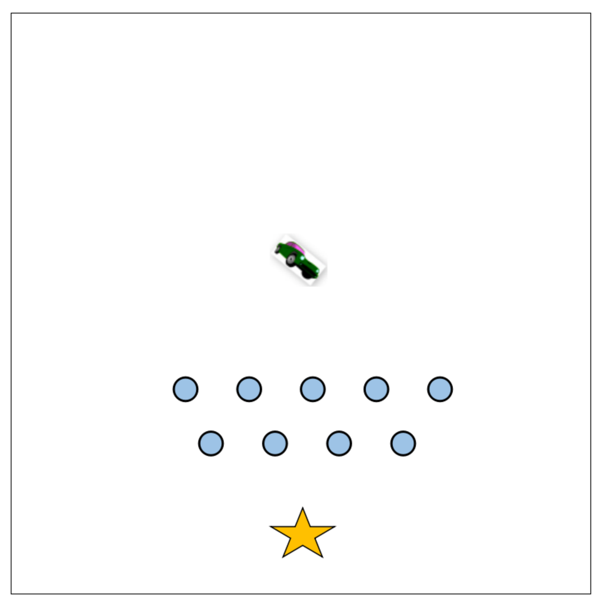
Fast Kinodynamic Motion Planning

Scott Kiesel, Tianyi Gu, Wheeler Ruml

(presented at IROS 2017)

University of New Hampshire

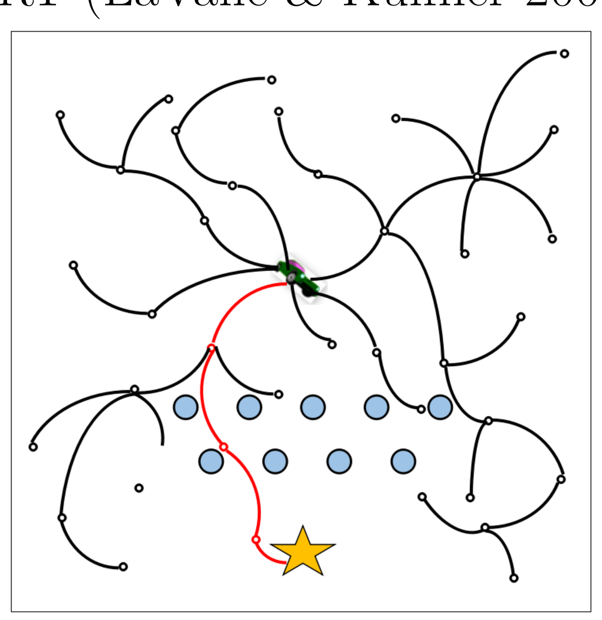
Problem Setup



- Given: environment, start state, goal region, vehicle dynamics
- Find: dynamically-feasible continuous trajectory (sequence of piece-wise constant controls) as quickly as possible!

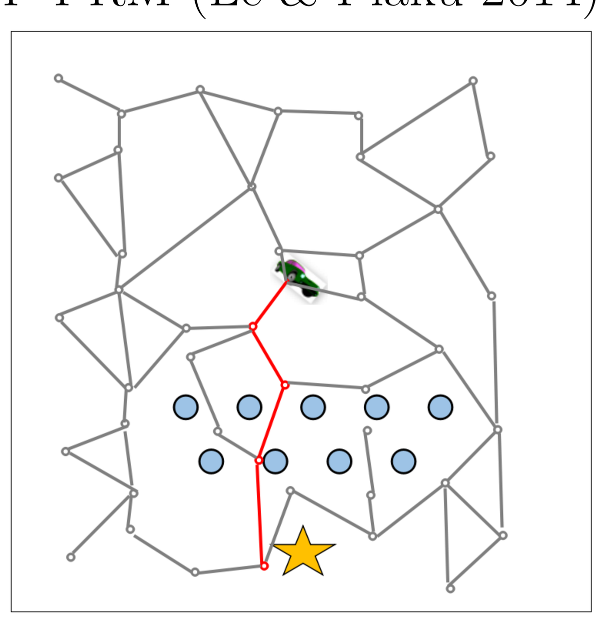
Previous Work: RRT & P-PRM

RRT (LaVallee & Kuffner 2001)

