An Effort Bias for Sampling-based Motion Planning

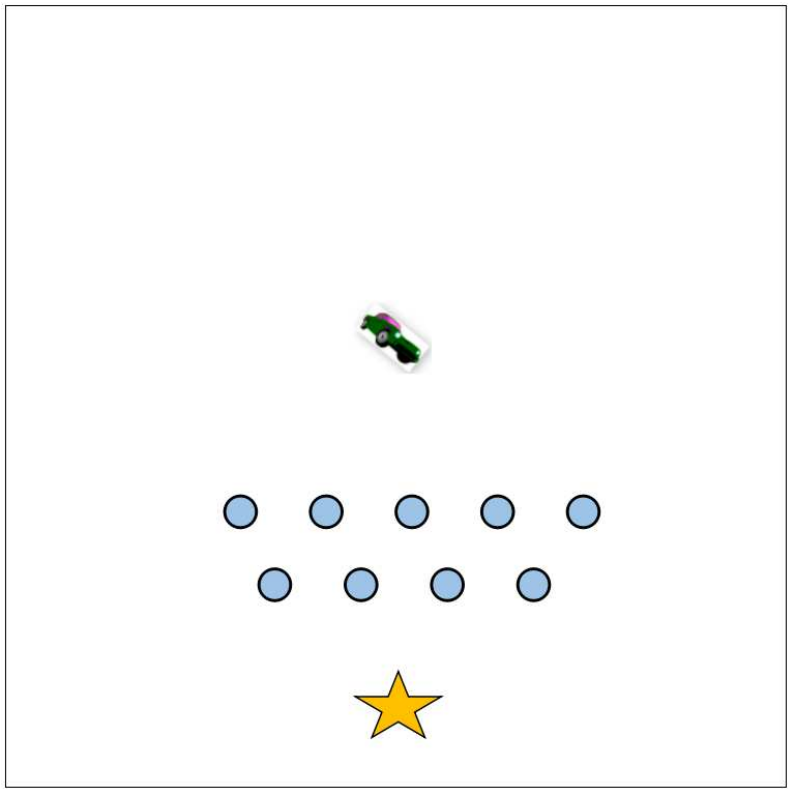
Scott Kiesel and Tianyi Gu and Wheeler Ruml


University of New Hampshire

Department of Computer Science

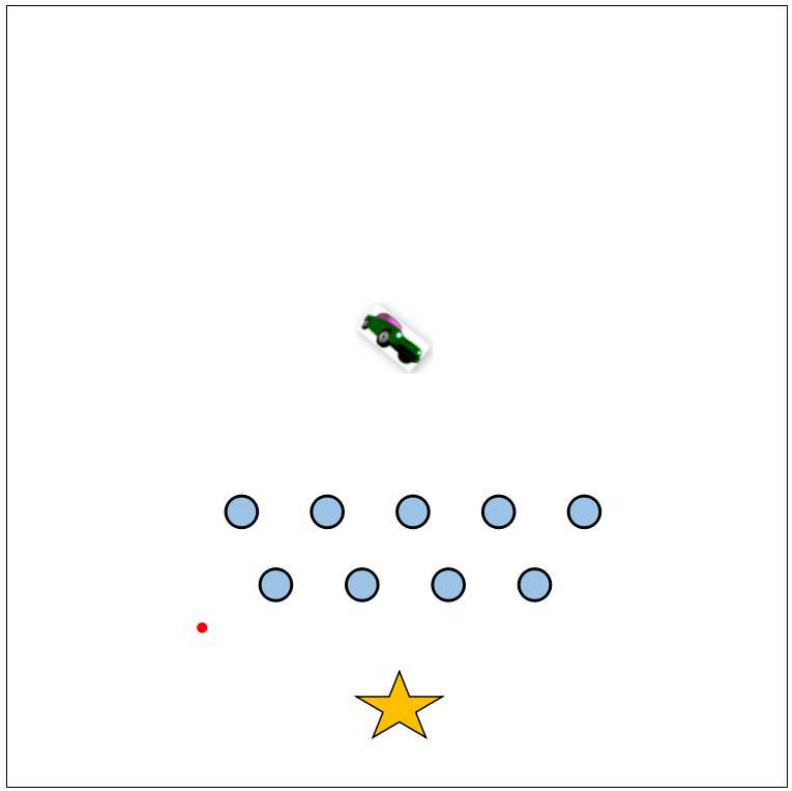
Grateful thanks to NSF (grant 1150068) for support

The Problem: Fast Kinodynamic Motion Planning



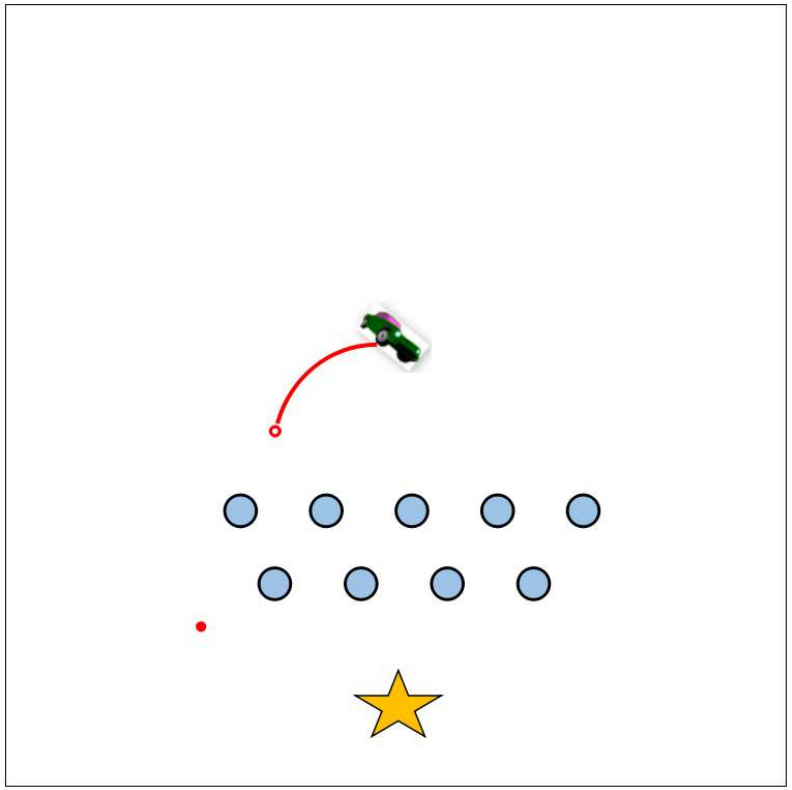
- Given: environment, start state, goal region, vehicle dynamics
- Find: dynamically-feasible continuous trajectory (sequence of piece-wise constant controls) as quickly as possible!

Growing a Motion Tree: RRT (LaValle & Kuffner 2001)



