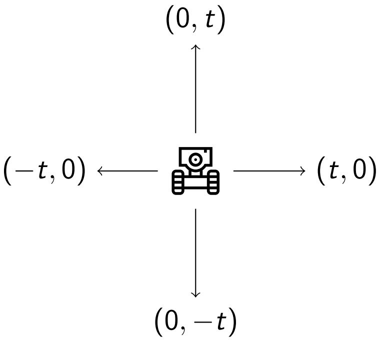
Motion planning in known environment

Viktória Vozárová

December 9, 2018

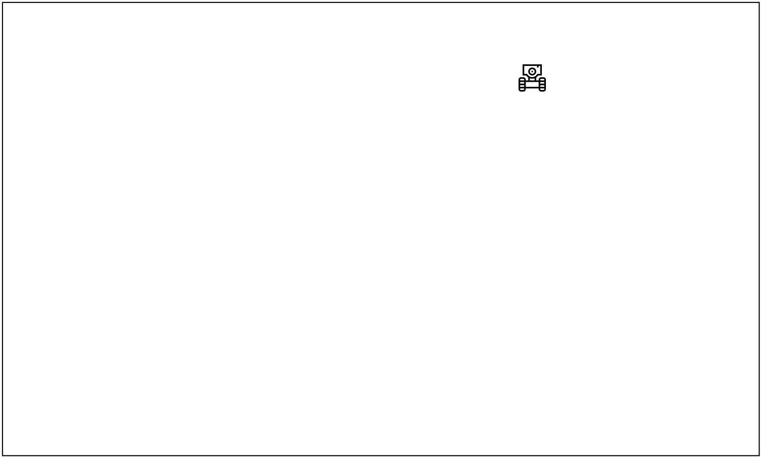
Example – model

- $\mathbb{R}^2$ space with obstacles
- restricted robot movements
- the robot is **nonholonomic**



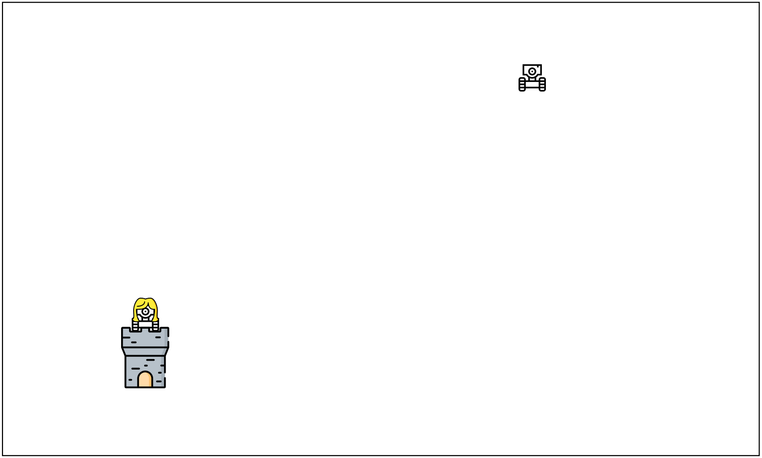
Example – problem

The robot is set in a metric space X on initial position x_{init}.



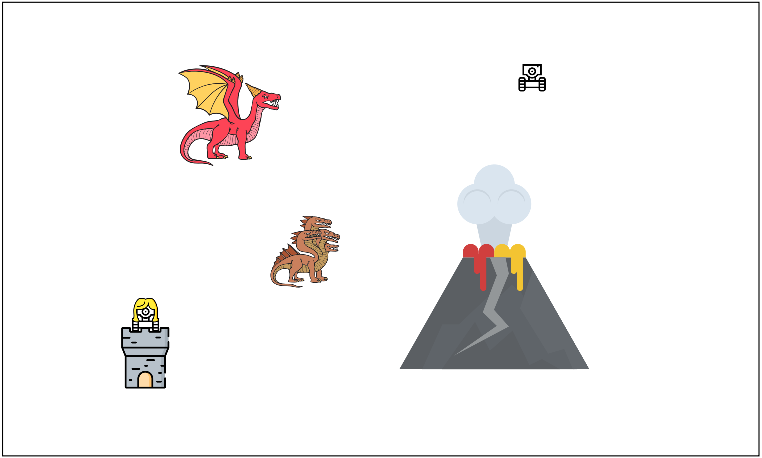
Example – problem

His objective is to reach the roboprincess in $X_{goal} \subset X$.



