Algorithmic Robotics and Motion Planning

Sampling-based motion planning II: Single query planners and the RRT family

Dan Halperin
School of Computer Science
Tel Aviv University

Fall 2019-2020

Overview

- RRT
- bi-RRT
- Poor quality solution paths
- RRT*
- Variants
- References

Two landmark papers

Steven M. LaValle, James J. Kuffner Jr.:
Randomized Kinodynamic Planning. ICRA 1999: 473-479

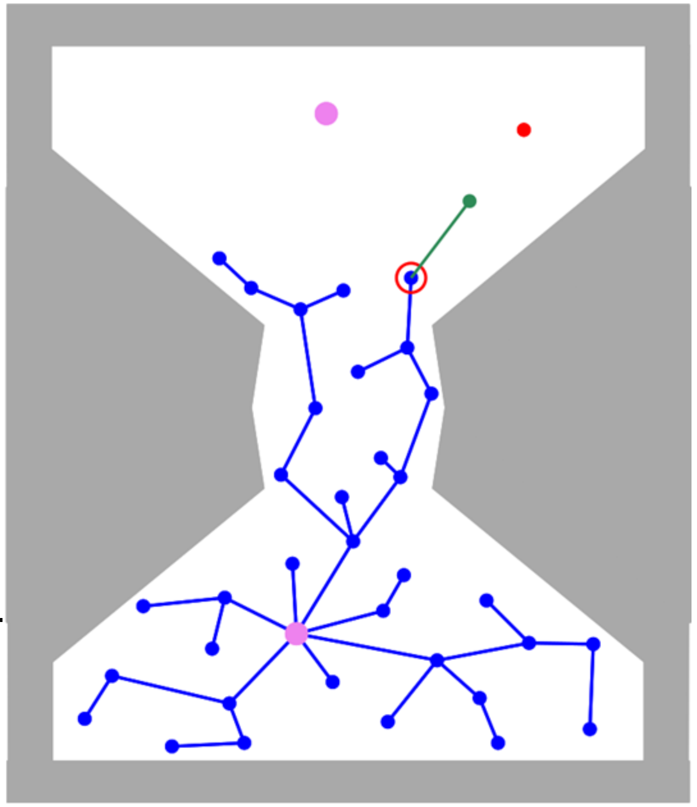
Sertac Karaman, Emilio Frazzoli:
Sampling-based algorithms for optimal motion planning.
I. J. Robotics Res. 30(7): 846-894 (2011)

The pseudocode (and more) in the following slides is from

Kiril Solovey, Lucas Janson, Edward Schmerling, Emilio Frazzoli, Marco Pavone: **Revisiting the Asymptotic Optimality of RRT.*** CoRR abs/1909.09688 (2019)

Algorithm 1 RRT $(x_{\texttt{init}} := s, x_{\texttt{goal}} := t, n, \eta)$

```
1: $V = \{x_{\text{init}}\}$
2: **for** $j = 1$ to $n$ **do**
3:     $x_{\text{rand}} \leftarrow$ SAMPLE-FREE( )
4:     $x_{\text{near}} \leftarrow$ NEAREST$(x_{\text{rand}}, V)$
5:     $x_{\text{new}} \leftarrow$ STEER$(x_{\text{near}}, x_{\text{rand}}, \eta)$
6:     **if** COLLISION-FREE$(x_{\text{near}}, x_{\text{new}})$ **then**
7:         $V = V \cup \{x_{\text{new}}\}$
8:         $E = E \cup \{(x_{\text{near}}, x_{\text{new}})\}$
9: **return** $G = (V, E)$
```



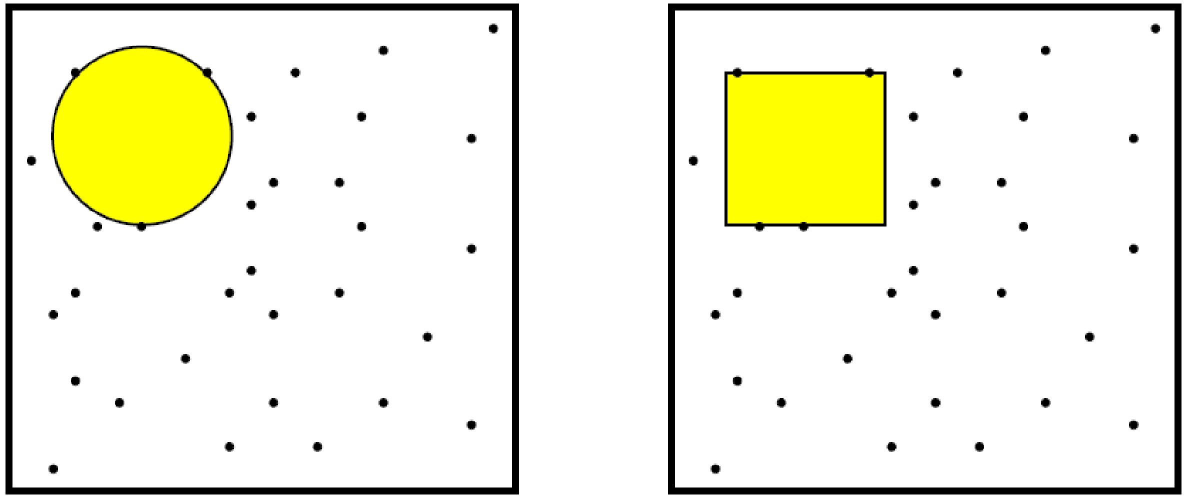
RRT –Rapidly-exploring Random Tree

Voronoi bias

Dispersion definition The *dispersion*[6] of a finite set P of samples in a metric space (X, ρ) is[7]

$$\delta(P) = \sup_{x \in X} \left\{ \min_{p \in P} \{\rho(x, p)\} \right\}. \tag{5.19}$$

- Reducing the dispersion = reducing the radius of the largest empty ball



(a) L_2 dispersion (b) L_∞ dispersion

[LaValle's book]

Voronoi bias, cont'd

- The exploration strategy of RRT has the Voronoi bias property:

The probability of a node in the tree to be expanded is proportional to the volume of its Voronoi cell in the Voronoi diagram of the existing nodes

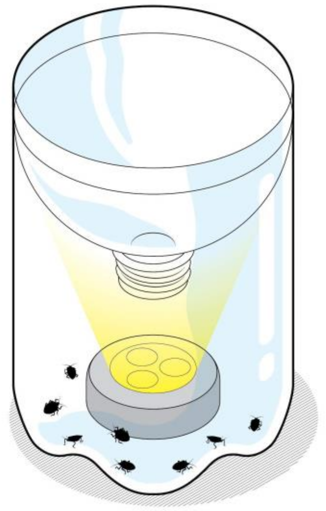
- The strategy can be viewed (roughly) as aiming to reduce the dispersion
- To exactly reduce the dispersion we should grow the tree toward the Voronoi vertex most distant from the tree nodes, from one of its nearest neighbors

Stephen R. Lindemann, Steven M. LaValle:
Incrementally Reducing Dispersion by Increasing Voronoi Bias in RRTs. ICRA 2004: 3251-3257

animation

BiRRT – Bidirectional RRT

- One can grow two trees T_s, T_t from start and target
  - In every iteration select one of the trees for expansion
- Force the trees to meet:
  - Every once in a while attempt to extend both trees toward the same sample
- Approach is helpful when one of the sides is harder than the other

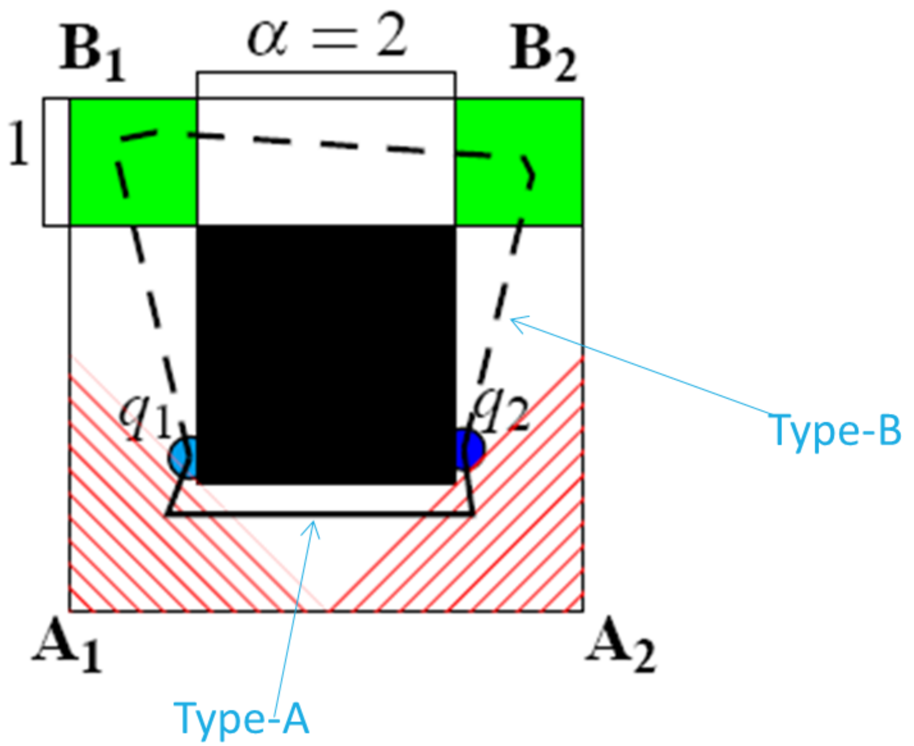


RRT, path quality

- RRT is probabilistically complete
- But can produce arbitrarily bad paths

Oren Nechushtan, Barak Raveh, Dan Halperin:
Sampling-Diagram Automata: A Tool for Analyzing Path Quality in Tree Planners. WAFR 2010: 285-301