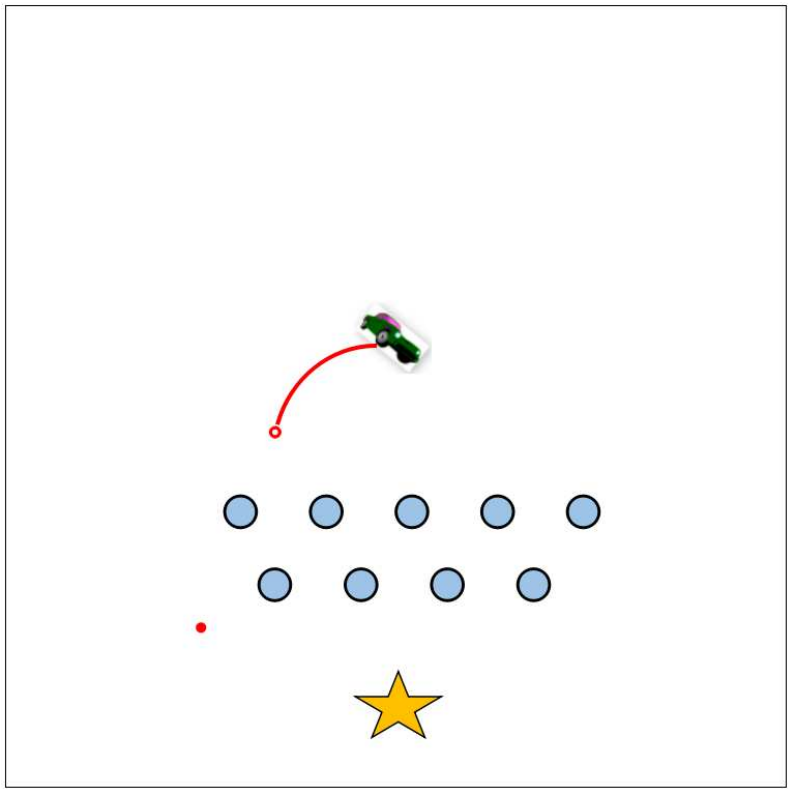
- Generate a (random) sample state

Growing a Motion Tree: RRT



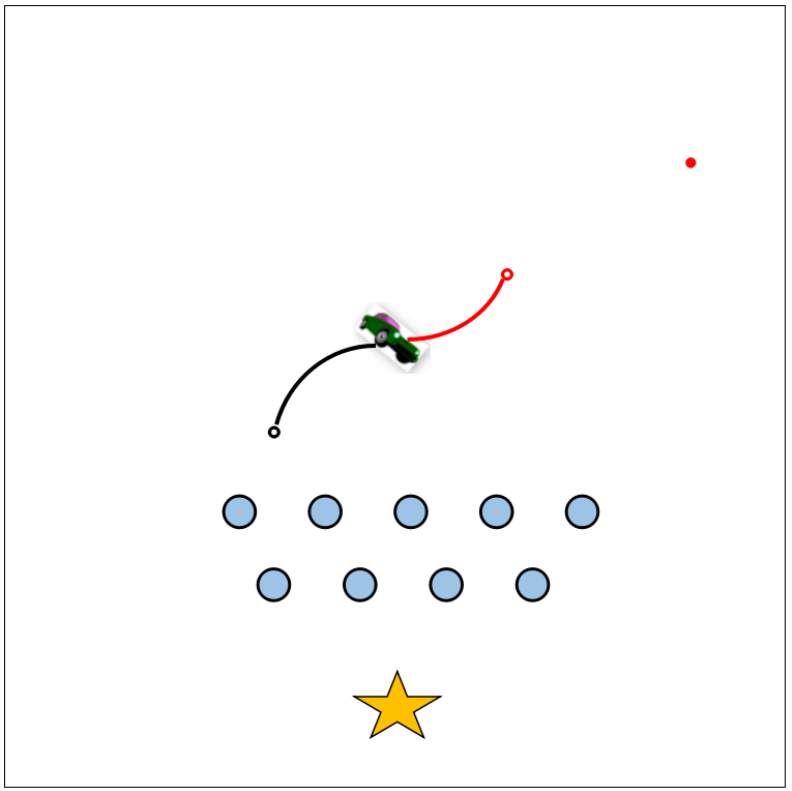
- Generate a (random) sample state
- Select nearest state in the existing motion tree

Growing a Motion Tree: RRT



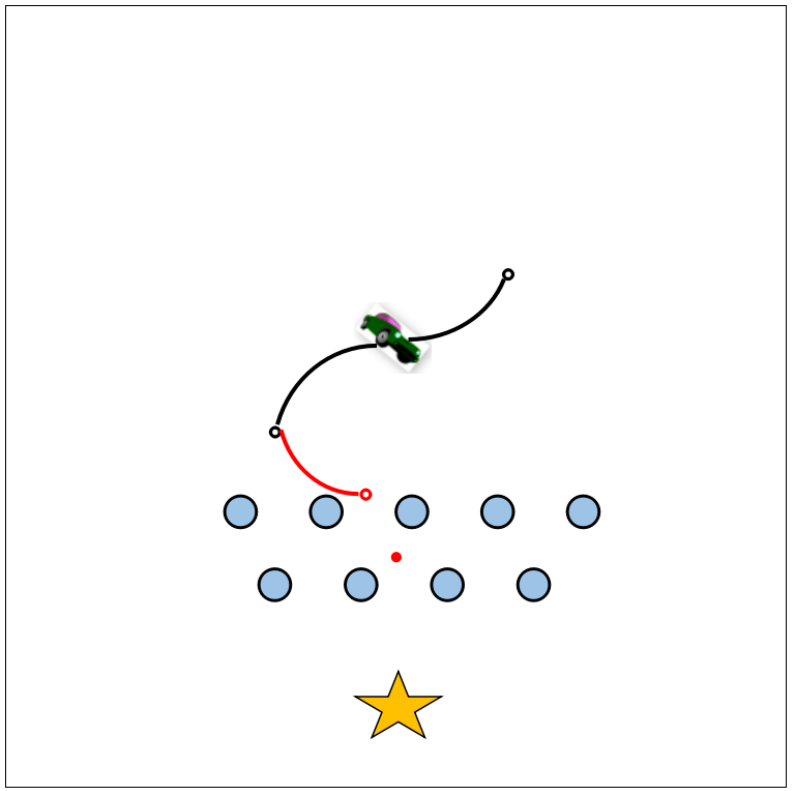
- Generate a (random) sample state
- Select nearest state in the existing motion tree
- Steer toward the sample, generating new state
  (or use several random controls if no steering)

Growing a Motion Tree: RRT



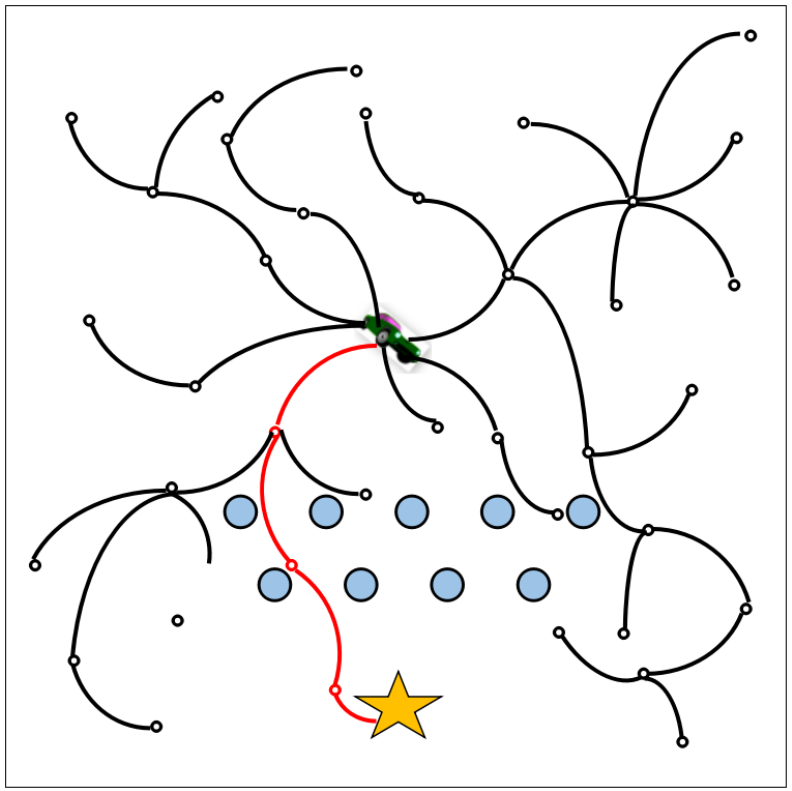
- Generate a (random) sample state
- Select nearest state in the existing motion tree
- Steer toward the sample, generating new state
  (or use best of several random controls if no steering)
- Repeatedly grow the motion tree until it touchs the goal region