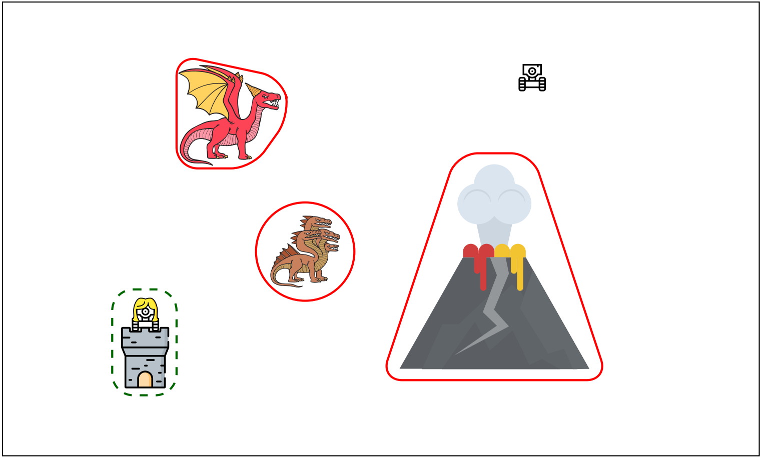
Example – problem

There are some obstacles on the way $X_{obs} \subset X$.



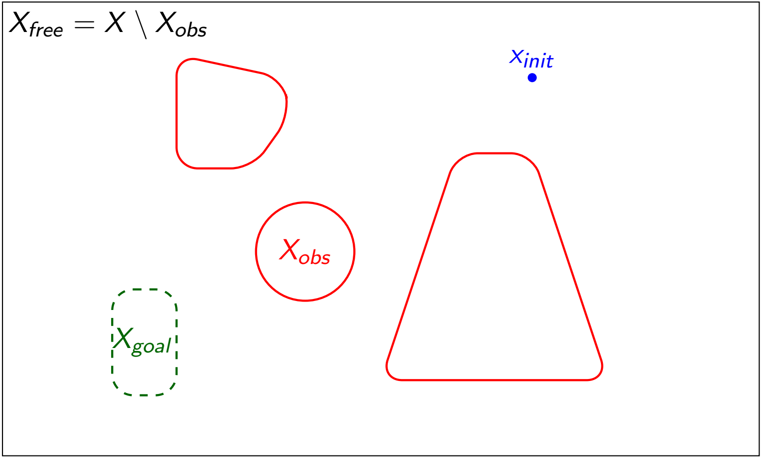
Example – problem

Let's make it more formal.



Example – problem

Let's make it more formal.



RRT – Rapidly-exploring random tree

Initialize tree $T = (\{x_{init}\}, \emptyset)$.

[1] d is a metric of X

RRT – Rapidly-exploring random tree

Initialize tree $T = (\{x_{init}\}, \emptyset)$.

1. Pick random $x_{rand} \in X_{free}$.

[1] d is a metric of X

RRT – Rapidly-exploring random tree

Initialize tree $T = (\{x_{init}\}, \emptyset)$.

1. Pick random $x_{rand} \in X_{free}$.
2. Find the closest point x_{near} to x_{rand} from T.

[1] d is a metric of X

RRT – Rapidly-exploring random tree

Initialize tree $T = (\{x_{init}\}, \emptyset)$.

1. Pick random $x_{rand} \in X_{free}$.
2. Find the closest point x_{near} to x_{rand} from T.
3. Find $x_{new} \in X_{free}$ such that:

---

[1] d is a metric of X

RRT – Rapidly-exploring random tree

Initialize tree $T = (\{x_{init}\}, \emptyset)$.

1. Pick random $x_{rand} \in X_{free}$.
2. Find the closest point x_{near} to x_{rand} from T.
3. Find $x_{new} \in X_{free}$ such that:
   - x_{new} is reachable from x_{near} in Δt,

[1] d is a metric of X

RRT – Rapidly-exploring random tree

Initialize tree $T = (\{x_{init}\}, \emptyset)$.