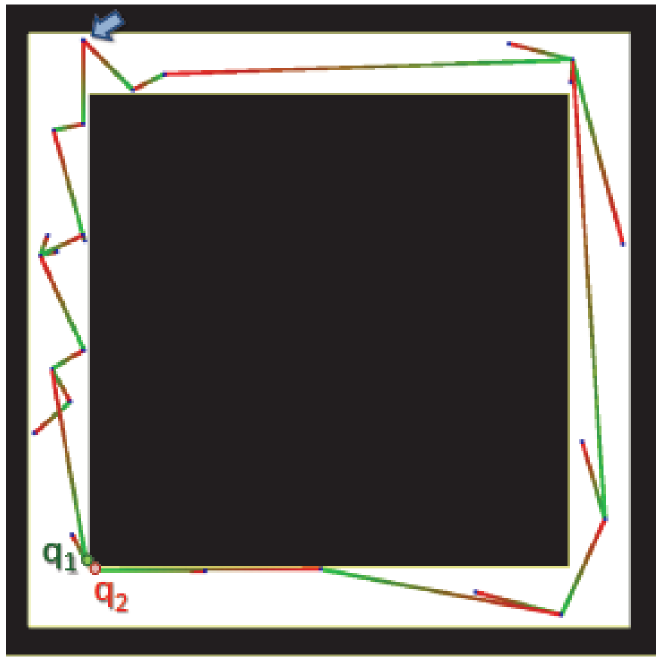
Experiments (I)

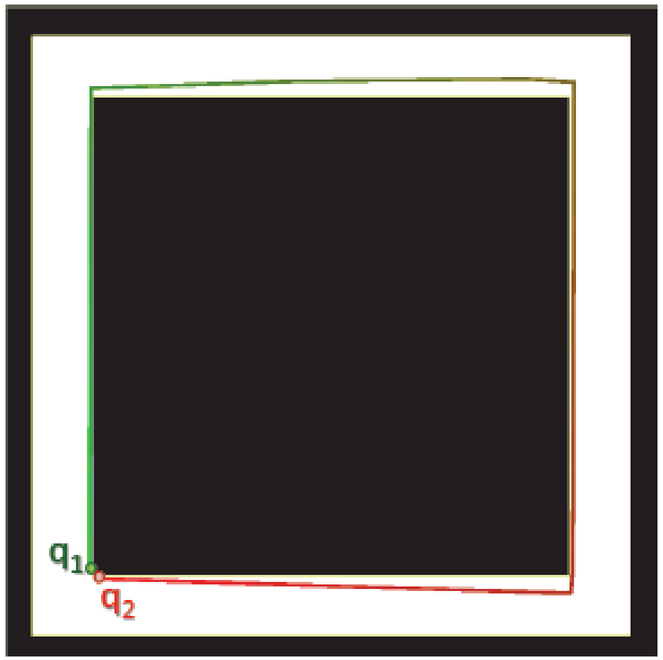


•49.4% of paths are <u>over three times </u>worse than optimal (even after smoothing)

•much larger than the theoretical bound

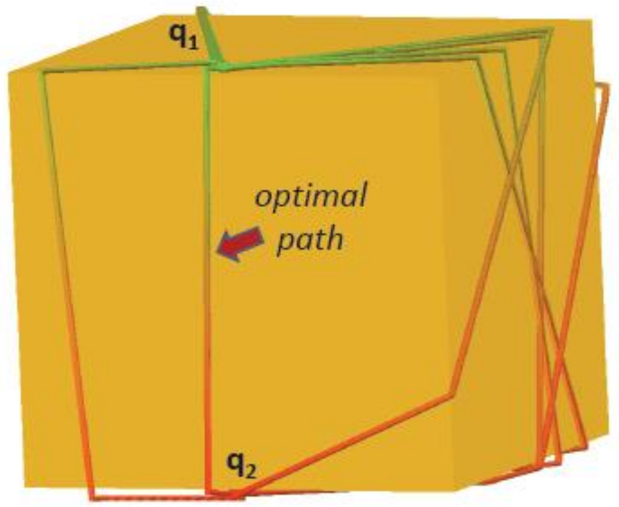
Experiments (II) – Close-By Start and Goal Configurations





- 5.9% of paths are over 140 times worse than optimal (even after smoothing)
- Conclusion: visibility blocking may be as important as narrow passages

Experiments with 3D Cube-Within-Cube



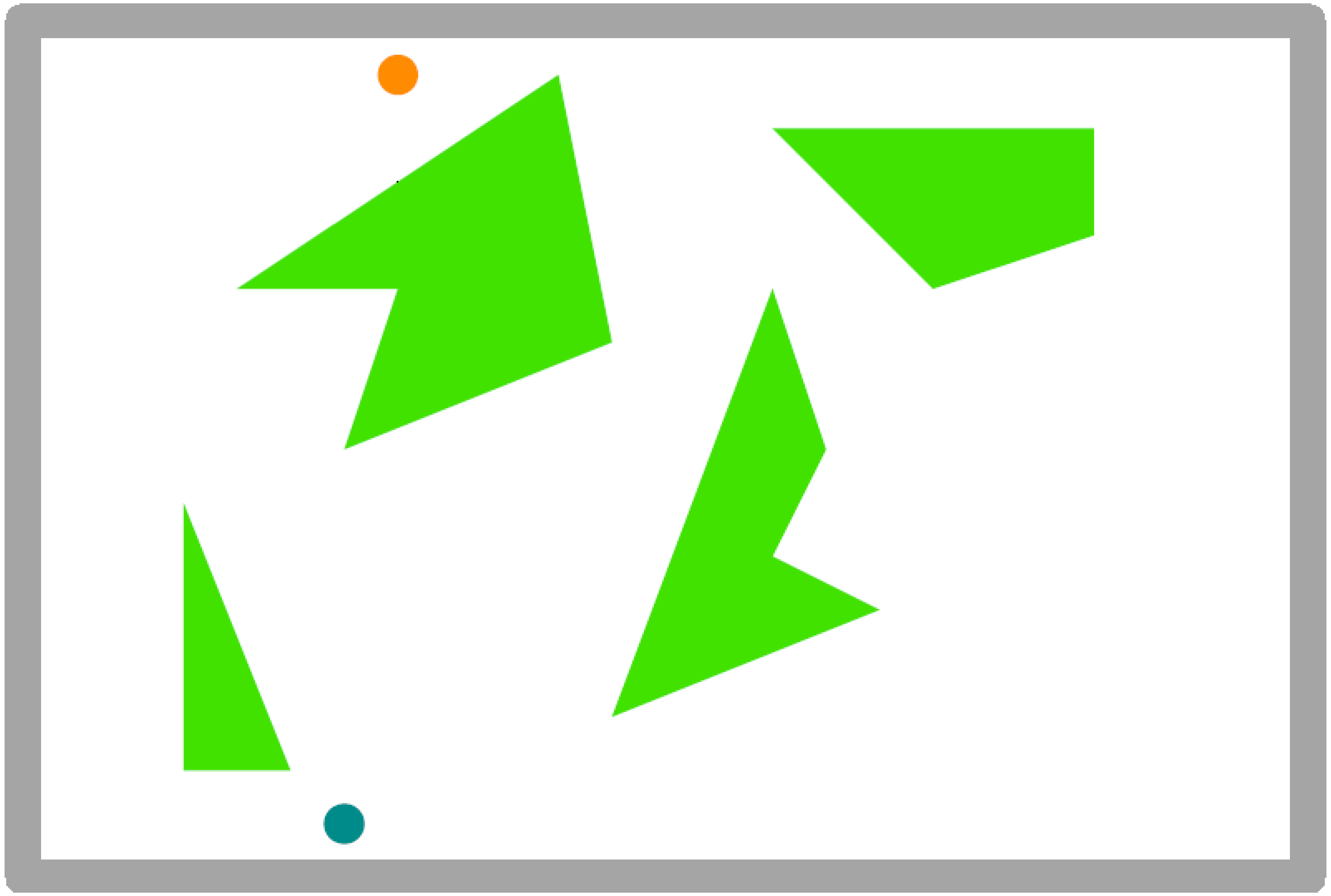
- 97.3% (!) of paths are over 1.2 times worse than optimal **after smoothing**
- Typically much more