Growing a Motion Tree: RRT



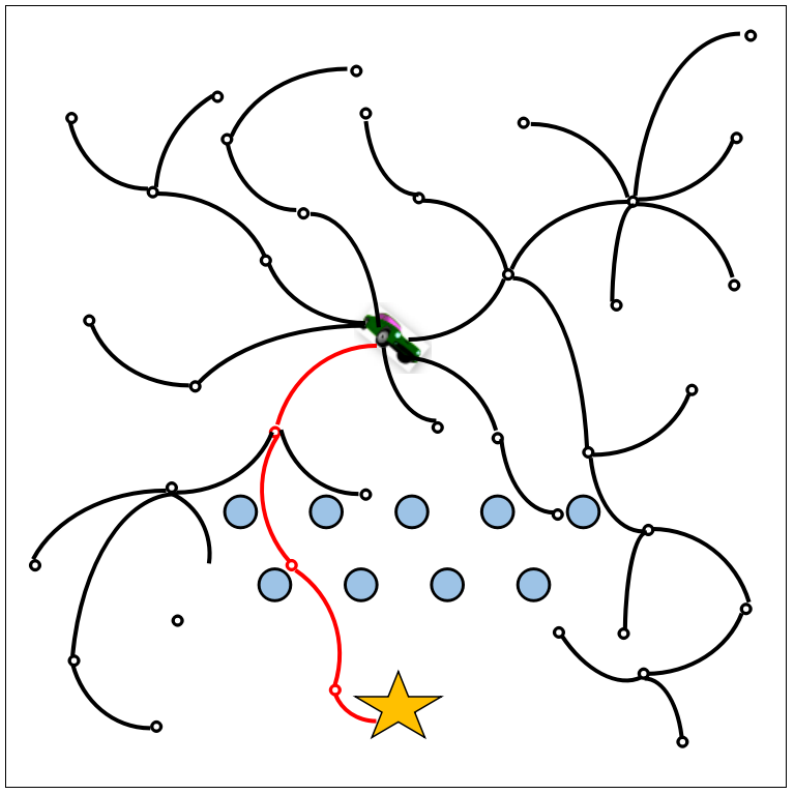
- Generate a (random) sample state
- Select nearest state in the existing motion tree
- Steer toward the sample, generating new state
  (or use best of several random controls if no steering)
- Repeatedly grow the motion tree until it touchs the goal region

Growing a Motion Tree: RRT



- 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

Growing a Motion Tree: RRT



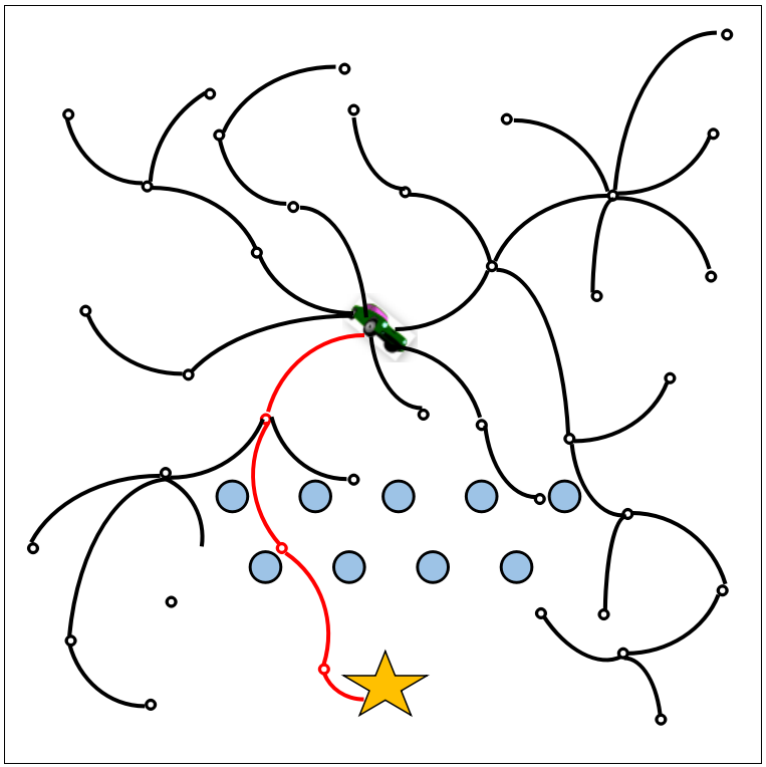
- General – only forward simulator required
- Voronoi bias to encourage coverage
- More recent work (EST, KPIECE) also emphasizes coverage

coverage $\neq$ fast planning

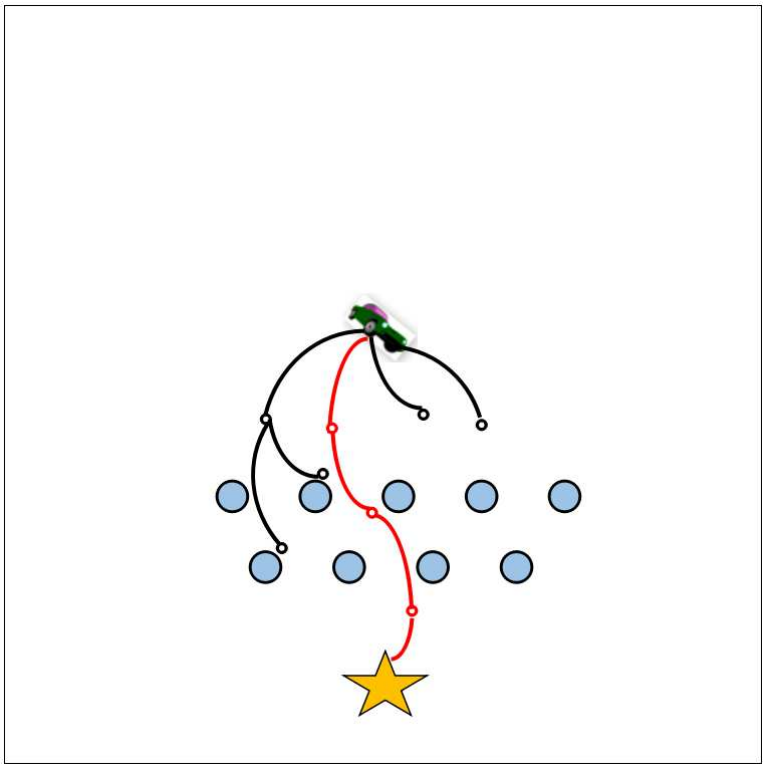
Cost-guided Planning: P-PRM (Le & Plaku 2014)

- To increase speed, be goal directed
- Cost-guided abstract path-based planner: P-PRM
  use discrete abstraction of state space to guide sampling