1. Pick random $x_{rand} \in X_{free}$.
2. Find the closest point x_{near} to x_{rand} from T.
3. Find $x_{new} \in X_{free}$ such that:
   - x_{new} is reachable from x_{near} in Δt,
   - all intermediate points are in X_{free},

[1] d is a metric of X

RRT – Rapidly-exploring random tree

Initialize tree $T = (\{x_{init}\}, \emptyset)$.

1. Pick random $x_{rand} \in X_{free}$.
2. Find the closest point x_{near} to x_{rand} from T.
3. Find $x_{new} \in X_{free}$ such that:
   - x_{new} is reachable from x_{near} in Δt,
   - all intermediate points are in X_{free},
   - $d(x_{new}, x_{rand})$ is minimal.[1]

[1] d is a metric of X

RRT – Rapidly-exploring random tree

Initialize tree $T = (\{x_{init}\}, \emptyset)$.

1. Pick random $x_{rand} \in X_{free}$.
2. Find the closest point x_{near} to x_{rand} from T.
3. Find $x_{new} \in X_{free}$ such that:
   - x_{new} is reachable from x_{near} in Δt,
   - all intermediate points are in X_{free},
   - $d(x_{new}, x_{rand})$ is minimal.[1]
4. Add the vertex x_{new} and the edge (x_{near}, x_{new}) to T.

[1] d is a metric of X

RRT – Rapidly-exploring random tree

Initialize tree $T = (\{x_{init}\}, \emptyset)$.

1. Pick random $x_{rand} \in X_{free}$.
2. Find the closest point x_{near} to x_{rand} from T.
3. Find $x_{new} \in X_{free}$ such that:
   - x_{new} is reachable from x_{near} in Δt,
   - all intermediate points are in X_{free},
   - $d(x_{new}, x_{rand})$ is minimal.[1]
4. Add the vertex x_{new} and the edge (x_{near}, x_{new}) to T.
5. Repeat until $x_{new} \in X_{goal}$.

---

[1] d is a metric of X

RRT – Rapidly-exploring random tree

Initialize tree $T = (\{x_{init}\}, \emptyset)$.

1. Pick random $x_{rand} \in X_{free}$.
2. Find the closest point x_{near} to x_{rand} from T.
3. Find $x_{new} \in X_{free}$ such that:
   - x_{new} is reachable from x_{near} in Δt,
   - all intermediate points are in X_{free},
   - $d(x_{new}, x_{rand})$ is minimal.[1]
4. Add the vertex x_{new} and the edge (x_{near}, x_{new}) to T.
5. Repeat until $x_{new} \in X_{goal}$.

[1] d is a metric of X

RRT – Rapidly-exploring random tree

Initialize tree $T = (\{x_{init}\}, \emptyset)$.

1. Pick random $x_{rand} \in X_{free}$.
2. Find the closest point x_{near} to x_{rand} from T.
3. Find $x_{new} \in X_{free}$ such that:
   - x_{new} is reachable from x_{near} in Δt,
   - all intermediate points are in X_{free},
   - $d(x_{new}, x_{rand})$ is minimal.[1]
4. Add the vertex x_{new} and the edge (x_{near}, x_{new}) to T.
5. Repeat until $x_{new} \in X_{goal}$.

Return path from x_{init} to $x \in X_{goal}$.

[1] d is a metric of X

RRT*

Initialize tree $T = (\{x_{init}\}, \emptyset)$.

[2] c is a cost function

RRT*

Initialize tree $T = (\{x_{init}\}, \emptyset)$.

1. Pick random $x_{rand} \in X_{free}$.

[2] c is a cost function

RRT*

Initialize tree $T = (\{x_{init}\}, \emptyset)$.

1. Pick random $x_{rand} \in X_{free}$.
2. Find the closest point x_{near} to x_{rand} from T.

[2] c is a cost function

RRT*

Initialize tree $T = (\{x_{init}\}, \emptyset)$.

1. Pick random $x_{rand} \in X_{free}$.
2. Find the closest point x_{near} to x_{rand} from T.
3. Find $x_{new} \in X_{free}$ as in RRT.

[2] c is a cost function

RRT*

Initialize tree $T = (\{x_{init}\}, \emptyset)$.