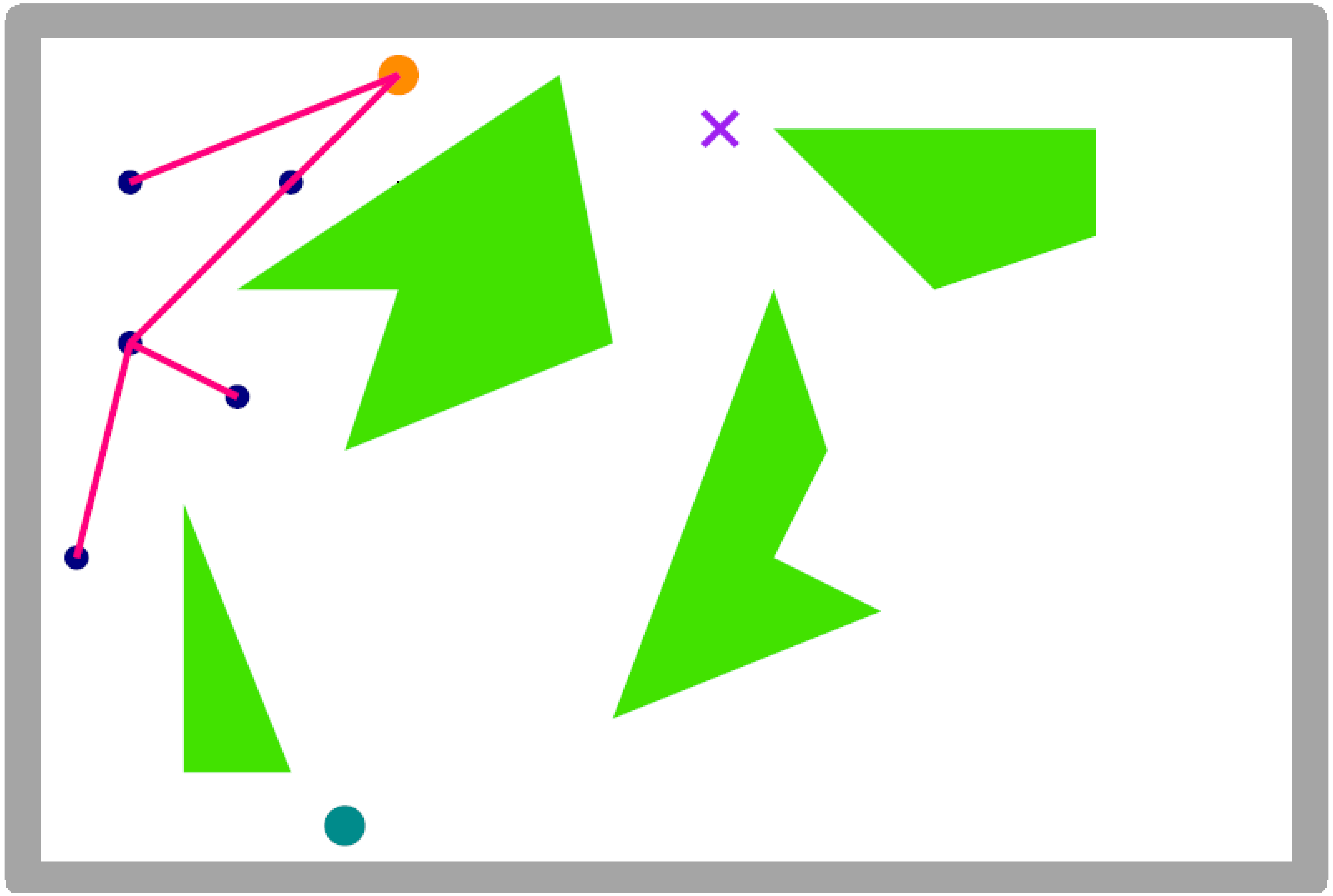
RRT*

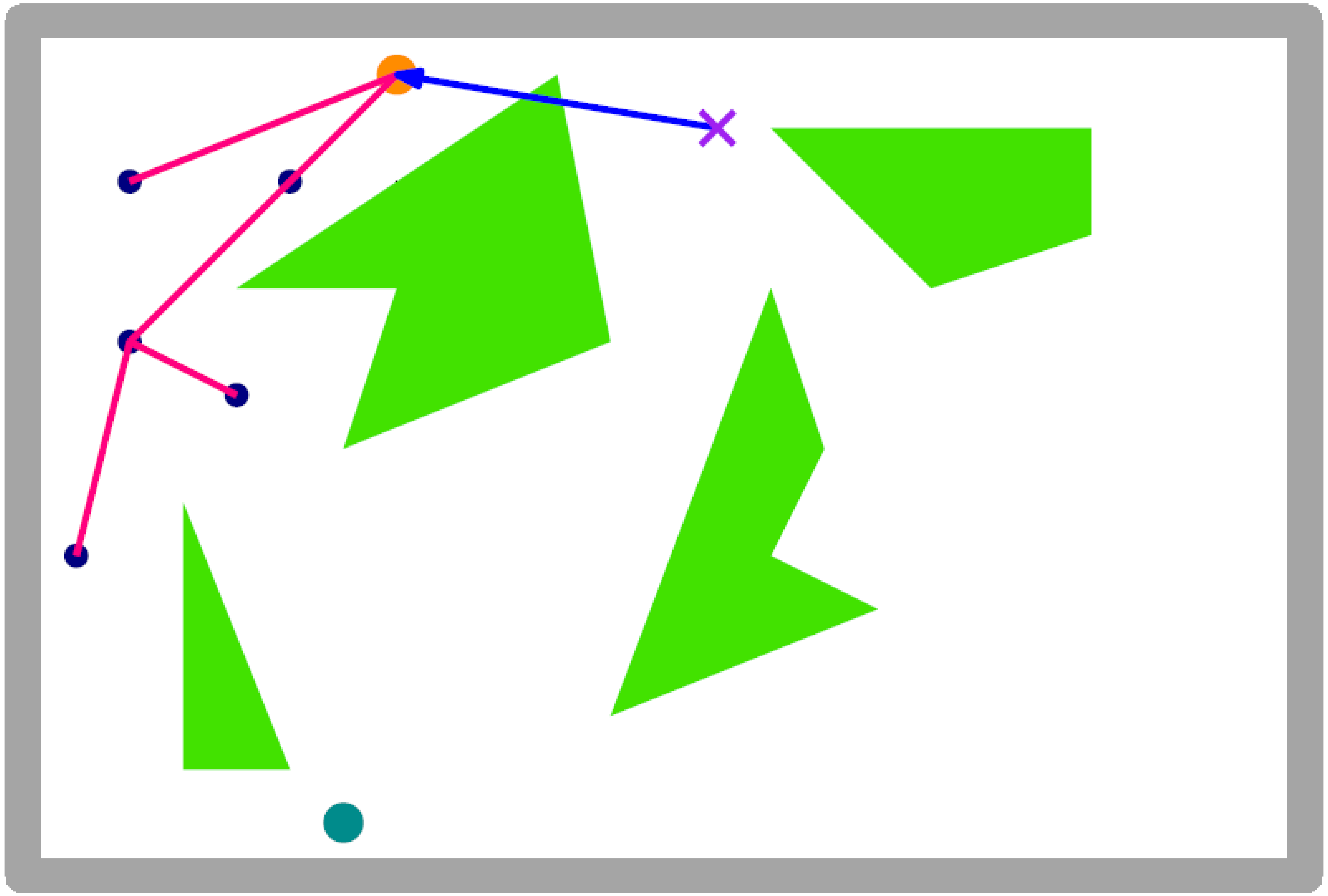
Enter Karaman and Frazzoli, 2011

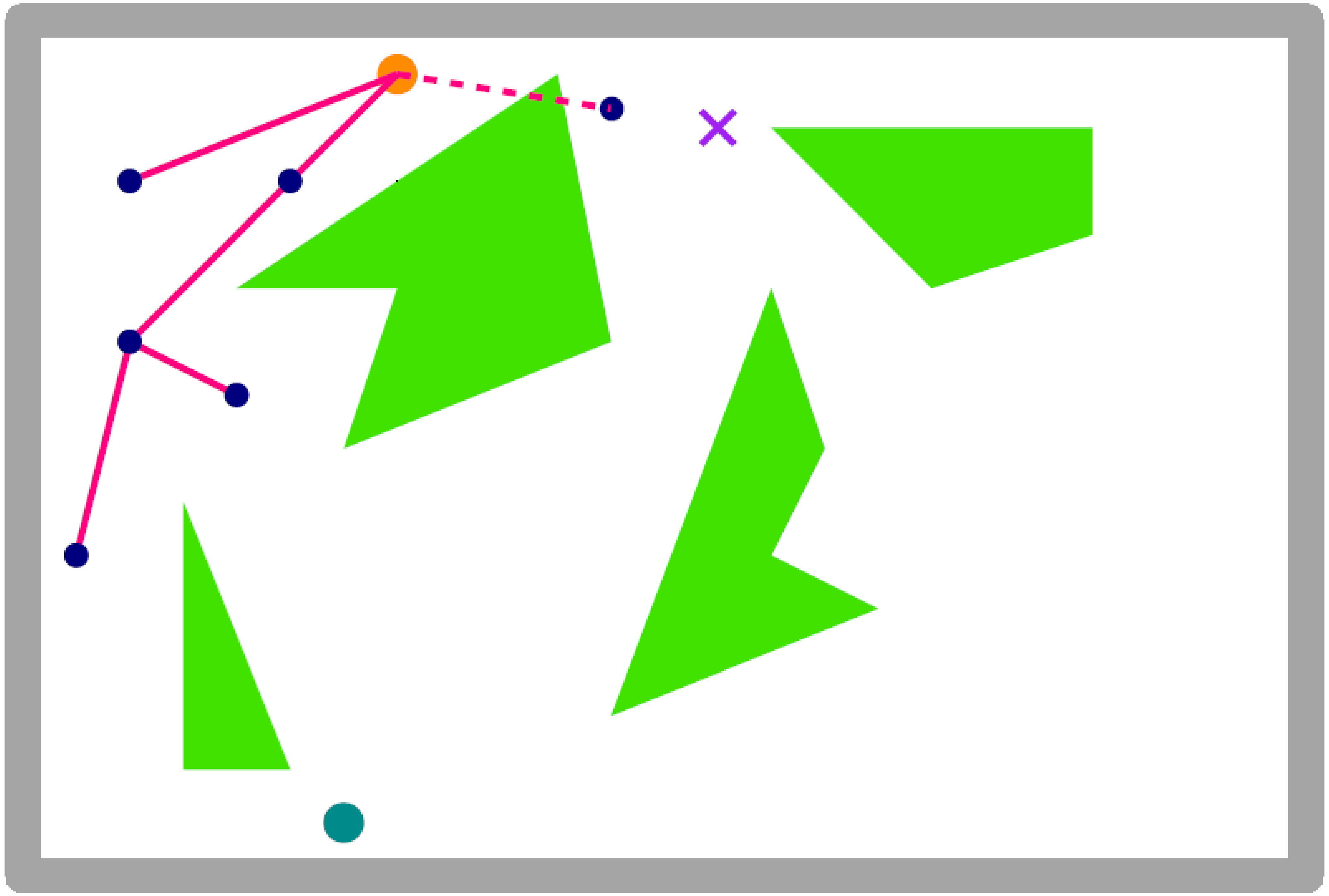
Algorithm 2 $\texttt{RRT}^*$ $(x_{\texttt{init}} := s, x_{\texttt{goal}} := t, n, r, \eta)$