RRT

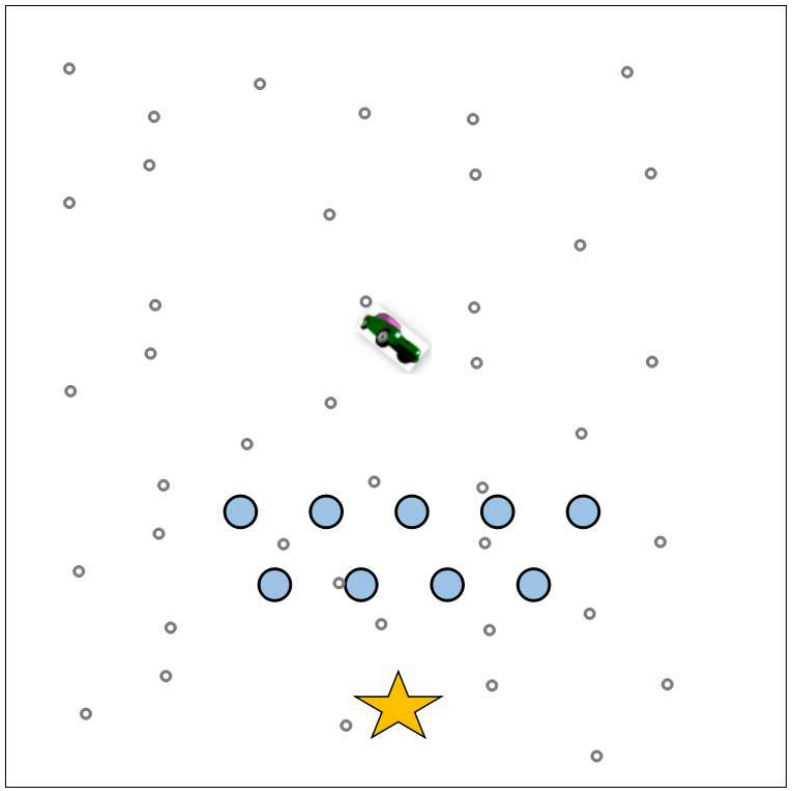


P-PRM



Cost-guided Planning: P-PRM

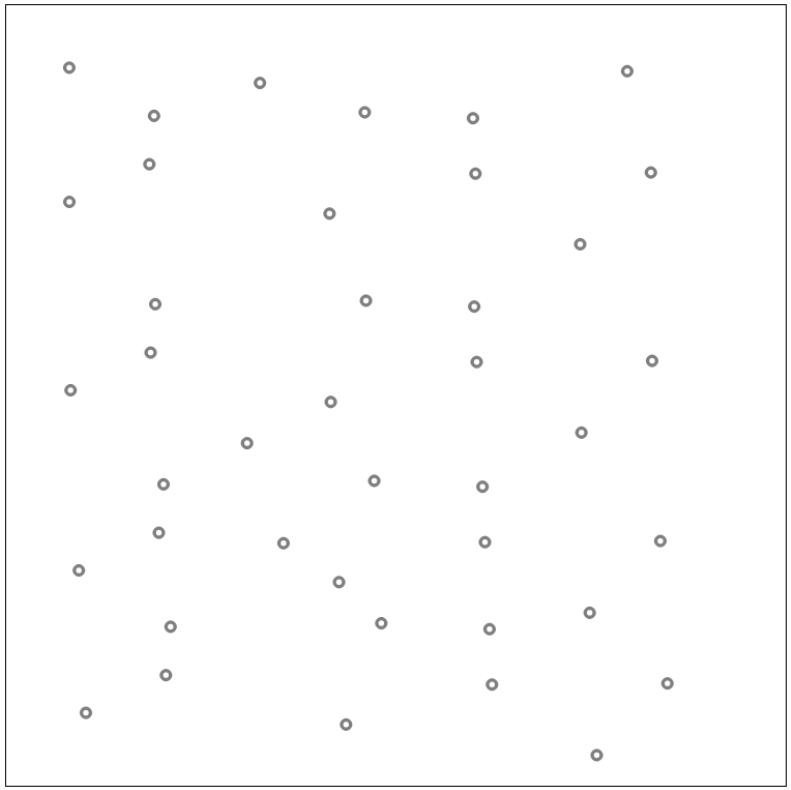
Abstract the state space:



- Randomly sample low dimensional abstract states
  (Use as vertices, each vertex represent an abstract region)

Cost-guided Planning: P-PRM

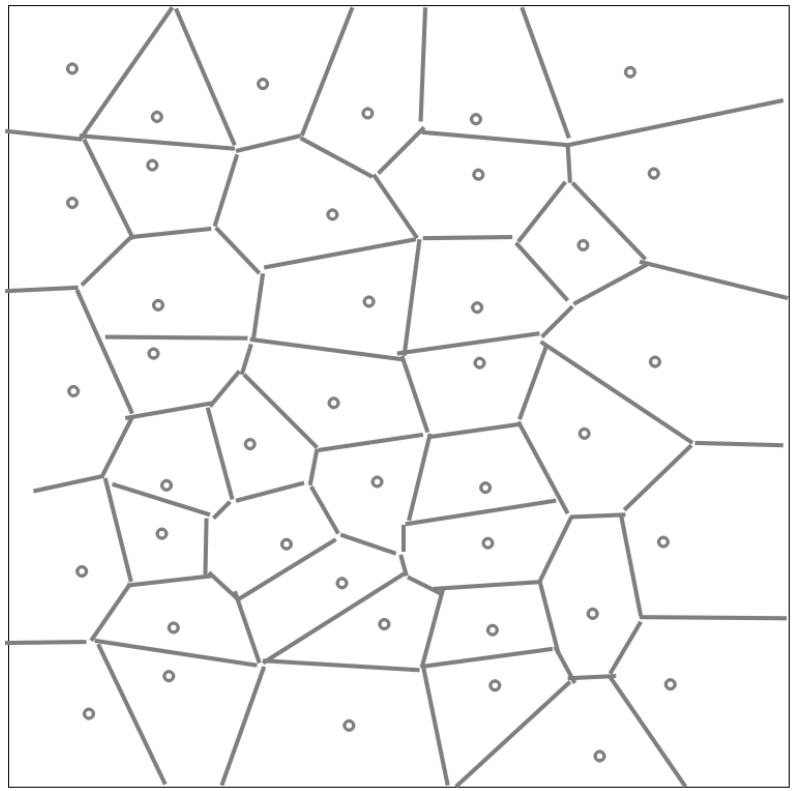
Abstract the state space:



- Randomly sample low dimensional abstract states
  (Use as vertices, each vertex represent an abstract region)

Cost-guided Planning: P-PRM

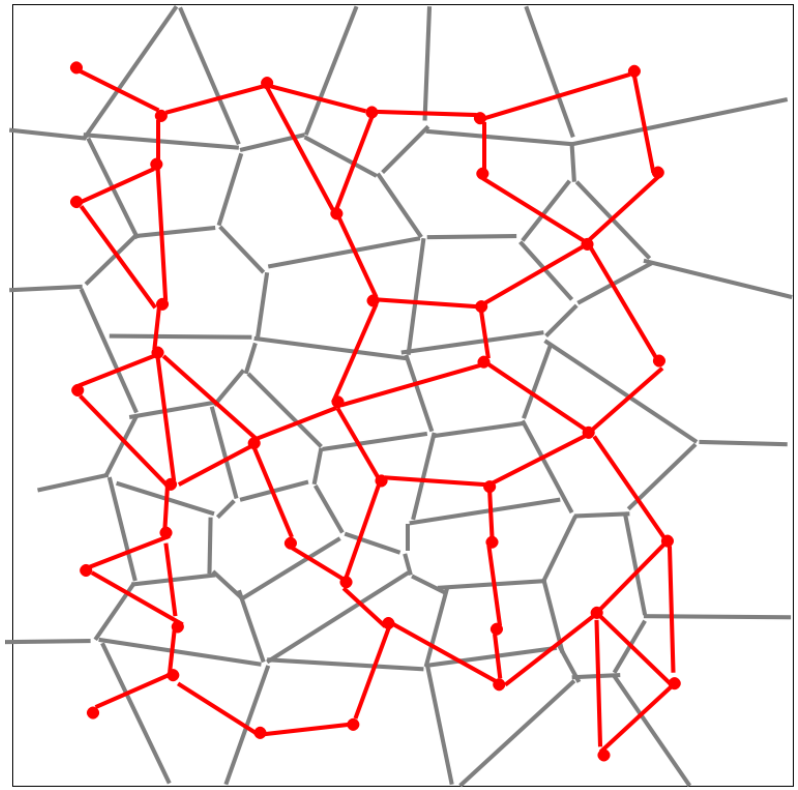
Abstract the state space:



- Randomly sample low dimensional abstract states
  (Use as vertices, each vertex represent an abstract region)

Cost-guided Planning: P-PRM

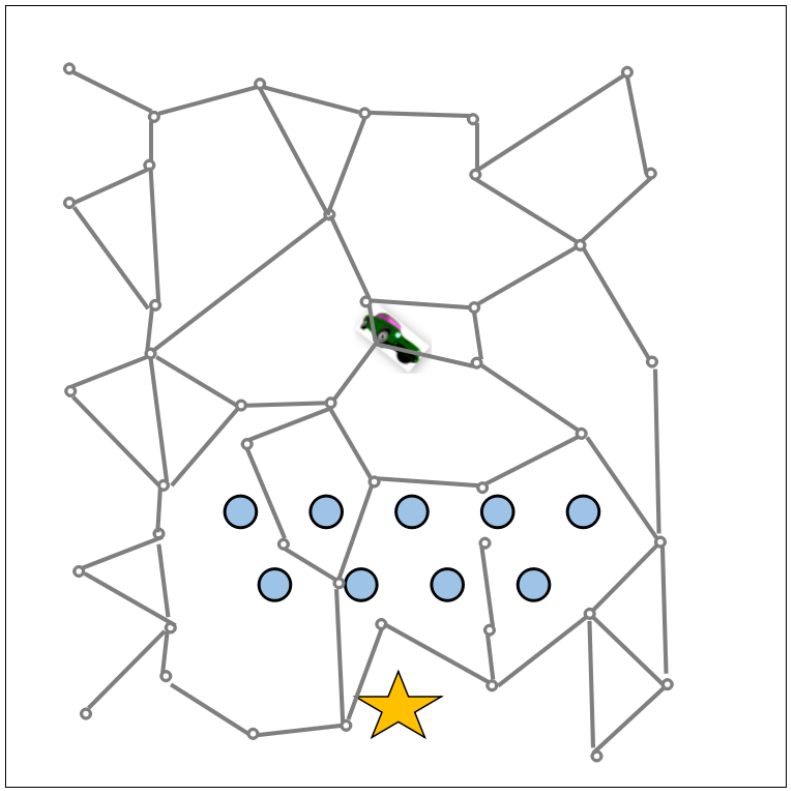
Abstract the state space:



- Randomly sample low dimensional abstract vertices (Each Vertex represent an abstract region)
- Connect neighbor vertices

