

Thuật toán BIT* (Batch Informed Trees)

1 Giới thiệu

BIT* là một thuật toán lập kế hoạch đường đi (path planning) kết hợp ưu điểm của cả hai phương pháp: thuật toán tìm kiếm trên đồ thị và thuật toán dựa trên lấy mẫu ngẫu nhiên. Thuật toán này được thiết kế để giải quyết bài toán tìm đường tối ưu trong không gian liên tục có chướng ngại vật, đặc biệt hiệu quả trong không gian nhiều chiều cao.

2 Ý tưởng chính của BIT*

- **Mô hình hóa không gian trạng thái:** BIT* xem tập hợp các điểm lấy mẫu ngẫu nhiên trong không gian trạng thái như một đồ thị ngẫu nhiên hình học ẩn (Implicit Random Geometric Graph - RGG). Các đỉnh là các điểm mẫu, các cạnh được xác định dựa trên khoảng cách địa lý.
- **Tìm kiếm có hướng dẫn heuristic:** Thuật toán sử dụng heuristic để ưu tiên mở rộng các nút có khả năng dẫn đến đường đi tối ưu, tương tự như A*.
- **Tìm kiếm theo từng lô (batch):** Thuật toán xử lý các điểm mẫu theo từng lô, mỗi lô bổ sung thêm điểm mới để tăng độ chính xác của đồ thị ẩn. Mỗi lô được tìm kiếm theo heuristic và kết quả được cải thiện dần theo thời gian.
- **Tái sử dụng thông tin:** BIT* áp dụng kỹ thuật tìm kiếm tăng dần như trong Lifelong Planning A* để tái sử dụng thông tin từ các lô trước, giúp tăng hiệu quả tính toán.

3 Định nghĩa bài toán và các biến

- Không gian trạng thái: $X \subseteq \mathbb{R}^n$
- Không gian tự do: $X_{\text{free}} = X \setminus X_{\text{obs}}$
- Điểm bắt đầu: $x_{\text{start}} \in X_{\text{free}}$
- Tập điểm đích: $X_{\text{goal}} \subset X_{\text{free}}$
- Đường đi: $\sigma : [0, 1] \rightarrow X$
- Hàm chi phí: $s(\sigma)$ (thường là độ dài đường đi)

- Mục tiêu:

$$\sigma^* = \arg \min_{\sigma \in \Sigma} \{s(\sigma) \mid \sigma(0) = x_{\text{start}}, \sigma(1) \in X_{\text{goal}}, \forall t, \sigma(t) \in X_{\text{free}}\}$$

- Tập điểm mẫu: X_{samples} (lấy ngẫu nhiên trong X_{free})
- Cây tìm kiếm: $T = (V, E)$, với V là tập đỉnh (các trạng thái đã mở rộng) và E là tập cạnh (các kết nối hợp lệ).

4 Mô tả chi tiết thuật toán

1. **Khởi tạo:** Lấy mẫu ngẫu nhiên tập điểm X_{samples} trong X_{free} , bao gồm x_{start} và X_{goal} . Xác định tham số RGG (bán kính r hoặc số láng giềng k).
2. **Xây dựng cây tìm kiếm:** Tạo cây $T = (V, E)$ bắt đầu từ x_{start} . Mở rộng cây theo heuristic:

$$\hat{f}(x) = \hat{g}(x) + \hat{h}(x)$$

Trong đó:

- $\hat{g}(x)$: Ước lượng chi phí từ x_{start} đến x .
 - $\hat{h}(x)$: Ước lượng chi phí từ x đến X_{goal} .
3. **Kết thúc batch:** Khi tìm được đường đi hoặc không thể mở rộng thêm, lưu lại chi phí tốt nhất c_{best} .
 4. **Thêm batch mới:** Thêm một lô mẫu mới vào tập điểm, tập trung vào vùng có thể cải thiện đường đi hiện tại (ví dụ: vùng elip giới hạn bởi c_{best}). Cập nhật lại cây tìm kiếm bằng phương pháp incremental search.
 5. **Lặp lại:** Quay lại bước 2 với lô mẫu mới, tiếp tục cải thiện đường đi.

5 Công thức tính toán chính

- Heuristic:

$$\hat{f}(x) = \hat{g}(x) + \hat{h}(x)$$

- Tập trạng thái tiềm năng:

$$X_{\hat{f}} := \{x \in X \mid \hat{f}(x) \leq c_{\text{best}}\}$$

- Cây tìm kiếm:

$$T = (V, E), \quad V \subseteq X_{\text{free}}, \quad E \subseteq V \times V$$

- Chi phí thực tế trên cây:

$$g_T(x) = \text{Chi phí từ } x_{\text{start}} \text{ đến } x \text{ trên } T.$$

6 Minh họa quá trình tìm kiếm của BIT*

- (a) Bắt đầu mở rộng cây từ x_{start} , các điểm mẫu được phân bố ngẫu nhiên, cây dần phát triển hướng về mục tiêu.
- (b) Khi tìm được một đường đi, batch hiện tại kết thúc.
- (c) Thêm batch mới tập trung vào vùng elip có thể cải thiện đường đi, cập nhật và mở rộng tiếp.
- (d) Batch kết thúc khi không còn cải thiện được đường đi hoặc hết các cạnh khả thi.

7 Ưu điểm của BIT*

- **Kết hợp A* và RRT***: Tìm kiếm có hướng dẫn như A*, sử dụng lấy mẫu như RRT*.
- **Anytime và hội tụ tối ưu**: Nhanh chóng tìm lời giải ban đầu và cải thiện dần theo thời gian.
- **Hiệu quả trong không gian nhiều chiều**: Thích hợp cho bài toán robot phức tạp như điều khiển cánh tay robot.

8 Tóm tắt thuật toán BIT*

1. Lấy mẫu ngẫu nhiên batch đầu tiên và xây dựng RGG.
2. Tìm kiếm cây mở rộng theo heuristic $\hat{f} = \hat{g} + \hat{h}$.
3. Kết thúc batch khi tìm được đường đi hoặc không thể mở rộng.
4. Thêm batch mẫu mới trong vùng tiềm năng cải thiện.
5. Tái sử dụng cây hiện có và mở rộng tiếp.
6. Lặp lại bước 2-5 cho đến khi đủ thời gian hoặc đạt độ chính xác mong muốn.