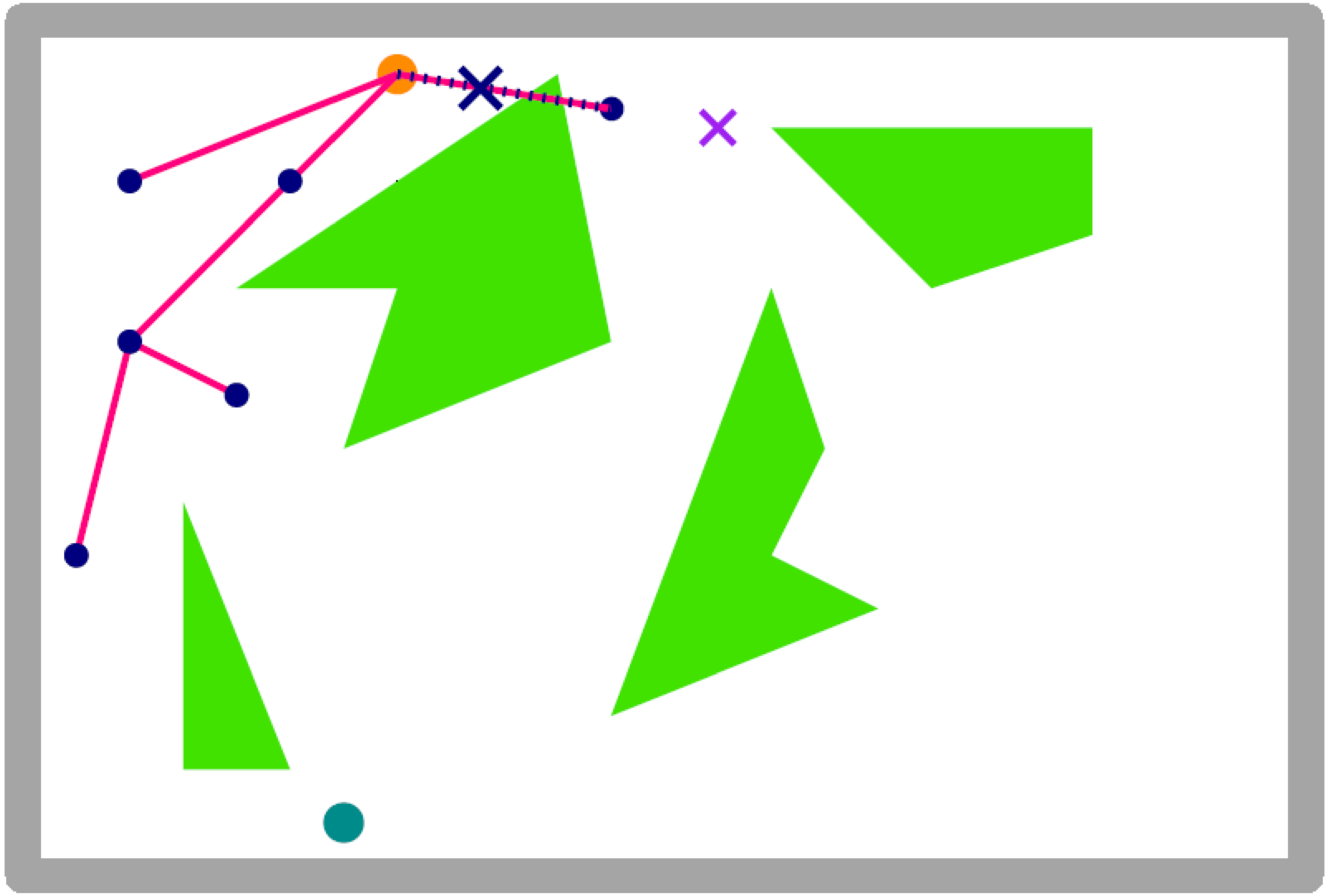
```
1:  $V = \{x_{\text{init}}\}$
2:  **for** $j = 1$ to $n$ **do**
3:      $x_{\text{rand}} \leftarrow$ SAMPLE-FREE( )
4:      $x_{\text{near}} \leftarrow$ NEAREST$(x_{\text{rand}}, V)$
5:      $x_{\text{new}} \leftarrow$ STEER$(x_{\text{near}}, x_{\text{rand}}, \eta)$
6:      **if** COLLISION-FREE$(x_{\text{near}}, x_{\text{new}})$ **then**
7:          $X_{\text{near}} =$ NEAR$(x_{\text{new}}, V, \min\{r(|V|), \eta\})$
8:          $V = V \cup \{x_{\text{new}}\}$
9:          $x_{\text{min}} = x_{\text{near}}$
10:         $c_{\text{min}} =$ COST$(x_{\text{near}}) + \|x_{\text{new}} - x_{\text{near}}\|$
11:         **for** $x_{\text{near}} \in X_{\text{near}}$ **do**
12:             **if** COLLISION-FREE$(x_{\text{near}}, x_{\text{new}})$ **then**
13:                 **if** COST$(x_{\text{near}}) + \|x_{\text{new}} - x_{\text{near}}\| < c_{\text{min}}$ **then**
14:                     $x_{\text{min}} = x_{\text{near}}$
15:                     $c_{\text{min}} =$ COST$(x_{\text{near}}) + \|x_{\text{new}} - x_{\text{near}}\|$
16:         $E = E \cup \{(x_{\text{min}}, x_{\text{new}})\}$
17:         **for** $x_{\text{near}} \in X_{\text{near}}$ **do**
18:             **if** COLLISION-FREE$(x_{\text{new}}, x_{\text{near}})$ **then**
19:                 **if** COST$(x_{\text{new}}) + \|x_{\text{near}} - x_{\text{new}}\| <$ COST$(x_{\text{near}})$
    **then**
20:                     $x_{\text{parent}} =$ PARENT$(x_{\text{near}})$
21:                     $E = E \cup \{(x_{\text{new}}, x_{\text{near}})\} \setminus \{(x_{\text{parent}}, x_{\text{near}})\}$
22: **return** $G = (V, E)$
```

