16-350
Planning Techniques for Robotics

Planning Representations:
Probabilistic Roadmaps for Continuous Spaces

Maxim Likhachev
Robotics Institute
Carnegie Mellon University
Example

- Planning for manipulation
Example

- Planning for manipulation
Example

- Planning for manipulation
  - robot state is defined by joint angles \( Q = \{q_1, \ldots, q_6\} \)
  - need to find a (least-cost) motion that connects \( Q_{\text{start}} \) to \( Q_{\text{goal}} \)

Constraints?
Example

• Planning for manipulation
  – robot state is defined by joint angles $Q = \{q_1, \ldots, q_6\}$
  – need to find a (least-cost) motion that connects $Q_{\text{start}}$ to $Q_{\text{goal}}$
  – Constraints:
    • All joint angles should be within corresponding joint limits
    • No collisions with obstacles and no self-collisions
Planning for manipulation

- robot state is defined by joint angles $Q = \{q_1, ..., q_6\}$
- need to find a (least-cost) motion that connects $Q_{\text{start}}$ to $Q_{\text{goal}}$
- Constraints:
  - All joint angles should be within corresponding joint limits
  - No collisions with obstacles and no self-collisions

Can we use a grid-based representation for planning?
Resolution Complete vs. Sampling-based Planning

- Resolution complete planning (e.g. Grid-based):
  - generate a systematic (uniform) representation (graph) of a free C-space ($C_{\text{free}}$)
  - search the generated representation for a solution guaranteeing to find it if one exists (completeness)
  - can interleave the construction of the representation with the search (i.e., construct only what is necessary)

The example above is borrowed from “AI: A Modern Approach” by S. Russell & P. Norvig
Resolution Complete vs. Sampling-based Planning

- Resolution complete planning (e.g. Grid-based):
  - complete and provide sub-optimality bounds on the solution

[Diagram showing resolution complete planning example]

*the example above is borrowed from “AI: A Modern Approach” by S. Russell & P. Norvig*
Resolution Complete vs. Sampling-based Planning

- Resolution complete planning (e.g. Grid-based):
  - complete and provide sub-optimality bounds on the solution
  - can get computationally very expensive, especially in high-D

the example above is borrowed from "AI: A Modern Approach" by S. Russell & P. Norvig
Resolution Complete vs. Sampling-based Planning

- Sampling-based planning:

  Main observation:
  The space is continuous and rather benign!

the example above is borrowed from “AI: A Modern Approach” by S. Russell & P. Norvig
Resolution Complete vs. Sampling-based Planning

- **Sampling-based planning:**
  - generate a sparse (sample-based) representation (graph) of a free C-space \(C_{\text{free}}\)
  - search the generated representation for a solution

The example above is borrowed from “AI: A Modern Approach” by S. Russell & P. Norvig.
Resolution Complete vs. Sampling-based Planning

- Sampling-based planning:
  - provide **probabilistic** completeness guarantees
    - guaranteed to find a solution, if one exists, but only in the limit of the number of samples (that is, only as the number of samples approaches infinity)
  - well-suited for high-dimensional planning

---

the example above is borrowed from “AI: A Modern Approach” by S. Russell & P. Norvig
Main Questions in Sampling-based Planning

• How to select samples to construct a “good” graph

• How to search the graph

• Can we interleave these steps
Probabilistic Roadmaps (PRMs)

Step 1. Preprocessing Phase: Build a roadmap (graph) $G$ which, hopefully, should be accessible from any point in $C_{free}$

Step 2. Query Phase: Given a start configuration $q_I$ and goal configuration $q_G$, connect them to the roadmap $G$ using a local planner, and then search the augmented roadmap for a shortest path from $q_I$ to $q_G$
Probabilistic Roadmaps (PRMs)

**Step 1. Preprocessing Phase:** Build a roadmap (graph) $G$ which, hopefully, should be accessible from any point in $C_{free}$

**Step 2. Query Phase:** Given a start configuration $q_I$ and goal configuration $q_G$, connect them to the roadmap $G$ using a local planner, and then search the augmented roadmap for a shortest path from $q_I$ to $q_G$.

*Any ideas for the local planner?*
Probabilistic Roadmaps (PRMs)

Step 1. Preprocessing Phase: Build a roadmap (graph) $\mathcal{G}$ which, hopefully, should be accessible from any point in $C_{\text{free}}$.

Step 2. Query Phase: Given a start configuration $q_I$ and goal configuration $q_G$, connect them to the roadmap $\mathcal{G}$ using a local planner, and then search the augmented roadmap for a shortest path from $q_I$ to $q_G$.

Any ideas for the local planner?

Can be as simple as a straight line (interpolation) connecting start (or goal) configuration to the nearest vertex in the roadmap.
Probabilistic Roadmaps (PRMs)

Step 1: Preprocessing Phase.