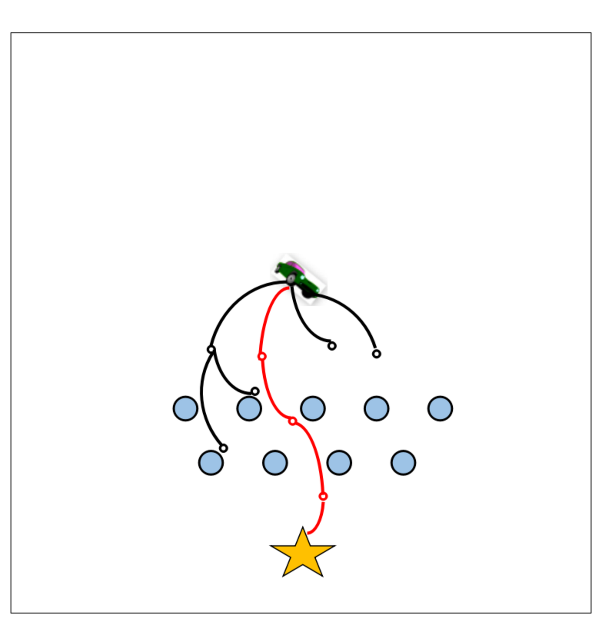


- Generate a (random) sample state
- Select nearest state in the existing motion tree
- Steer toward the sample, generating new state (or use a random control if no steering)
- Repeatedly grow the motion tree until it touchs the goal region

P-PRM (Le & Plaku 2014)



- Find a shortest path in an abstract graph from the start vertex to the goal vetex



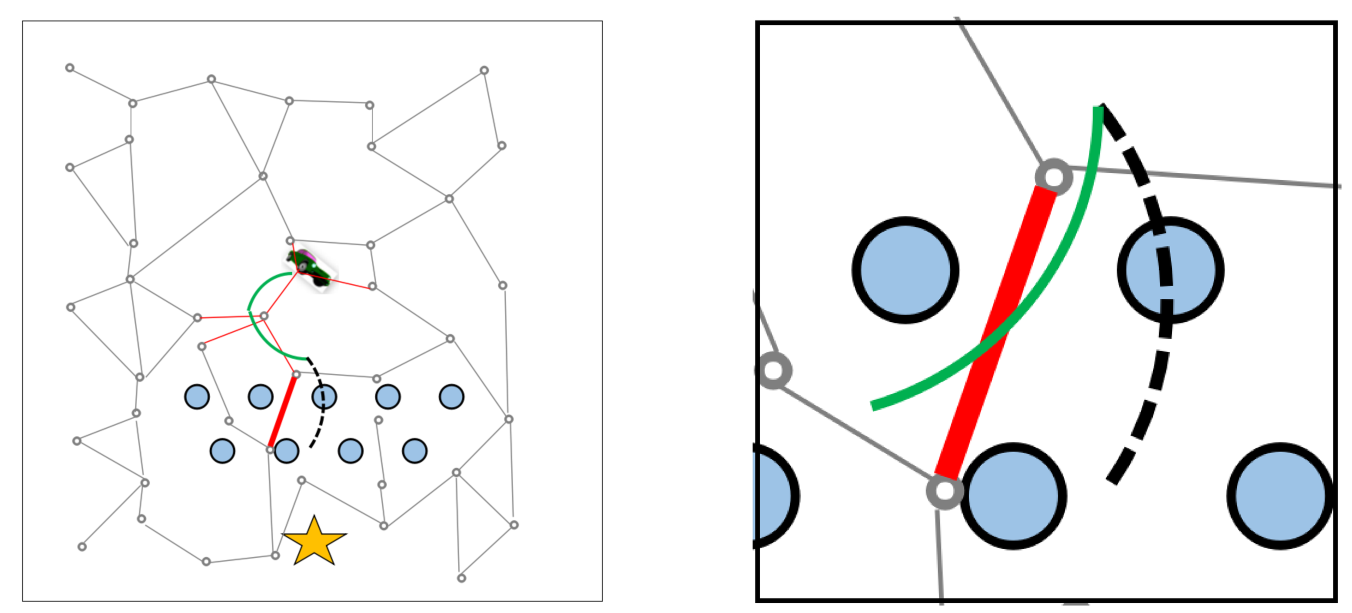
- Use heuristic cost-to-go information to guide growth of the motion tree

BEAST: Bayesian Effort-Aided Search Trees

How to estimate effort?