1. Pick random $x_{rand} \in X_{free}$.
2. Find the closest point x_{near} to x_{rand} from T.
3. Find $x_{new} \in X_{free}$ as in RRT.
4. Find x_{parent} in T such that:

[2] c is a cost function

RRT*

Initialize tree $T = (\{x_{init}\}, \emptyset)$.

1. Pick random $x_{rand} \in X_{free}$.
2. Find the closest point x_{near} to x_{rand} from T.
3. Find $x_{new} \in X_{free}$ as in RRT.
4. Find x_{parent} in T such that:
   - x_{new} is reachable from x_{parent} in Δt,

[2] c is a cost function

RRT*

Initialize tree $T = (\{x_{init}\}, \emptyset)$.

1. Pick random $x_{rand} \in X_{free}$.
2. Find the closest point x_{near} to x_{rand} from T.
3. Find $x_{new} \in X_{free}$ as in RRT.
4. Find x_{parent} in T such that:
   - x_{new} is reachable from x_{parent} in Δt,
   - the path from x_{init} to x_{parent} + $c(x_{parent}, x_{new})$ is minimal.[2]

[2] c is a cost function

RRT*

Initialize tree $T = (\{x_{init}\}, \emptyset)$.

1. Pick random $x_{rand} \in X_{free}$.
2. Find the closest point x_{near} to x_{rand} from T.
3. Find $x_{new} \in X_{free}$ as in RRT.
4. Find x_{parent} in T such that:
   - x_{new} is reachable from x_{parent} in Δt,
   - the path from x_{init} to $x_{parent} + c(x_{parent}, x_{new})$ is minimal.[2]
5. Add the vertex x_{new} and the edge (x_{parent}, x_{new}) to T.

[2] c is a cost function

RRT*

Initialize tree $T = (\{x_{init}\}, \emptyset)$.

1. Pick random $x_{rand} \in X_{free}$.
2. Find the closest point x_{near} to x_{rand} from T.
3. Find $x_{new} \in X_{free}$ as in RRT.
4. Find x_{parent} in T such that:
   - x_{new} is reachable from x_{parent} in Δt,
   - the path from x_{init} to $x_{parent} + c(x_{parent}, x_{new})$ is minimal.[2]
5. Add the vertex x_{new} and the edge (x_{parent}, x_{new}) to T.
6. For each x in T where $d(x, x_{new}) < \varepsilon$:

[2] c is a cost function

RRT*

Initialize tree $T = (\{x_{init}\}, \emptyset)$.

1. Pick random $x_{rand} \in X_{free}$.
2. Find the closest point x_{near} to x_{rand} from T.
3. Find $x_{new} \in X_{free}$ as in RRT.
4. Find x_{parent} in T such that:
   - x_{new} is reachable from x_{parent} in Δt,
   - the path from x_{init} to x_{parent} + $c(x_{parent}, x_{new})$ is minimal.[2]
5. Add the vertex x_{new} and the edge (x_{parent}, x_{new}) to T.
6. For each x in T where $d(x, x_{new}) < \varepsilon$:
   - if the path from x_{init} to x can be improved by changing x's parent to x_{new}, rewire.

[2] c is a cost function

RRT*

Initialize tree $T = (\{x_{init}\}, \emptyset)$.

1. Pick random $x_{rand} \in X_{free}$.
2. Find the closest point x_{near} to x_{rand} from T.
3. Find $x_{new} \in X_{free}$ as in RRT.
4. Find x_{parent} in T such that:
   - x_{new} is reachable from x_{parent} in Δt,
   - the path from x_{init} to $x_{parent} + c(x_{parent}, x_{new})$ is minimal.[2]
5. Add the vertex x_{new} and the edge (x_{parent}, x_{new}) to T.
6. For each x in T where $d(x, x_{new}) < \varepsilon$:
   - if the path from x_{init} to x can be improved by changing x's parent to x_{new}, rewire.
7. Repeat until $x_{new} \in X_{goal}$.

[2] c is a cost function

RRT*

Initialize tree $T = (\{x_{init}\}, \emptyset)$.