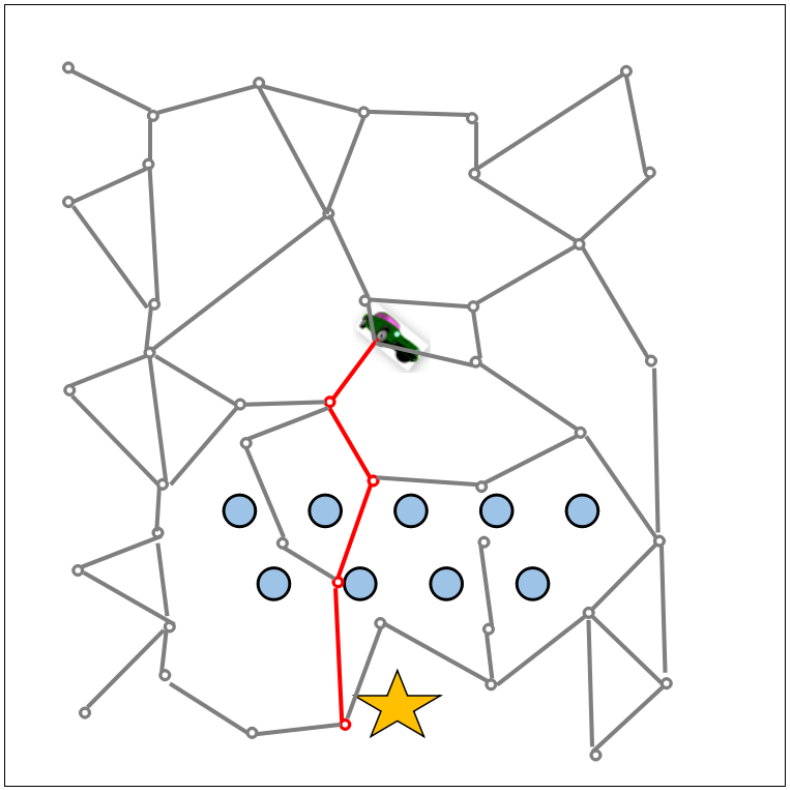
Cost-guided Planning: P-PRM

Abstract the state space:



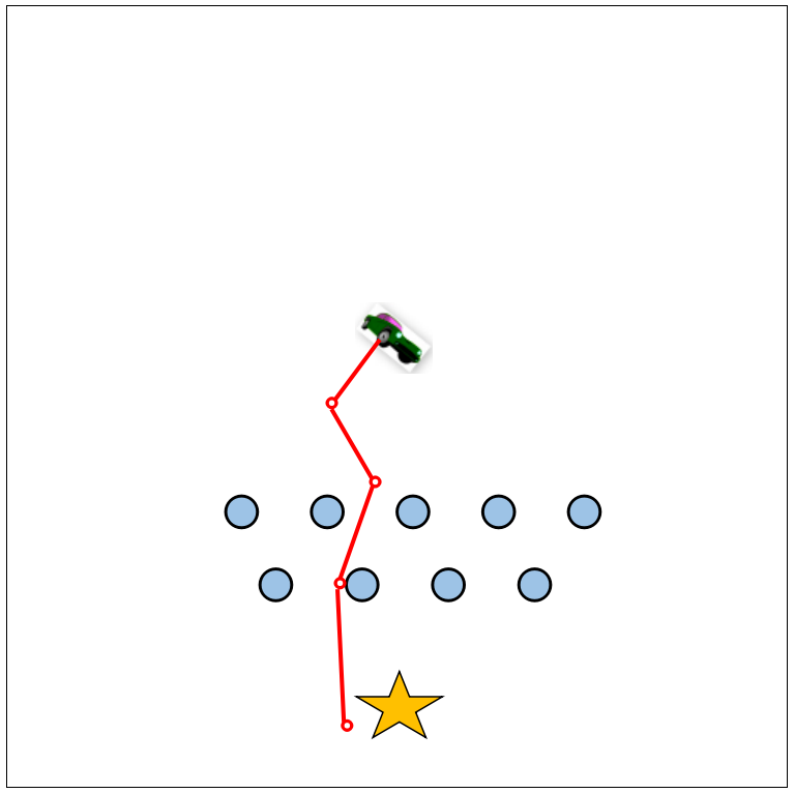
- Randomly sample low dimensional abstract vertices (Each Vertex represent an abstract region)
- Connect neighbor vertices
- Resulting abstract graph structure

Cost-guided Planning: P-PRM



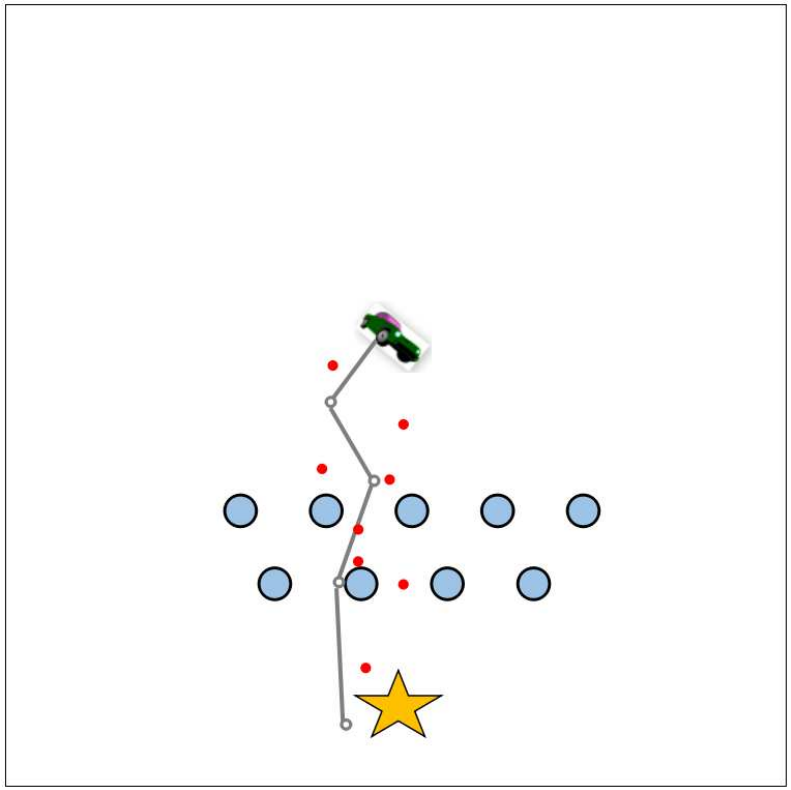
1. Find a shortest path from the start vertex to the goal vetex
2. Use heuristic cost-to-go information to guide growth of the motion tree.

Cost-guided Planning: P-PRM



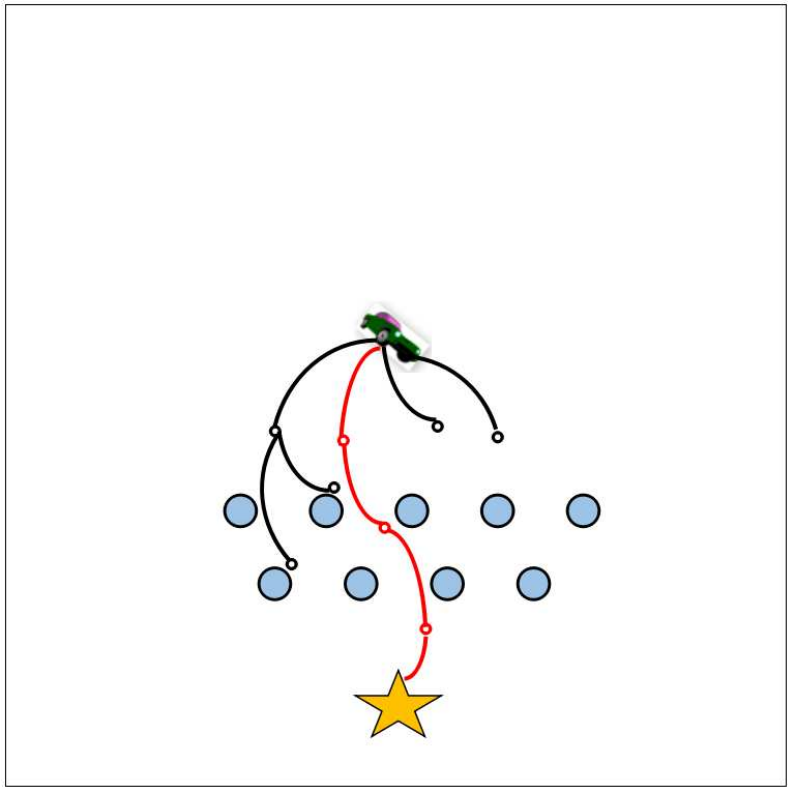
1. Find a shortest path from the start vertex to the goal vetex
2. Use heuristic cost-to-go information to guide growth of the motion tree.