Minimize planning effort
$\approx$ Minimize # of total state propagation attempts

Local Effort Estimates:

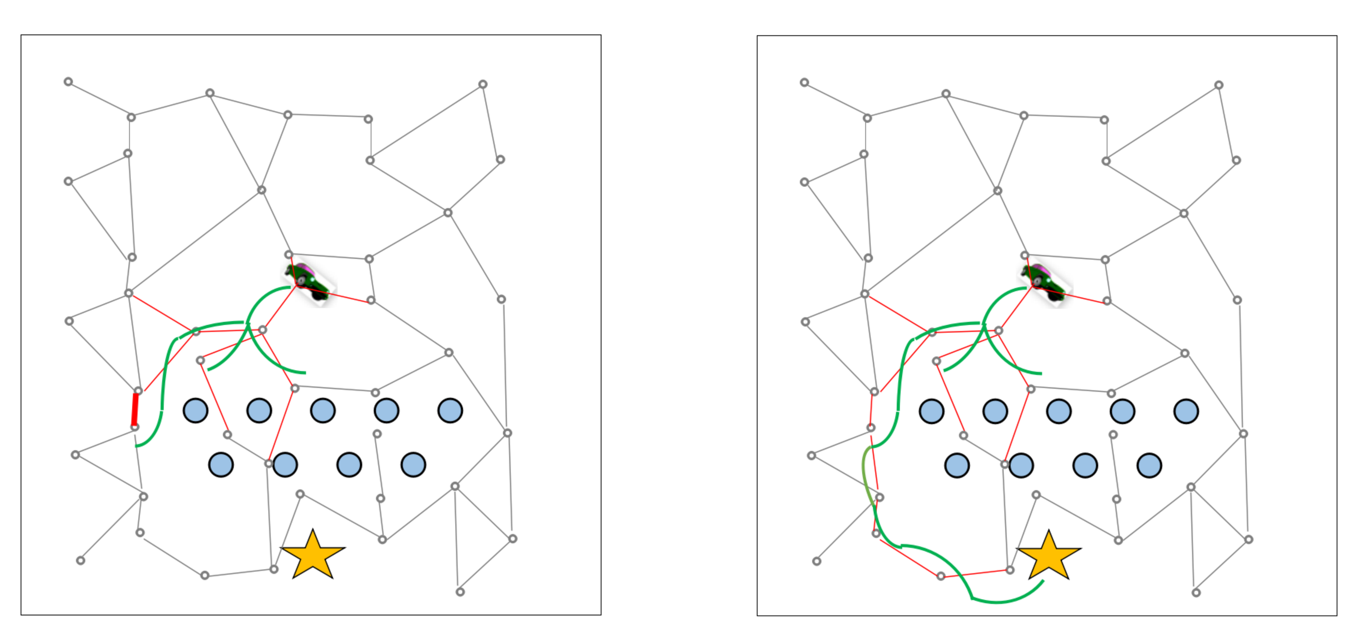


Maintain a beta Distribution for each edge:
Current belief regarding success rate along an edge

$$E[X] = \frac{success}{success + failure}$$

Edge weight in abstract graph
= expected # of propagation for one success attempt
$=E[X]^{-1}$

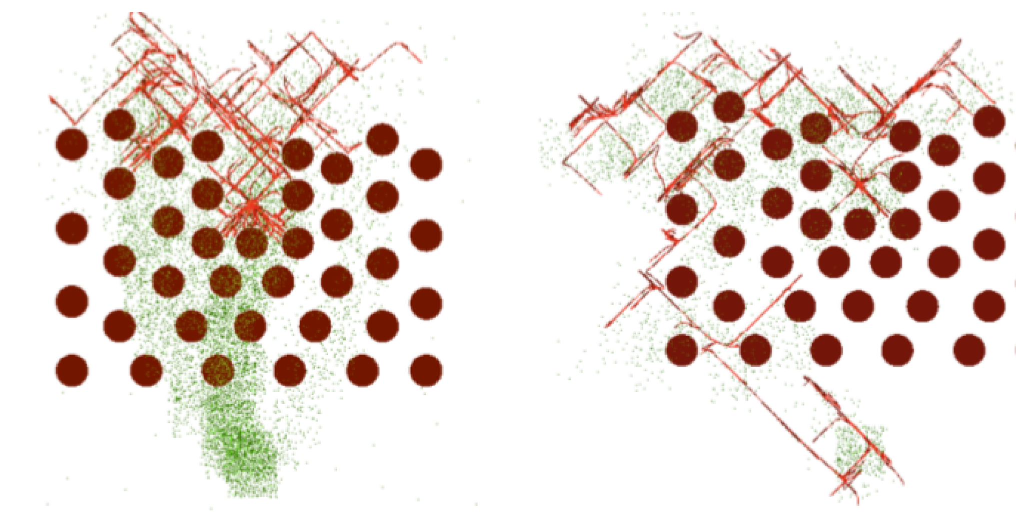
Global Effort Estimates:



- Given local effort estimates, we want estimate total effort to reach the goal.
- Accumulate local effort estimate along the shortest paths from each abstract state to the goal.
- Guide motion tree growth toward easy way

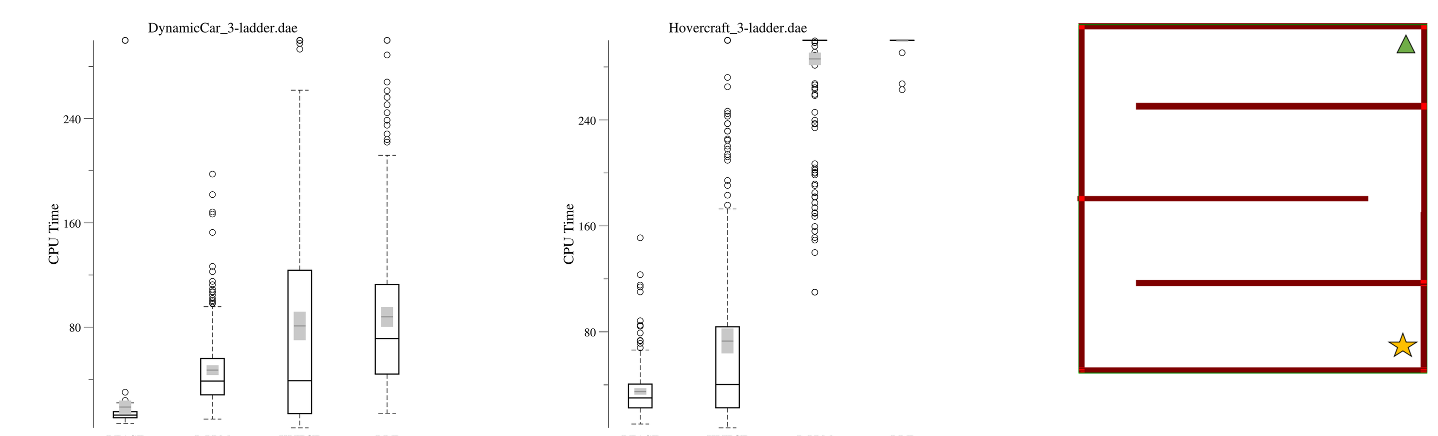
Experiments

Forest with Quadrotor:



Left: P-PRM generates samples (green dots) along low-cost abstract path but it is challenging to grow the motion tree (red lines). **Right**: BEAST has quickly learned that it is difficult to propagate the motion tree downward and has reached the goal faster.

3-ladder with Dynamic car & Hovercraft:



BEAST find complex path much faster than P-PRM, KPIECE, and RRT and is also much robust.

map	vehicle	P-PRM	KPIECE	RRT
open area	car	1.1–1.9	18–35	3.2–5.8
	hover.	0.7–1.1	3.2–6.9	5.9–10
single wall	car	4.9–6.3	6.2–9.1	6.2–8.1
	hover.	9.2–11	2.1–3.7	11–13
3 ladder	car	2.9–3.4	2.4–4.6	5.0–6.3
	hover.	8.7–10	1.3–1.7	9.1–10
2D forest	car	1.1–1.6	100–131	21–38
	hover.	1.0–1.8	9.4–16	17–25
3D forest	quad.	1.0–1.3	0.5–0.8	7.1–10
	blimp	2.5–3.3	60–75	28–43
fifthelement	quad.	1.0–1.2	3.8–4.9	4.6–6.1
	blimp	1.2–1.7	32–44	5.1–21

This table gives 95% confidence intervals on the median slowdown of the other planners relative to BEAST. A value of 1.9 means that the algorithm took 1.9 times as long as BEAST to find a solution. The gray cells are those case in which a planner did not find a solution within 300 seconds.