1. Pick random $x_{rand} \in X_{free}$.
2. Find the closest point x_{near} to x_{rand} from T.
3. Find $x_{new} \in X_{free}$ as in RRT.
4. Find x_{parent} in T such that:
   - x_{new} is reachable from x_{parent} in Δt,
   - the path from x_{init} to $x_{parent} + c(x_{parent}, x_{new})$ is minimal.[2]
5. Add the vertex x_{new} and the edge (x_{parent}, x_{new}) to T.
6. For each x in T where $d(x, x_{new}) < \varepsilon$:
   - if the path from x_{init} to x can be improved by changing x's parent to x_{new}, rewire.
7. Repeat until $x_{new} \in X_{goal}$.

[2] c is a cost function

RRT*

Initialize tree $T = (\{x_{init}\}, \emptyset)$.

1. Pick random $x_{rand} \in X_{free}$.
2. Find the closest point x_{near} to x_{rand} from T.
3. Find $x_{new} \in X_{free}$ as in RRT.
4. Find x_{parent} in T such that:
   - x_{new} is reachable from x_{parent} in Δt,
   - the path from x_{init} to $x_{parent} + c(x_{parent}, x_{new})$ is minimal.[2]
5. Add the vertex x_{new} and the edge (x_{parent}, x_{new}) to T.
6. For each x in T where $d(x, x_{new}) < \varepsilon$:
   - if the path from x_{init} to x can be improved by changing x's parent to x_{new}, rewire.
7. Repeat until $x_{new} \in X_{goal}$.

Return path from x_{init} to $x \in X_{goal}$.

---

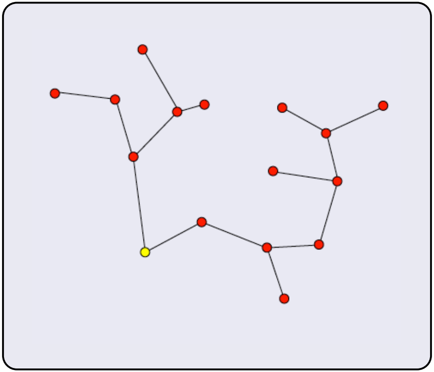
[2] c is a cost function

PRM, RRT, and RRT*

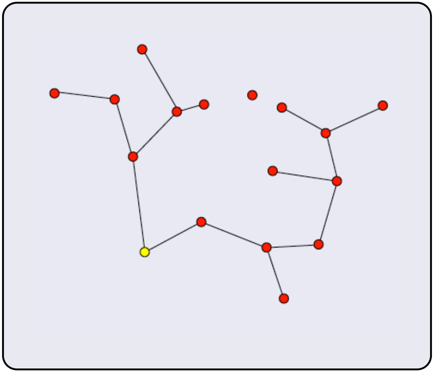
Frank Dellaert

Based on materials by Steve Lavalle, Lydia Kavraki, Emilio Frazzoli and their students