Cost-guided Planning: P-PRM



1. Find a shortest path from the start vertex to the goal vetex
2. Use heuristic cost-to-go information to guide growth of the motion tree.

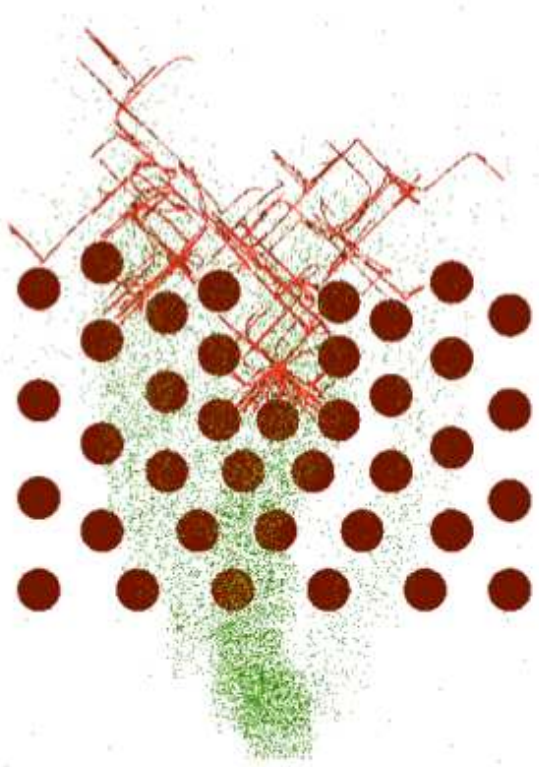
Cost-guided Planning: P-PRM



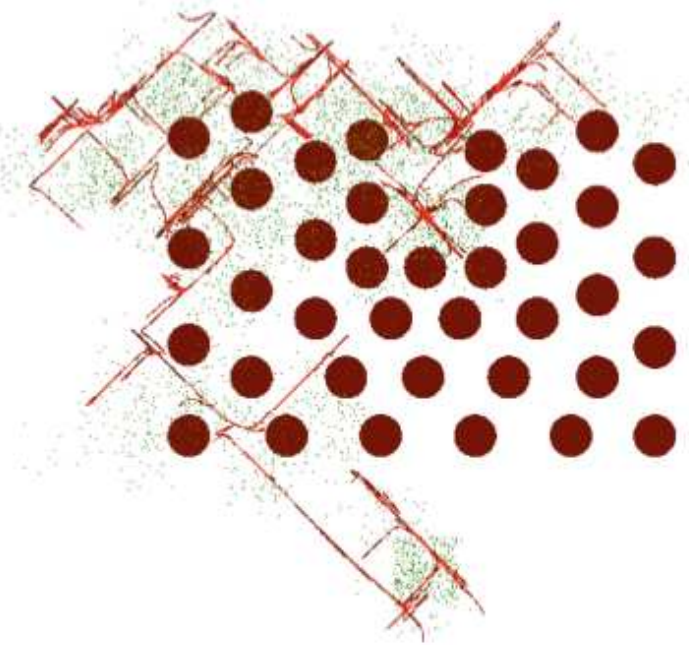
1. Find a shortest path from the start vertex to the goal vetex
2. Use heuristic cost-to-go information to guide growth of the motion tree.

Cost-guided Planning: P-PRM

P-PRM (cost-guided)



BEAST (our work)

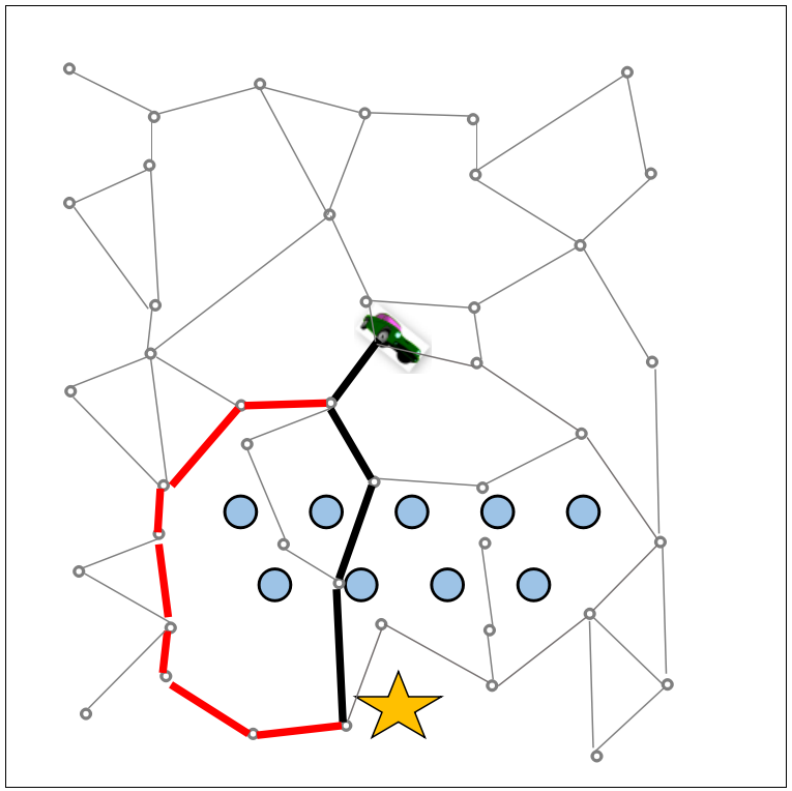


optimizing solution cost $\neq$ optimizing planning effort

Bayesian Effort-Aided Search Trees (BEAST)

Local Effort Estimates

Minimize planning effort
≈ Minimize # of total state propagation (steering) attempts

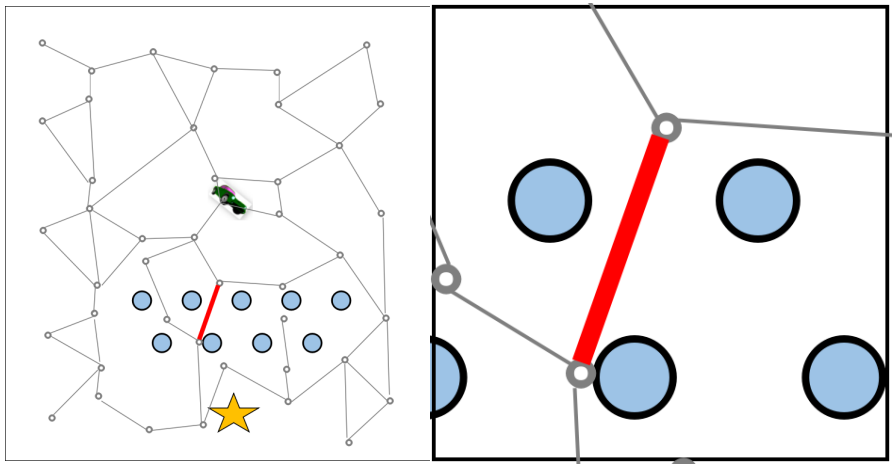


Local Effort Estimates

How to estimate # of propagation attempts?

