velocity}(v,\omega)$$

3. Pick trajectory with best score

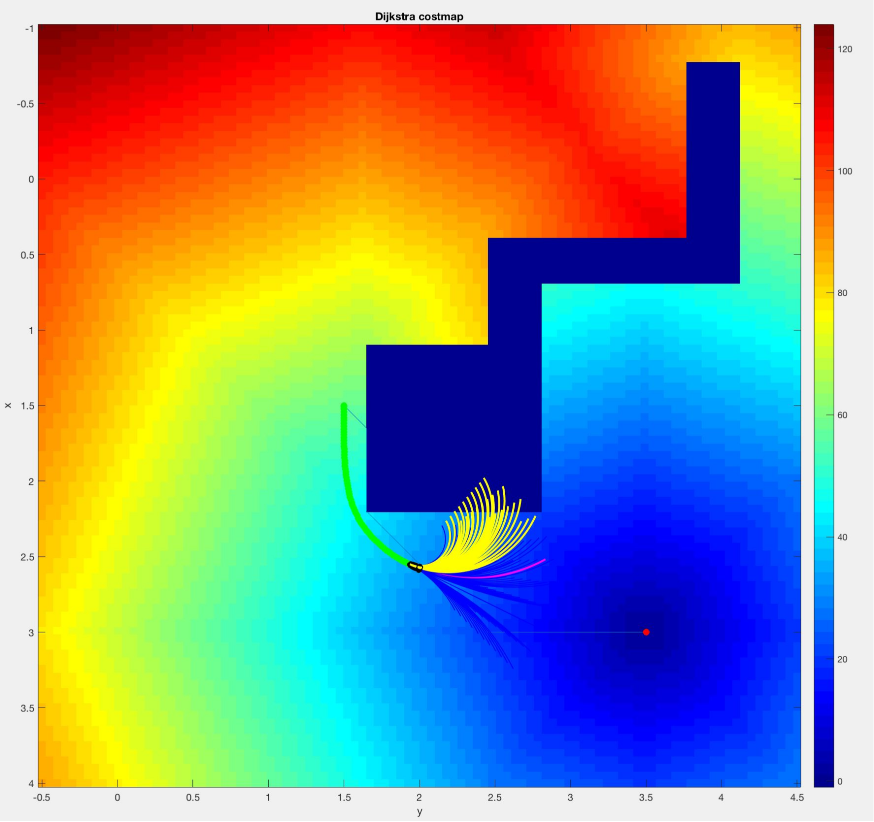


Recap Dynamic Window Approach

Local: Heading towards goal



Global: Heading towards gradient

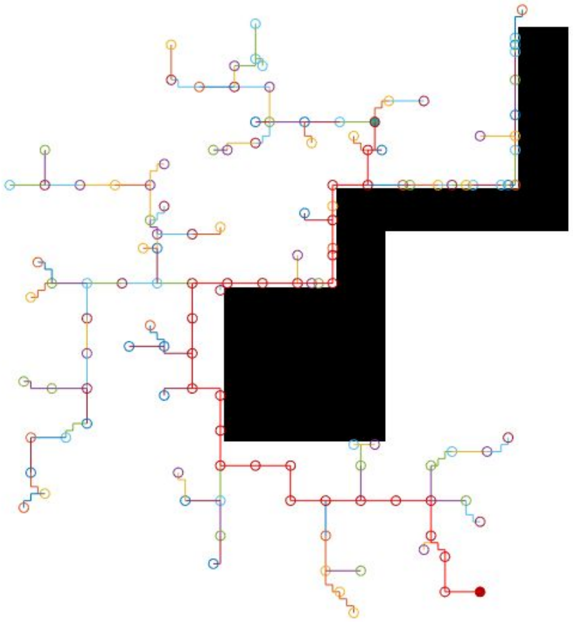


Recap RRT

Input: Obstacle Map, Current State, Goal Pose
Output: Feasible Robot Path
Algorithm:

1. Sample random pose
2. Try to link random pose to nearest pose in graph.
3. Repeat until goal pose is in graph

Extra - Not exam material



Recap Kinodynamic RRT

Input: Obstacle Map, Current State, Goal Pose
Output: Feasible Robot Path
Algorithm:
(Same as RRT, step 2 in linking sample with graph is kinetic/dynamically constrained)

Extra - Not exam material