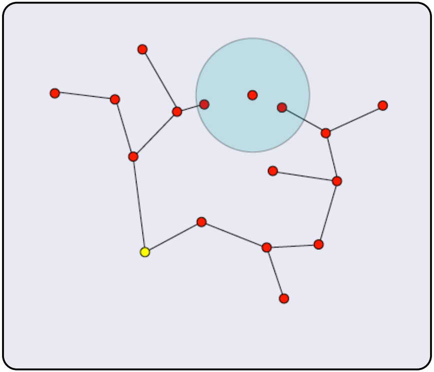
RRT*



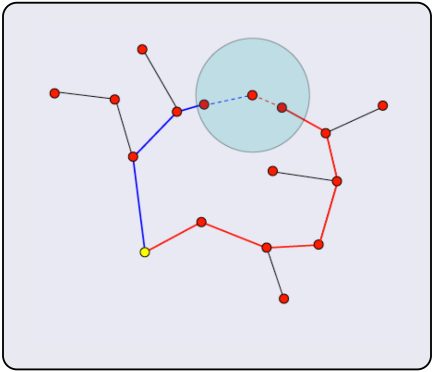
RRT*



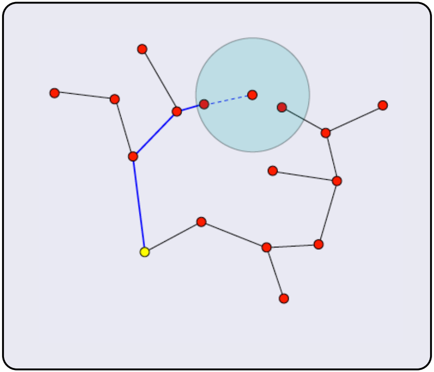
RRT*



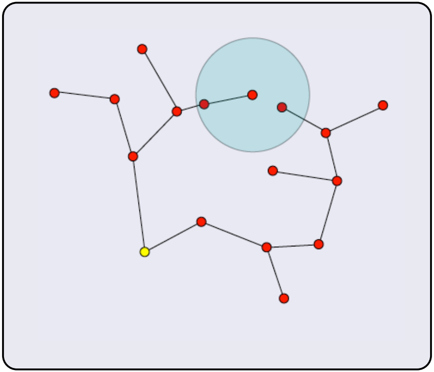
RRT*



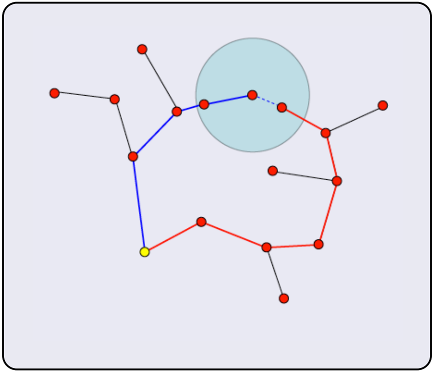
RRT*



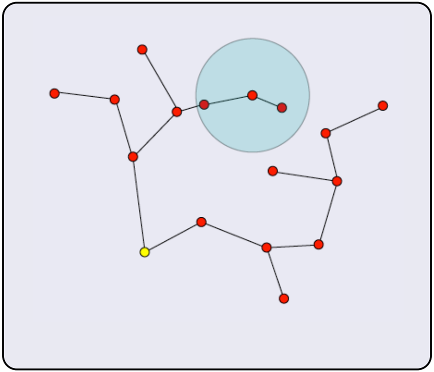
RRT*



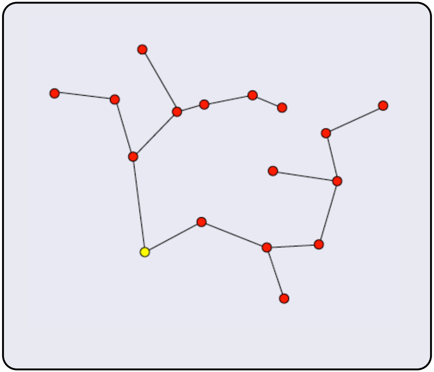
RRT*



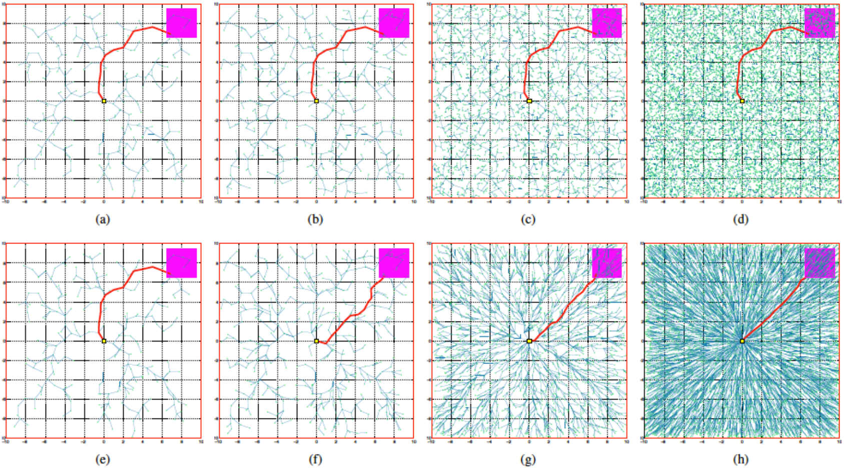
RRT*



RRT*



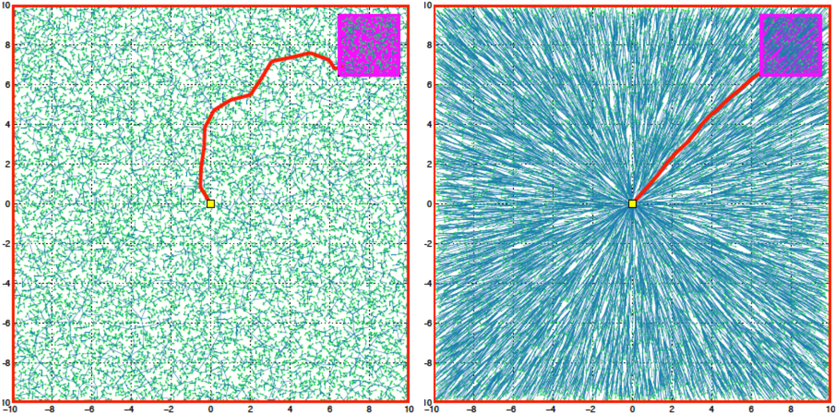