Beta Distribution:
current belief regarding
success rate
along an edge
(across abstract regions)

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



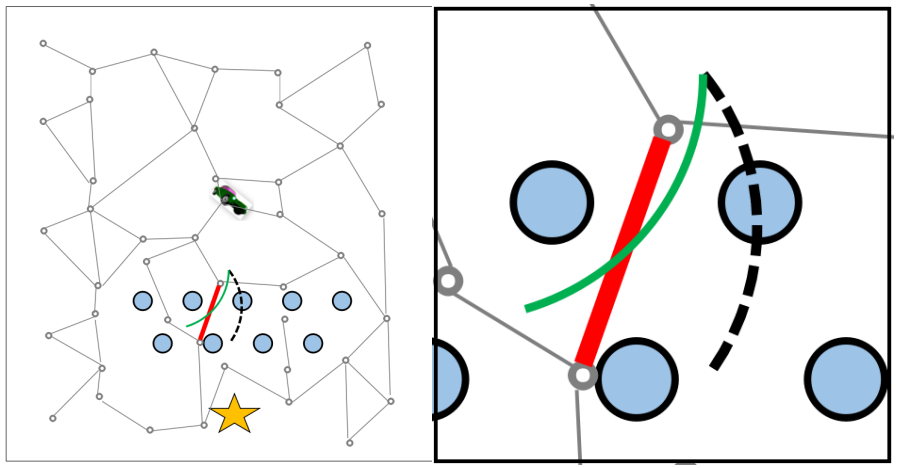
Edge weight in abstract graph
= expected # of propagation for one success attempt
$= E[X]^{-1}$

Local Effort Estimates

How to estimate # of propagation attempts?

Beta Distribution:
current belief regarding
success rate
along an edge
(across abstract regions)

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

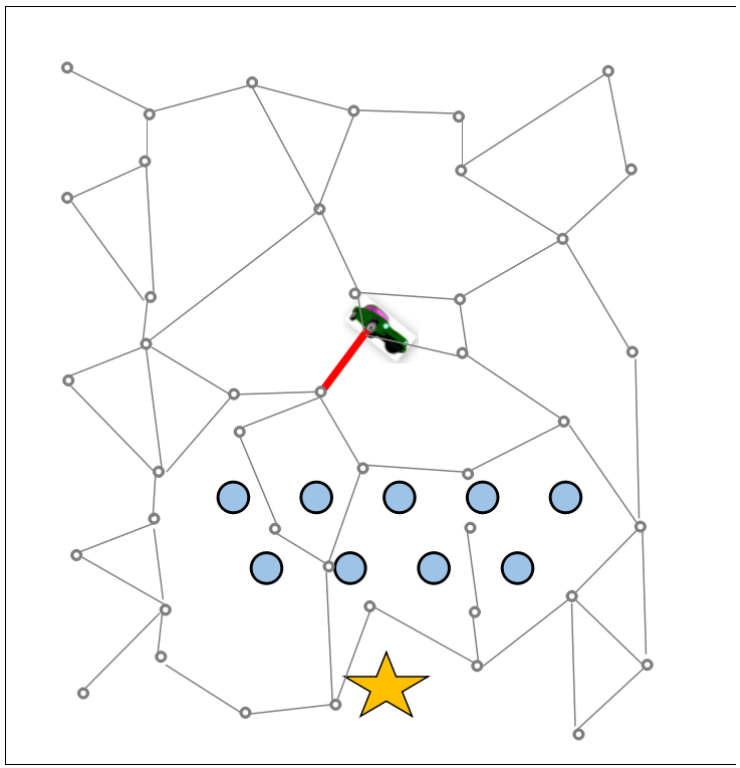


Edge weight in abstract graph
= expected # of propagation for one success attempt
$= E[X]^{-1}$

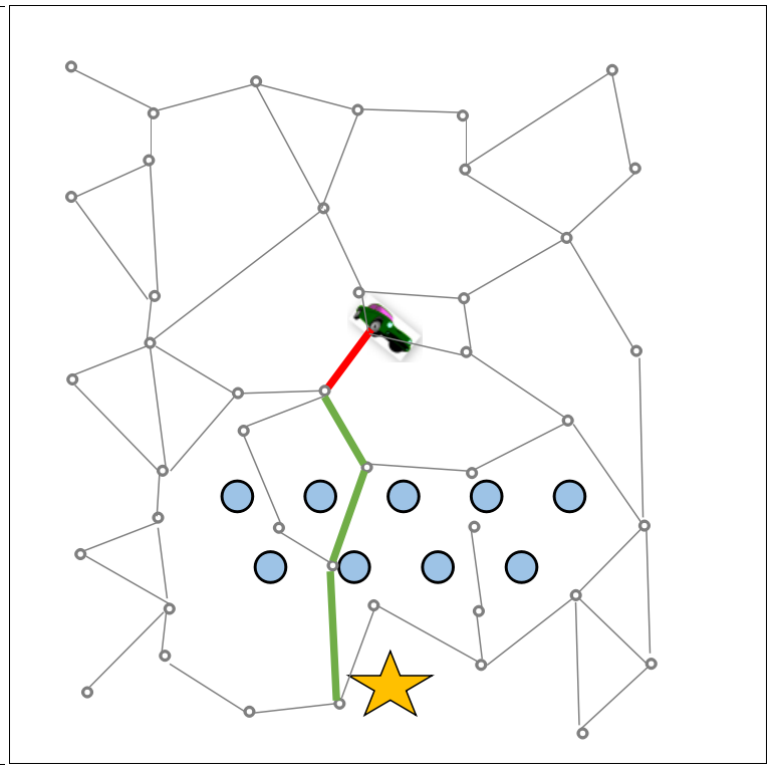
Global Effort Estimates

- Given local effort estimates, we want estimate total effort to reach the goal.
- Accumulate local effort estimate along the shortest paths from each state to the goal.