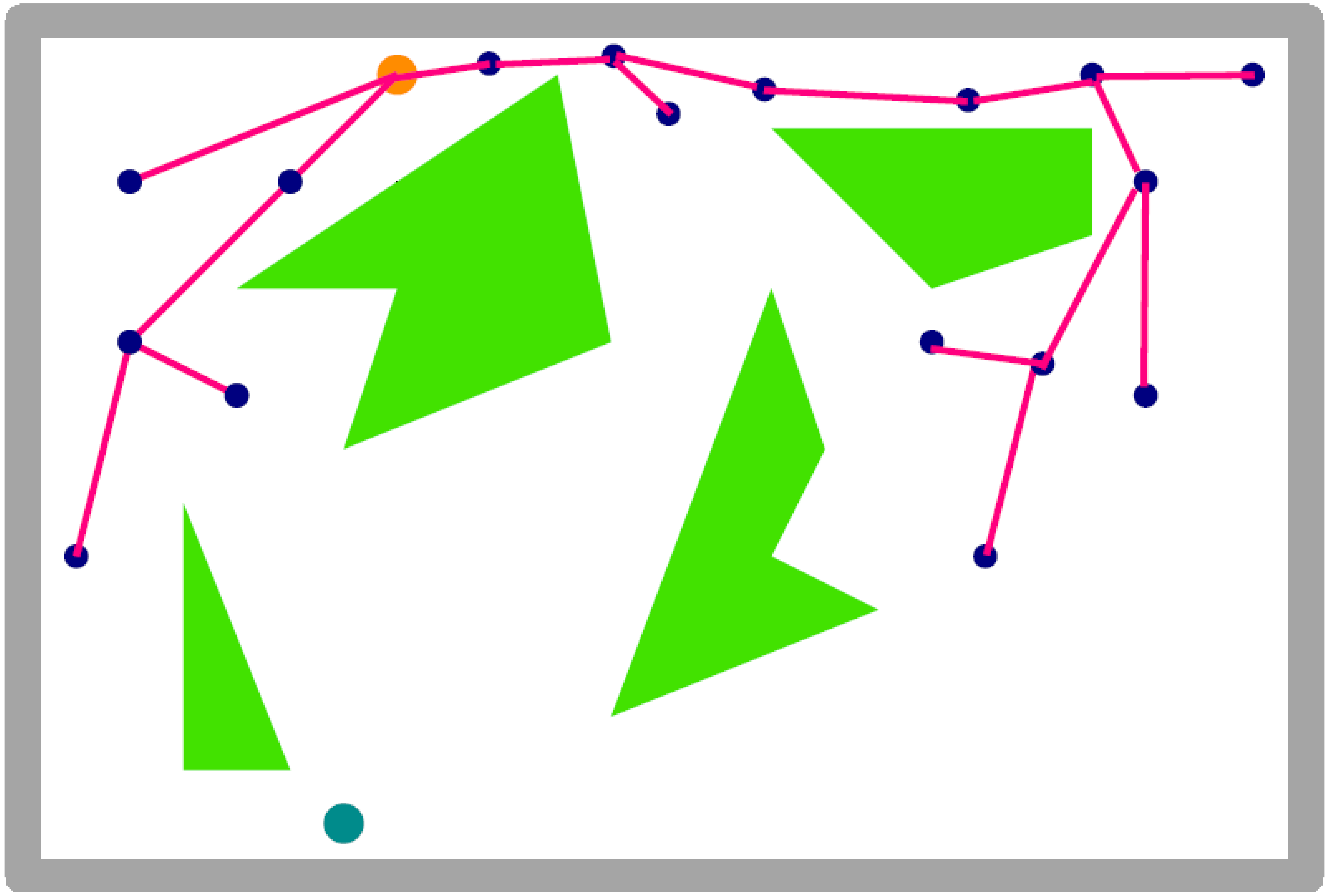


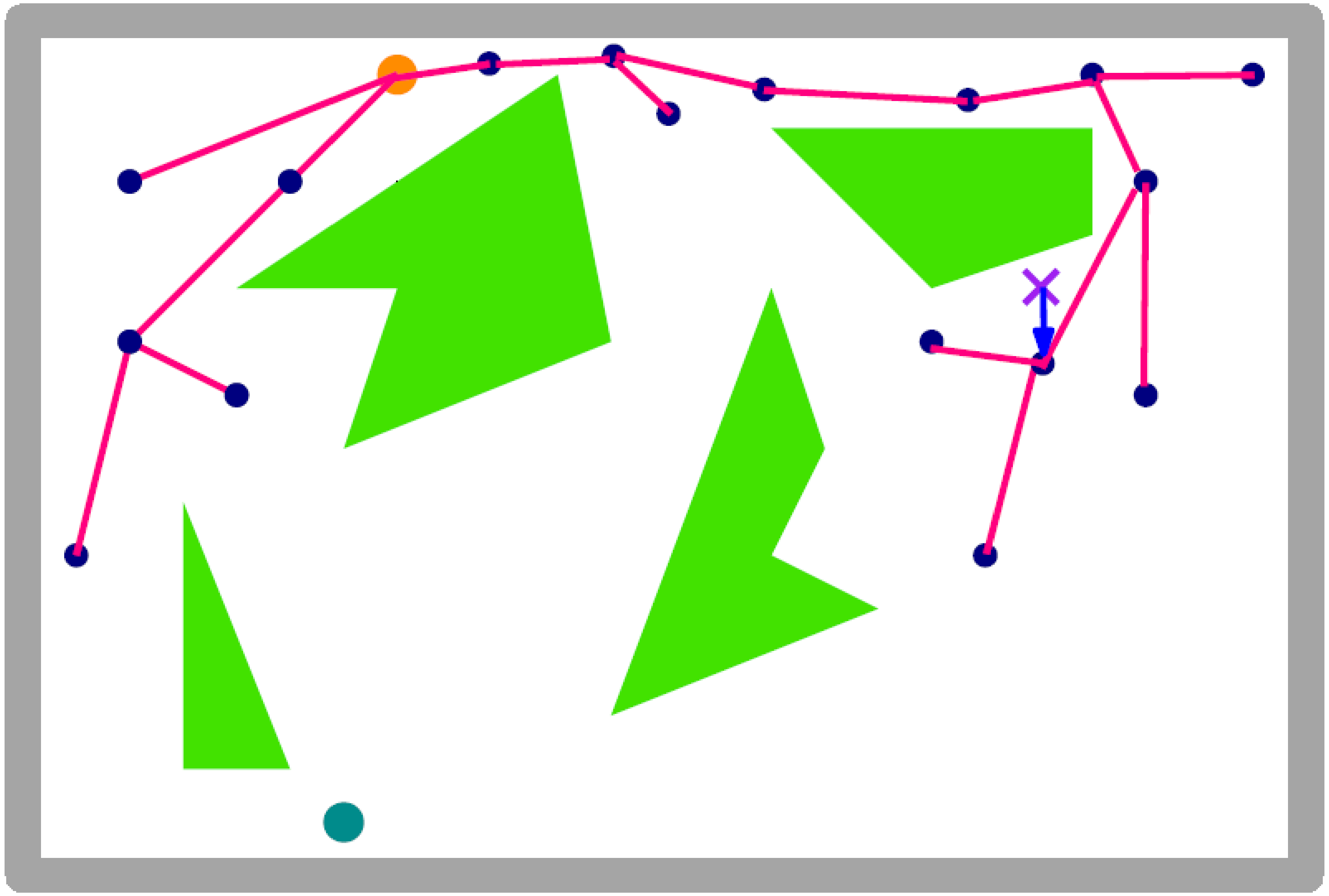


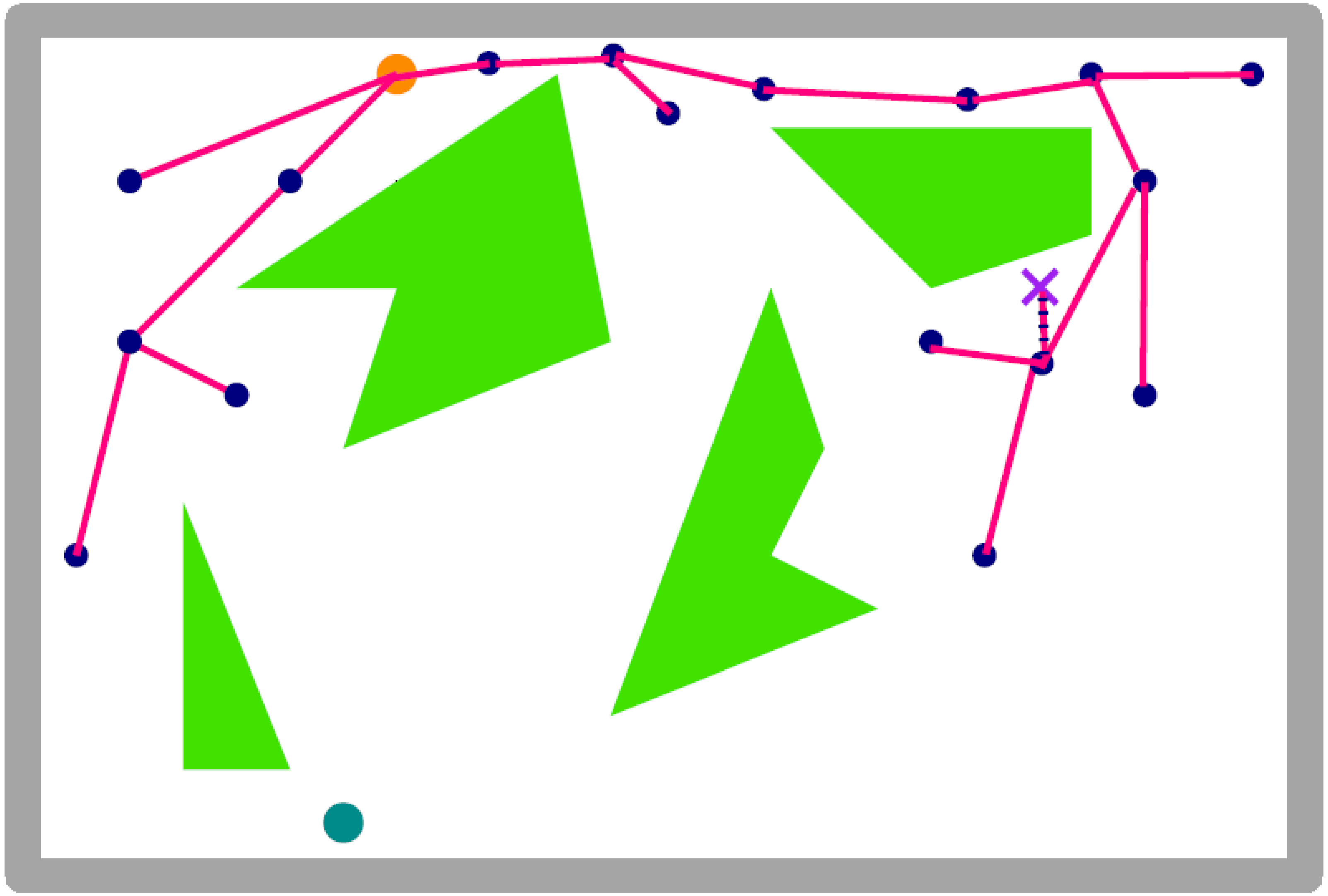


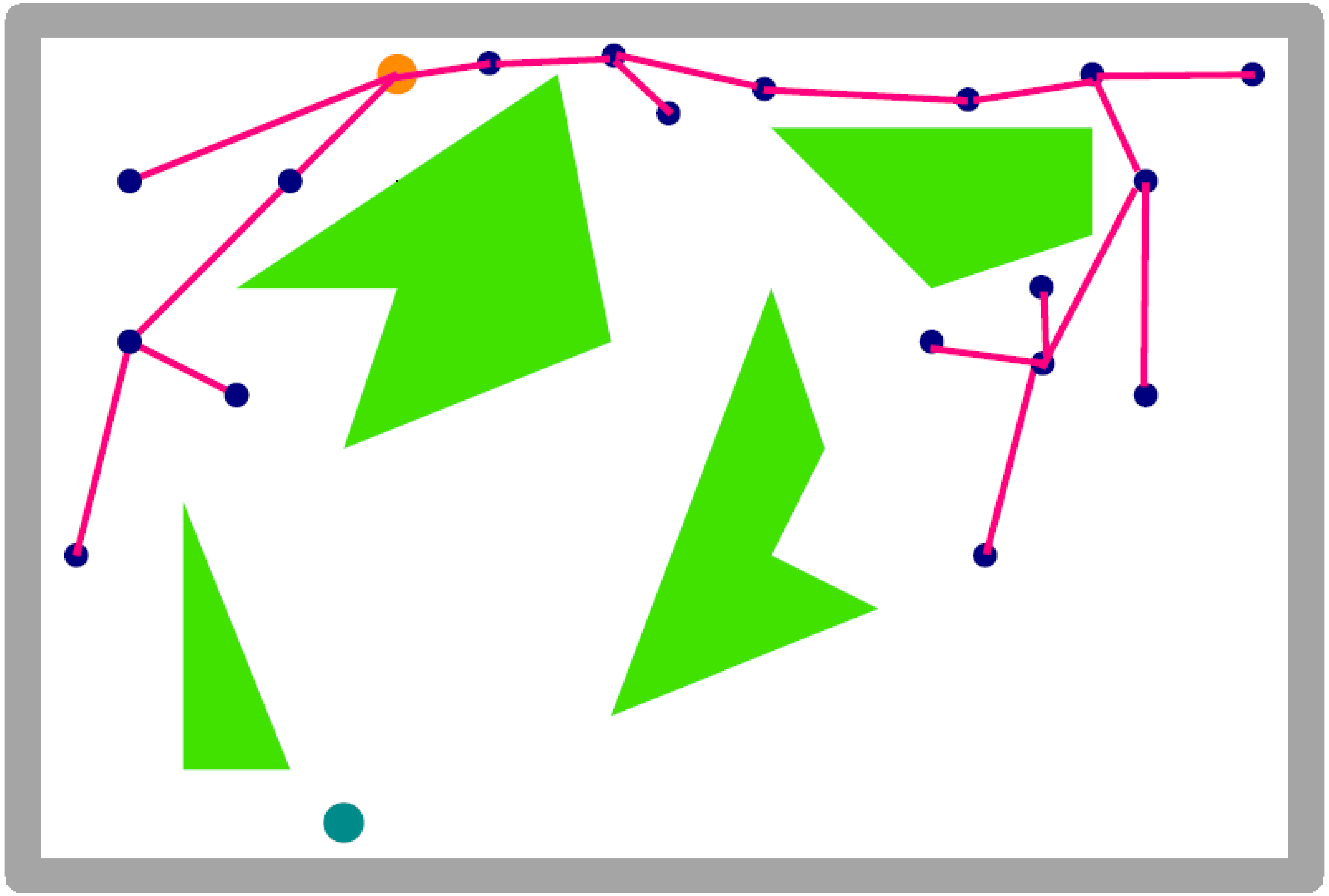


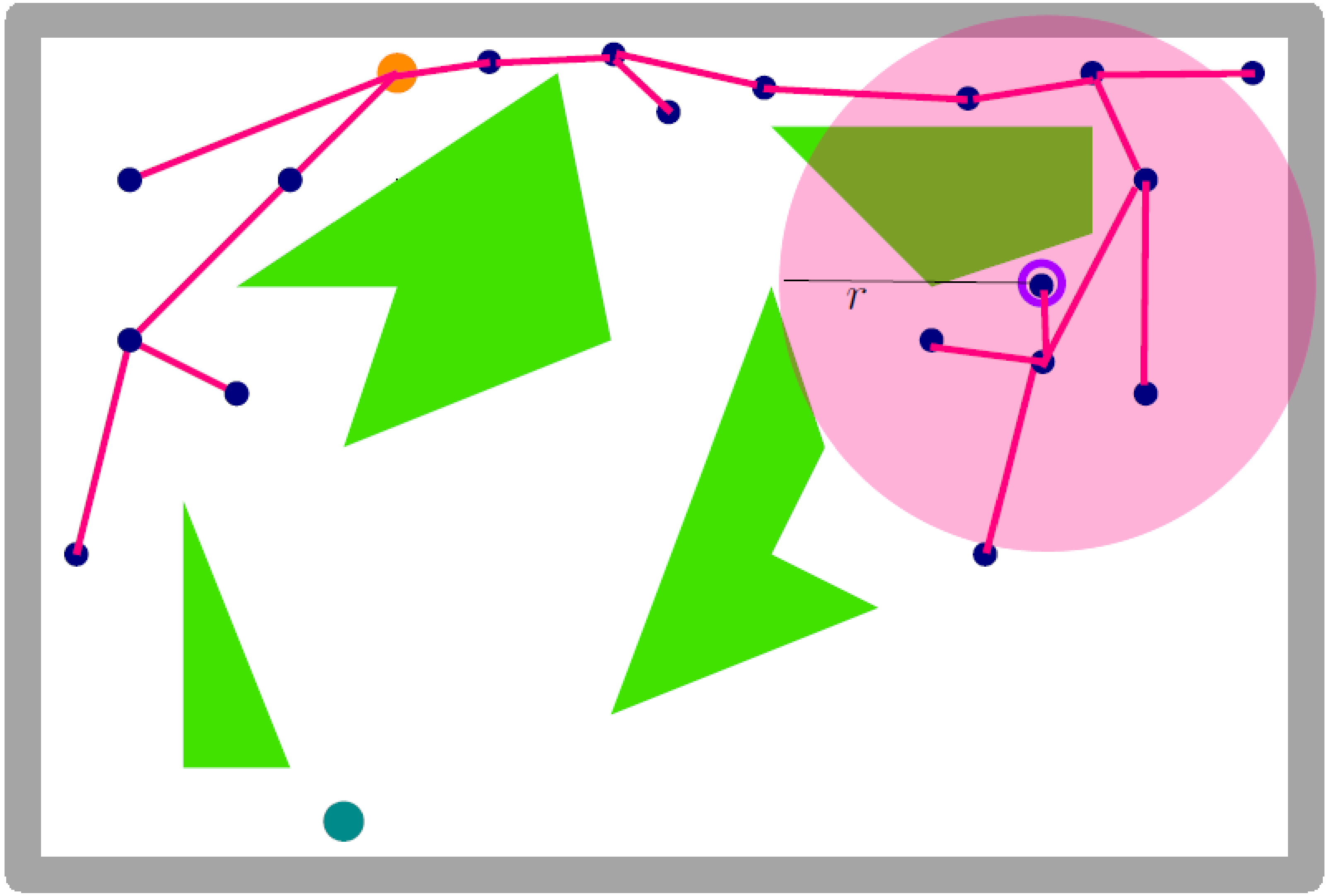


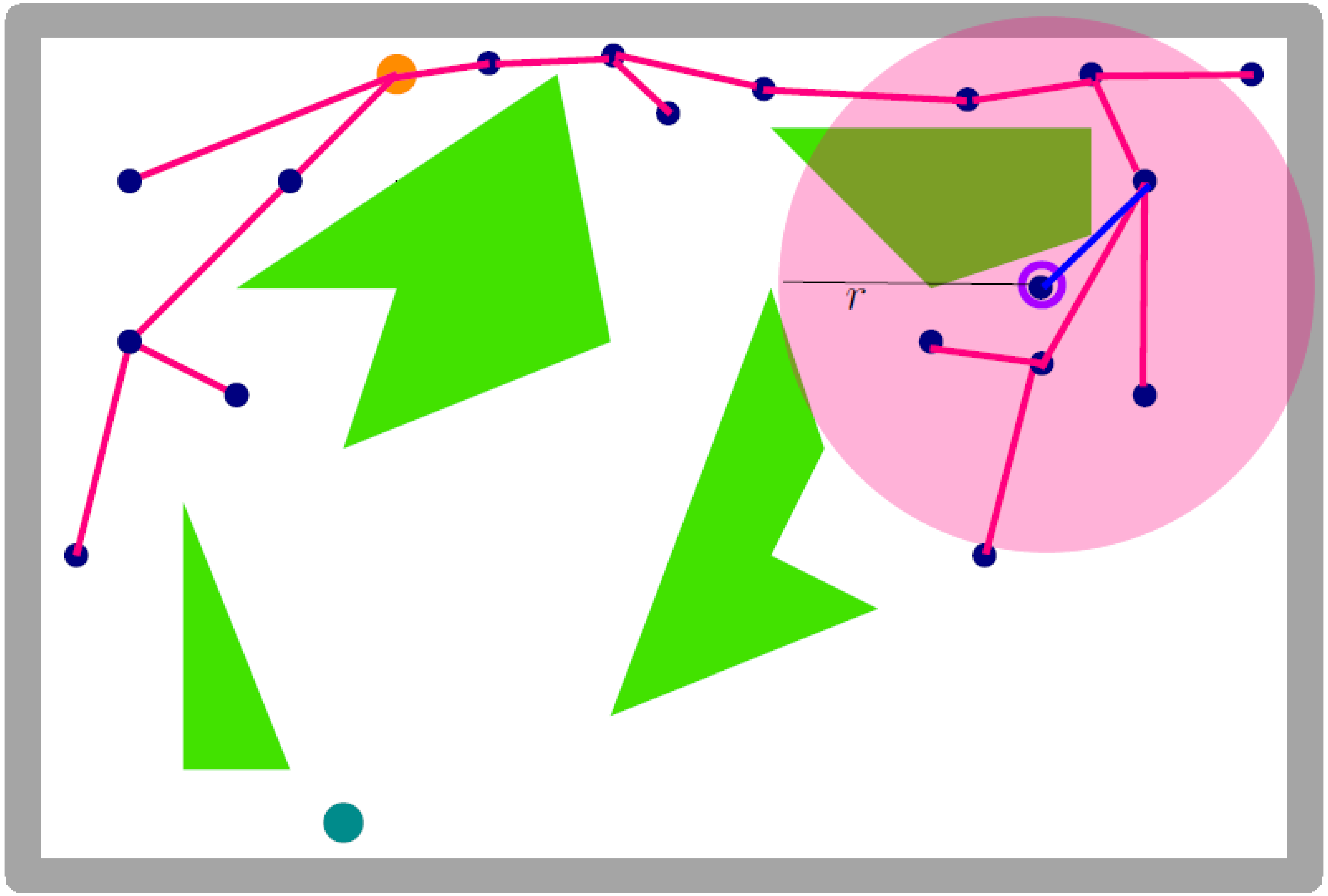


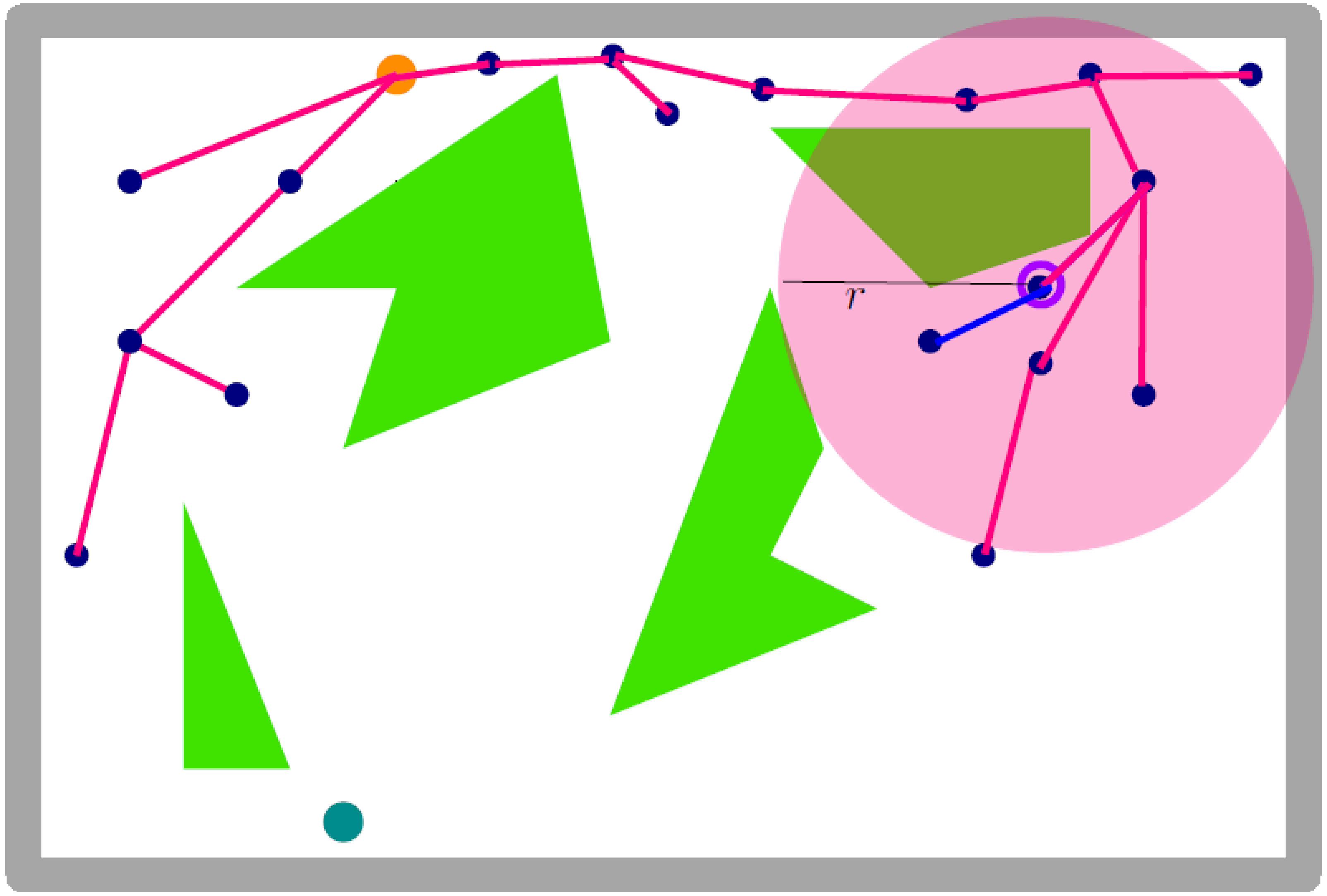







r

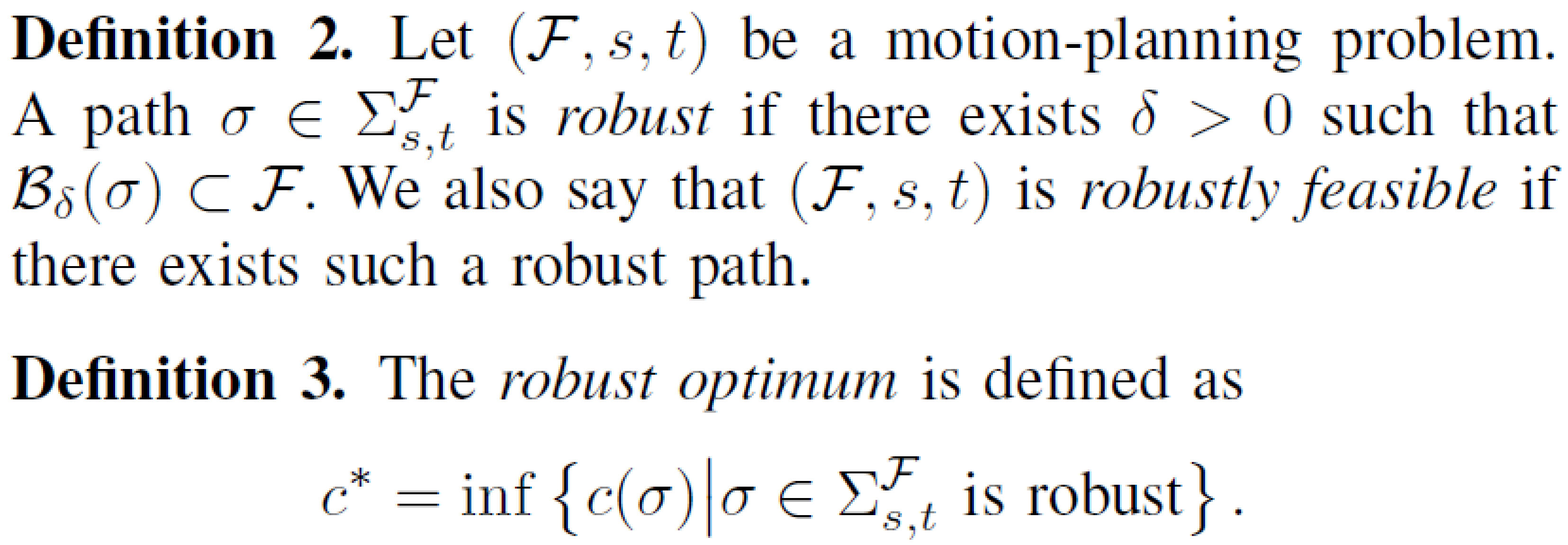

r


r

Asymptotic optimality

- A motion planner is asymptotically optimal if the solution returned by it converges to the optimal solution, as the number of samples tends to infinity

Robustly feasible motion-planning problem



Definition 2. Let $(\mathcal{F}, s, t)$ be a motion-planning problem. A path $\sigma \in \Sigma_{s,t}^{\mathcal{F}}$ is *robust* if there exists $\delta > 0$ such that $\mathcal{B}_\delta(\sigma) \subset \mathcal{F}$. We also say that $(\mathcal{F}, s, t)$ is *robustly feasible* if there exists such a robust path.

Definition 3. The *robust optimum* is defined as

$$c^* = \inf \left\{ c(\sigma) \middle| \sigma \in \Sigma_{s,t}^{\mathcal{F}} \text{ is robust} \right\}.$$

[Solovey et al, 2019]

Asymptotic optimality of RRT*

Theorem 1. *Suppose that* $(\mathcal{F}, s, t)$ *is robustly feasible, fix* $\eta > 0, \varepsilon \in (0,1), \theta \in (0, 1/4), \mu > 0$, *and define the radius of* RRT* *to be*

$$r(n) = \gamma \left(\frac{\log n}{n}\right)^{\frac{1}{d+1}}, \tag{2}$$

$$\gamma > (2+\theta)\left(\frac{(1+\varepsilon/4)c^*}{(d+1)\theta(1-\mu)} \cdot \frac{|\mathcal{F}|}{\zeta_d}\right)^{\frac{1}{d+1}}, \tag{3}$$

where ζ_d *is the volume of a unit* d*-dimensional hypersphere,* c^* *is the robust optimum. Then*

$$\lim_{n\to\infty} \Pr[c(\sigma_n) \le (1+\varepsilon)c^*] = 1.$$

[Solovey et al, 2019]

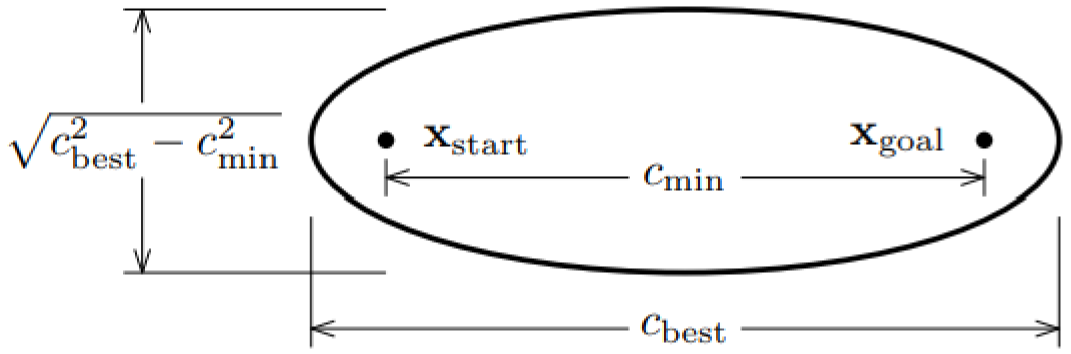
Variants

Speedup: Informed RRT* [Gammell et al., 2014]

- Denote by c_{best} the length of the shortest path found so far
- Based on this knowledge, how can we speedup the convergence of the following iterations?

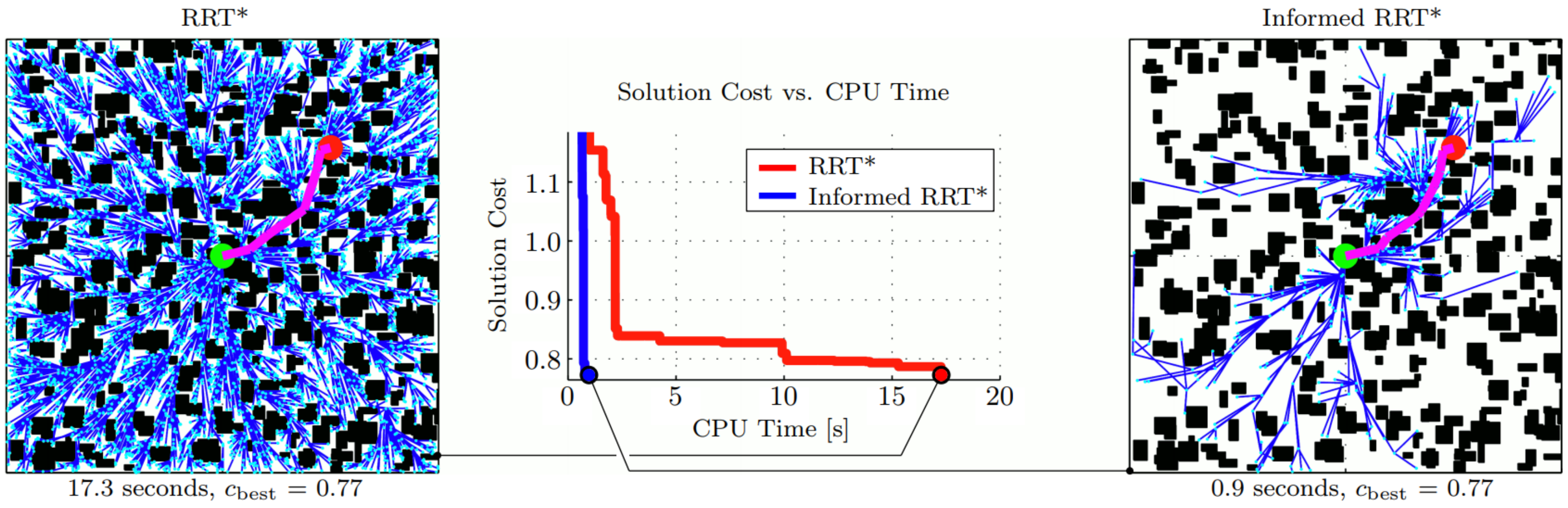
Speedup: Informed RRT*

- Denote by c_{best} the length of the shortest path found so far
- Based on this knowledge, how can we speedup the convergence of the following iterations?
- Observation: any point on a path shorter than c_{best} must lie in a hyperellipsoid:


$\sqrt{c_{\text{best}}^2 - c_{\text{min}}^2}$
$\mathbf{x}_{\text{start}}$
$\mathbf{x}_{\text{goal}}$
c_{min}
c_{best}

Speedup: Informed RRT*

- Improvement:
  - Maintain an ellipsoid E; initially, it covers the whole space
  - Every time an improving solution is found update E
  - Generate new samples only from E
  -



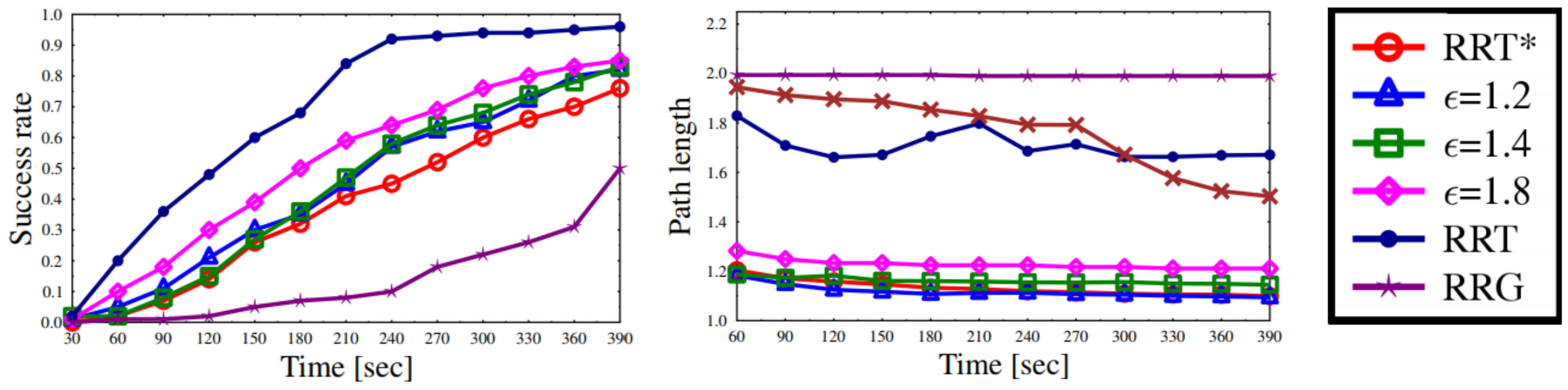
Remark: RRG [Karaman-Frazzoli, 2011]

- Like the RRT* algorithm, but we simply make all the valid (collision free) connections between x_{new} and the nodes in X_{near} , with two edges in opposite direction for each node in X_{near}
- The RRT* tree is a subgraph of RRG
- RRG requires more storage space and is practically more time consuming than RRT*

Tradeoff (speed vs. quality): LBT-RRT

[Salzman-Halperin, 2014]

- Lower-bound RRT:
  - Guarantees convergence to (1+ ε)OPT
    - When ε=0 behaves like RRG
    - When ε=∞ behaves like RRT
  - An edge (v,v’) is collision checked only if it can potentially improve the cost of any vertex on the shortest-path tree rooted in v’ by at least 1+ ε



Further variants

- RRT#
- FMT*
- SST
- many more: see the surveys

References

Papers

Steven M. LaValle, James J. Kuffner Jr.:
Randomized Kinodynamic Planning. ICRA 1999: 473-479

Sertac Karaman, Emilio Frazzoli:
Sampling-based algorithms for optimal motion planning.
I. J. Robotics Res. 30(7): 846-894 (2011)

Kiril Solovey, Lucas Janson, Edward Schmerling, Emilio Frazzoli, Marco Pavone:
Revisiting the Asymptotic Optimality of RRT.* CoRR abs/1909.09688 (2019)

Good starting point

Sampling-based algorithms

Chapter 7 of the book

Principles of robot motion:

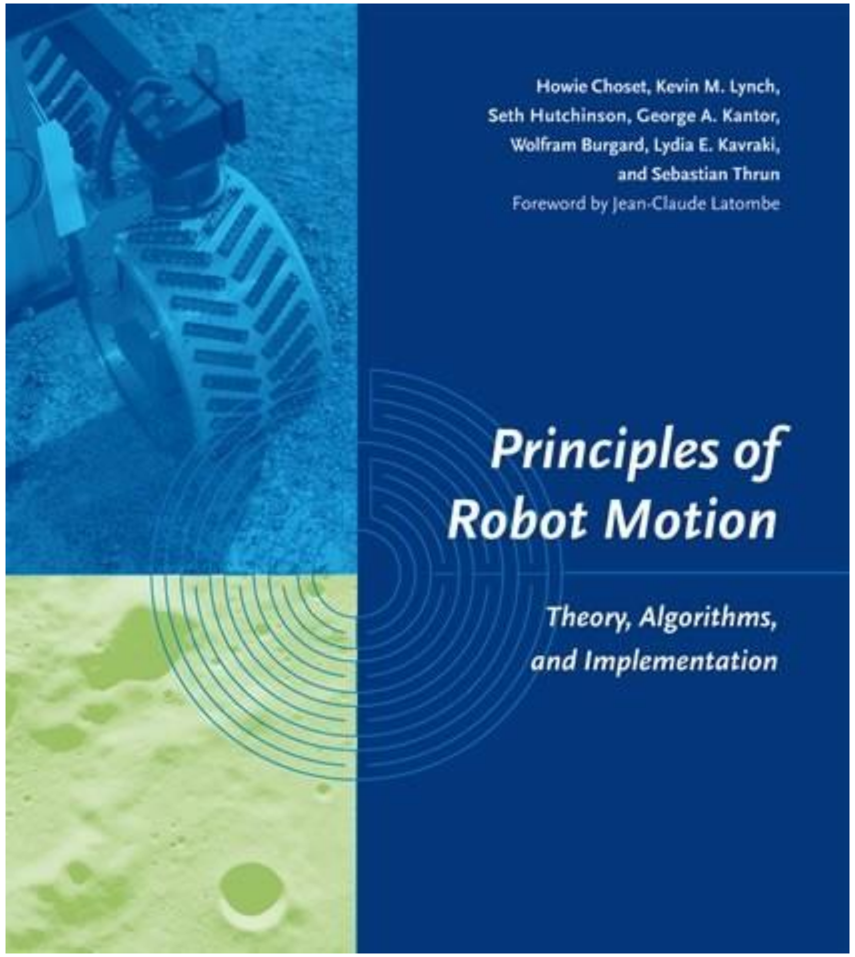
theory, algorithms, and implementation

by Choset et al

The MIT Press

2005

comprehensive survey with many references


Howie Choset, Kevin M. Lynch,
Seth Hutchinson, George A. Kantor,
Wolfram Burgard, Lydia E. Kavraki,
and Sebastian Thrun
Foreword by Jean-Claude Latombe
Principles of Robot Motion
Theory, Algorithms, and Implementation

the book

Planning Algorithms

By Steven LaValle

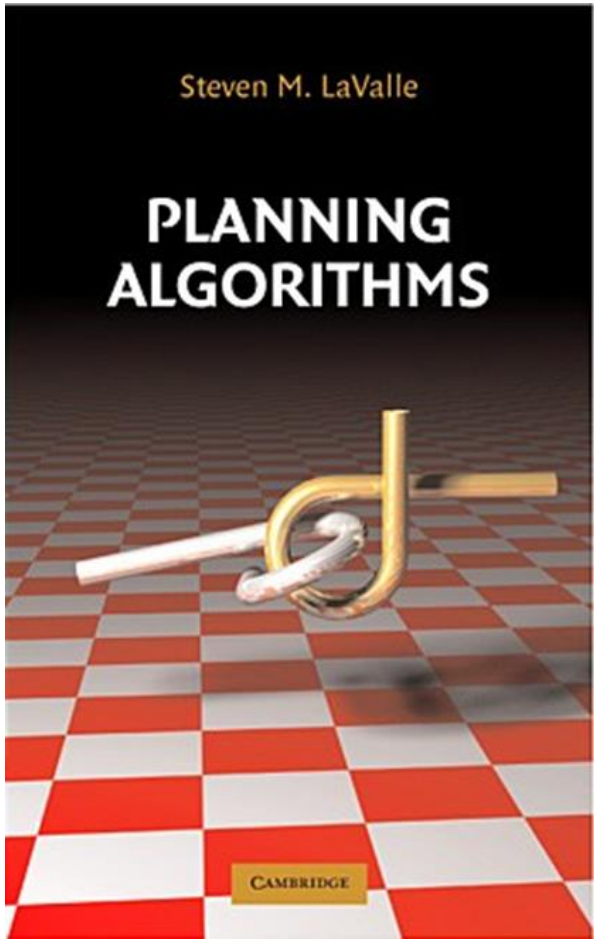
Camrdige University Press, 2006

in-depth coverage of motion planning

available online for free!

http://planning.cs.uiuc.edu/

online bibliography



More recent surveys

- Sampling-Based Robot Motion Planning, Oren Salzman, Communications of the ACM, October 2019
- Robotics, Halperin, Kavraki, Solovey, in Handbook of Computational Geometry, 3rd Edition, 2018
- Sampling-Based Robot Motion Planning: A Review, Elbanhawi and Simic, IEEE Access, 2014 (free online)

THE END