RRT vs RRT*



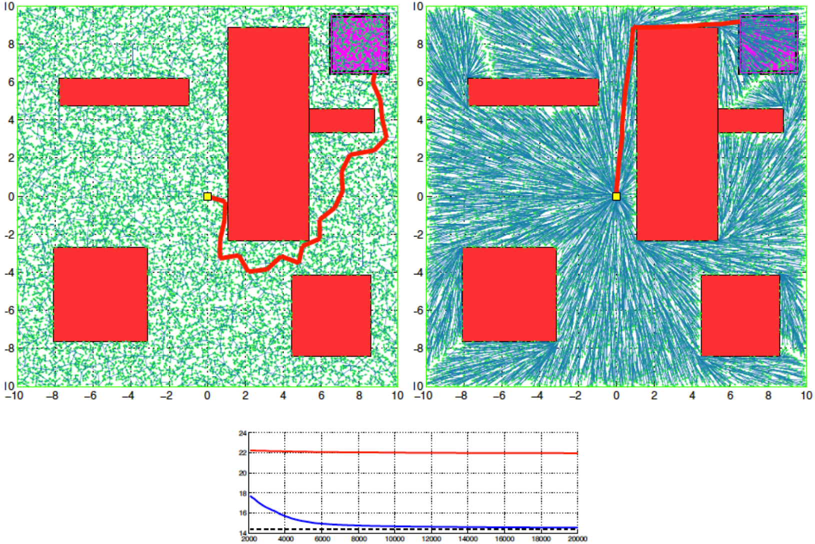
(a) (b) (c) (d)

(e) (f) (g) (h)

RRT vs RRT*



RRT* with Obstacles



Anytime RRT*

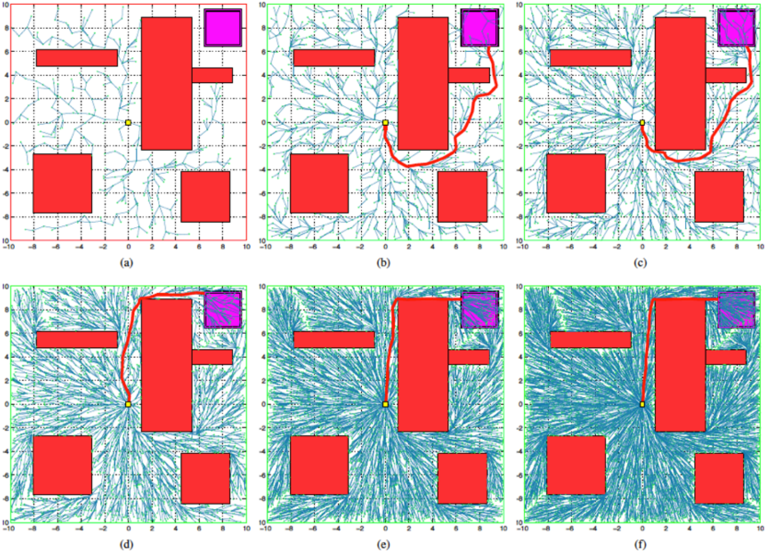


Fig. 4. RRT* algorithm shown after 500 (a), 1,500 (b), 2,500 (c), 5,000 (d), 10,000 (e), 15,000 (f) iterations.

Other heuristics

- Bidirectional RRT
- Rapidly-exploring random graph (RRG)
- Informed RRT*
- RRT*-Smart
- Real-Time RRT*