BUILD_ROADMAP
1 $\mathcal{G}.\text{init}(); i \leftarrow 0;$
2 while $i < N$
3 \hspace{1em} if $\alpha(i) \in \mathcal{C}_{\text{free}}$ then
4 \hspace{1em} $\mathcal{G}.\text{add\_vertex}(\alpha(i)); i \leftarrow i + 1;$
5 \hspace{1em} for each $q \in \text{NEIGHBORHOOD}(\alpha(i), \mathcal{G})$
6 \hspace{2em} if $((\text{not} \ \mathcal{G}.\text{same\_component}(\alpha(i), q)) \ \text{and} \ \text{CONNECT}(\alpha(i), q))$ then
7 \hspace{2em} $\mathcal{G}.\text{add\_edge}(\alpha(i), q);$
Probabilistic Roadmaps (PRMs)

Step 1: Preprocessing Phase.

```plaintext
BUILD_ROADMAP
1   G.init(); i ← 0;
2   while i < N
3     if α(i) ∈ C_free then
4         G.add_vertex(α(i))
5         for each q ∈ NEIG(α(i))
6             if (not G.same_conf(α(i), q))
7                 G.add_edge(α(i), q);
```

α(i) is an ith sample in the configuration space. Each sample can be drawn uniformly (or more intelligently as described later).

borrowed from “Planning Algorithms” by S. LaValle
Probabilistic Roadmaps (PRMs)

Step 1: Preprocessing Phase.

BUILD_ROADMAP
1 \( G\).init(); \( i \leftarrow 0; \)
2 while \( i < N \)
3 \quad \text{if } \alpha(i) \in C_{\text{free}} \text{ then}
4 \quad \quad G\).add_vertex(\alpha(i)); \( i \leftarrow i + 1; \)
5 \quad \quad \text{for each } q \in \text{NEIGHBORHOOD}(\alpha(i), G) \)
6 \quad \quad \quad \text{if } \((\text{not } G\).\text{same}_\text{component}(\alpha(i), q)) \text{ and } \text{CONNECT}(\alpha(i), q) \text{ then}
7 \quad \quad \quad \quad G\).add_edge(\alpha(i), q);

borrowed from “Planning Algorithms” by S. LaValle

some region around the configuration sample

can be connected by a local planner
Probabilistic Roadmaps (PRMs)

Step 1: Preprocessing Phase.

BUILDROADMAP
1 \( \mathcal{G}.\text{init}(); \ i \leftarrow 0; \)
2 while \( i < N \)
3 \text{if} \ \( \alpha(i) \in \mathcal{C}_{\text{free}} \) \text{then}
4 \( \mathcal{G}.\text{add}\_\text{vertex}(\alpha(i)); \ i \leftarrow i + 1; \)
5 \text{for each} \ q \in \text{NEIGHBORHOOD}(\alpha(i), \mathcal{G})
6 \text{if} \ ((\text{not} \ \mathcal{G}.\text{same}\_\text{component}(\alpha(i), q)) \ \text{and} \ \text{CONNECT}(\alpha(i), q)) \ \text{then}
7 \( \mathcal{G}.\text{add}\_\text{edge}(\alpha(i), q); \)

borrowed from “Planning Algorithms” by S. LaValle

\( \text{can be replaced with: “number of successors of } q < K “ \)
Probabilistic Roadmaps (PRMs)

Step 1: Preprocessing Phase.

Efficient implementation of $q \in \text{NEIGHBORHOOD}(\alpha(i), \mathcal{G})$

- select $K$ vertices closest to $\alpha(i)$
- select $K$ (often just 1) closest points from each of the components in $\mathcal{G}$
- select all vertices within radius $r$ from $\alpha(i)$

borrowed from “Planning Algorithms” by S. LaValle
Probabilistic Roadmaps (PRMs)

Step 1: Preprocessing Phase.

Sampling strategies
- sample uniformly from $C_{\text{free}}$

Why do we need anything better than uniform sampling?

borrowed from “Planning Algorithms” by S. LaValle
Probabilistic Roadmaps (PRMs)

Step 1: Preprocessing Phase.

Sampling strategies
- sample uniformly from $C_{\text{free}}$
- select at random an existing vertex with a probability distribution inversely proportional to how well-connected a vertex is, and then generate a random motion from it to get a sample $\alpha(i)$
- bias sampling towards obstacle boundaries

borrowed from “Planning Algorithms” by S. LaValle
Step 1: Preprocessing Phase.

Sampling strategies
- sample $q_1$ and $q_2$ from Gaussian around $q_1$ and if either is in $C_{obs}$, then the other one is set as $\alpha(i)$
- sample $q_1$, $q_2$, $q_3$ from Gaussian around $q_2$ and set $q_2$ as $\alpha(i)$ if $q_2$ is in $C_{free}$, and $q_1$ and $q_3$ are in $C_{obs}$
- bias sampling away from obstacles

borrowed from “Planning Algorithms” by S. LaValle
What You Should Know…

• Pros and cons of resolution-complete (grid-based and lattice-based graph search-based) methods for high-D planning

• Configuration spaces for manipulators

• Operation of PRM and its variants

• Theoretical properties of PRM

• Pros and cons of PRM