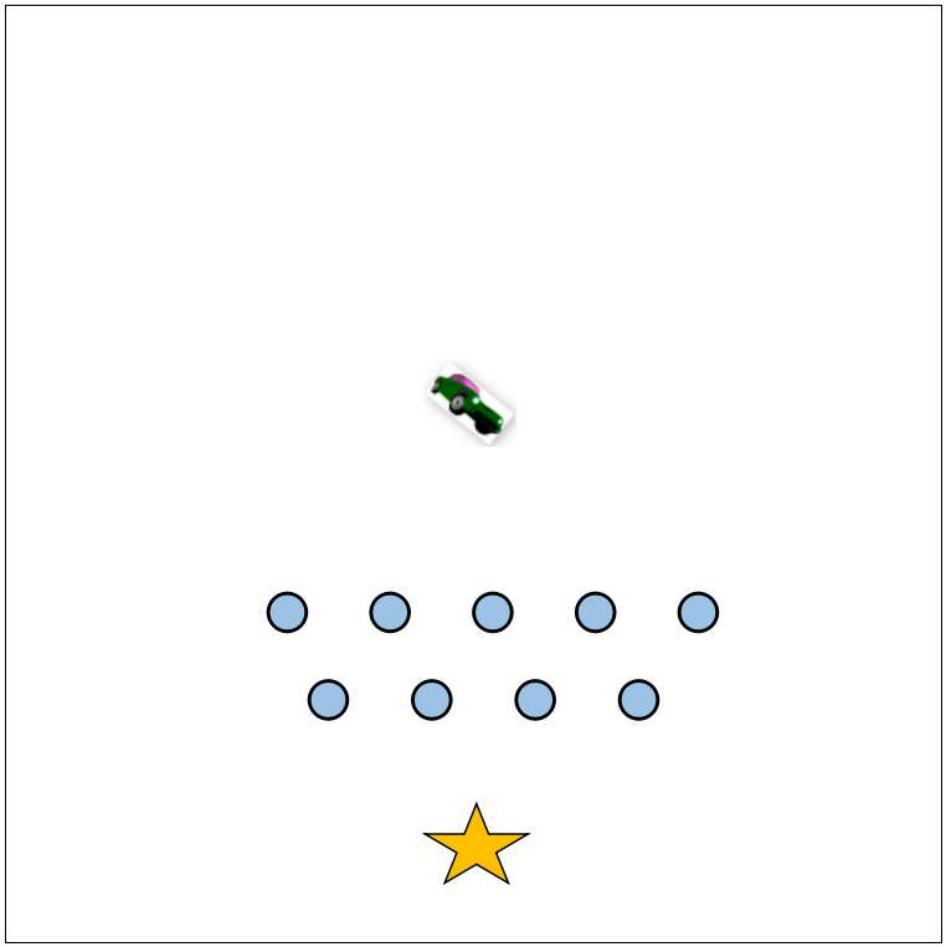
Local



Global

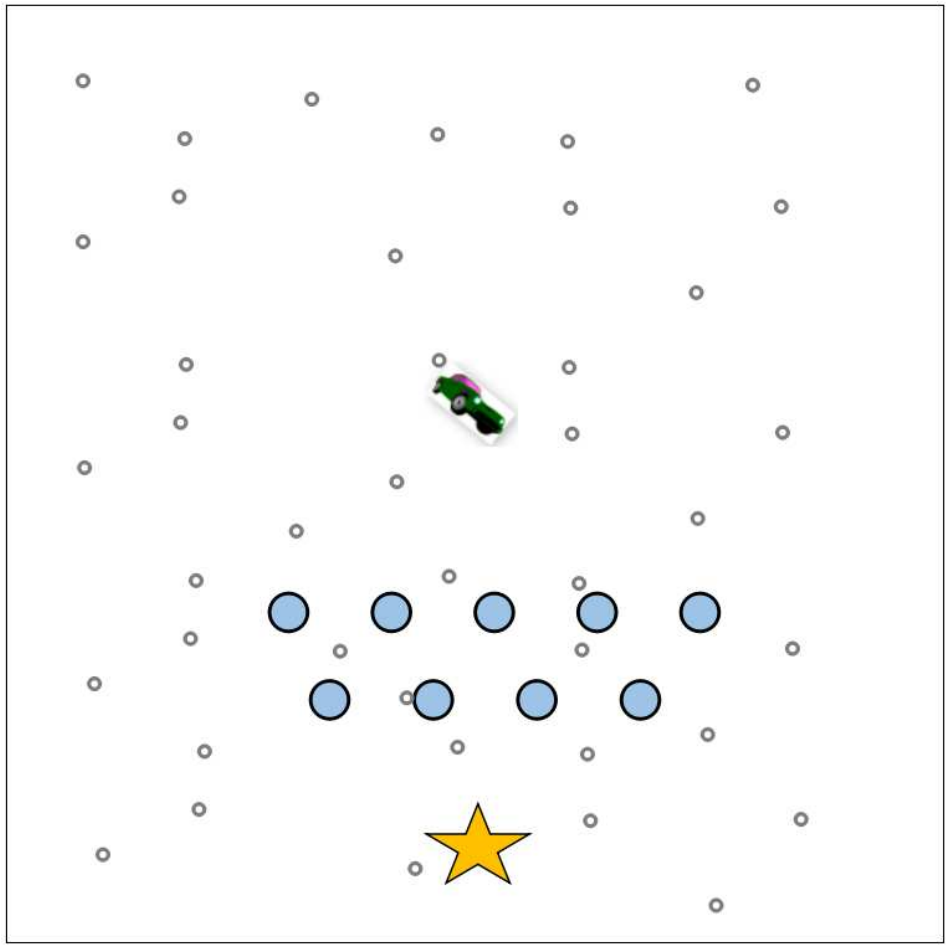


Effort-guided Planning: BEAST



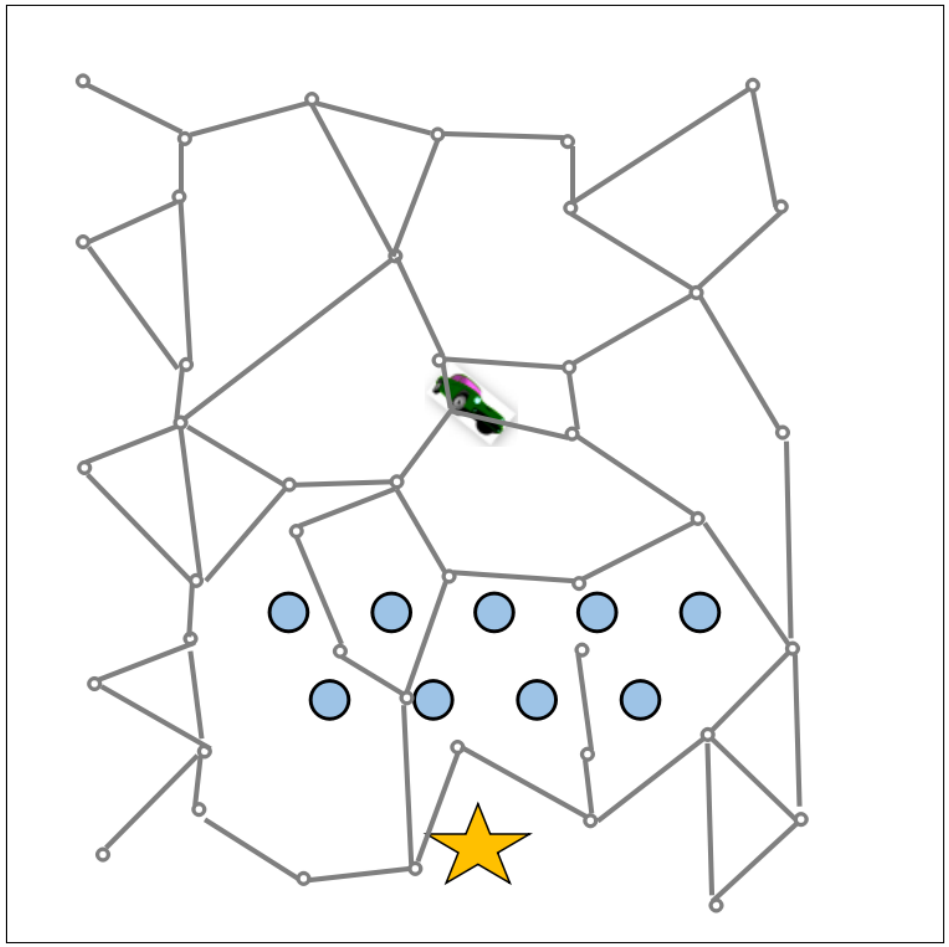
Generate abstract graph

Effort-guided Planning: BEAST



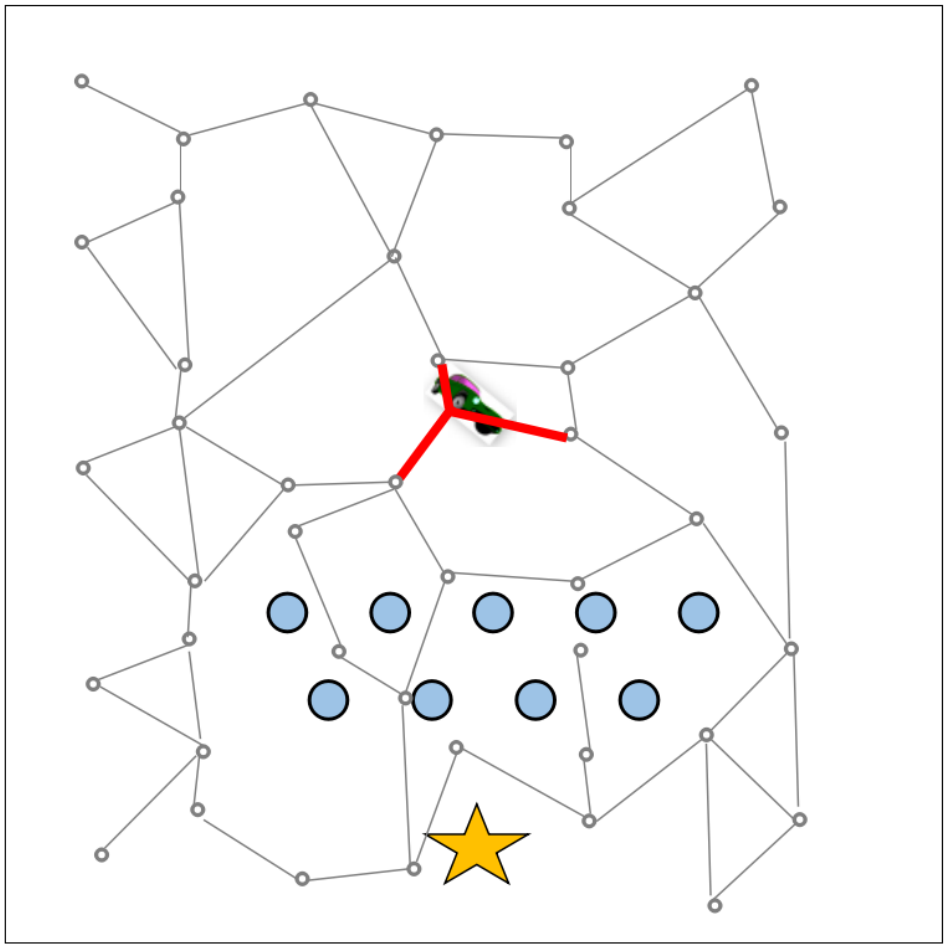
Generate abstract graph

Effort-guided Planning: BEAST



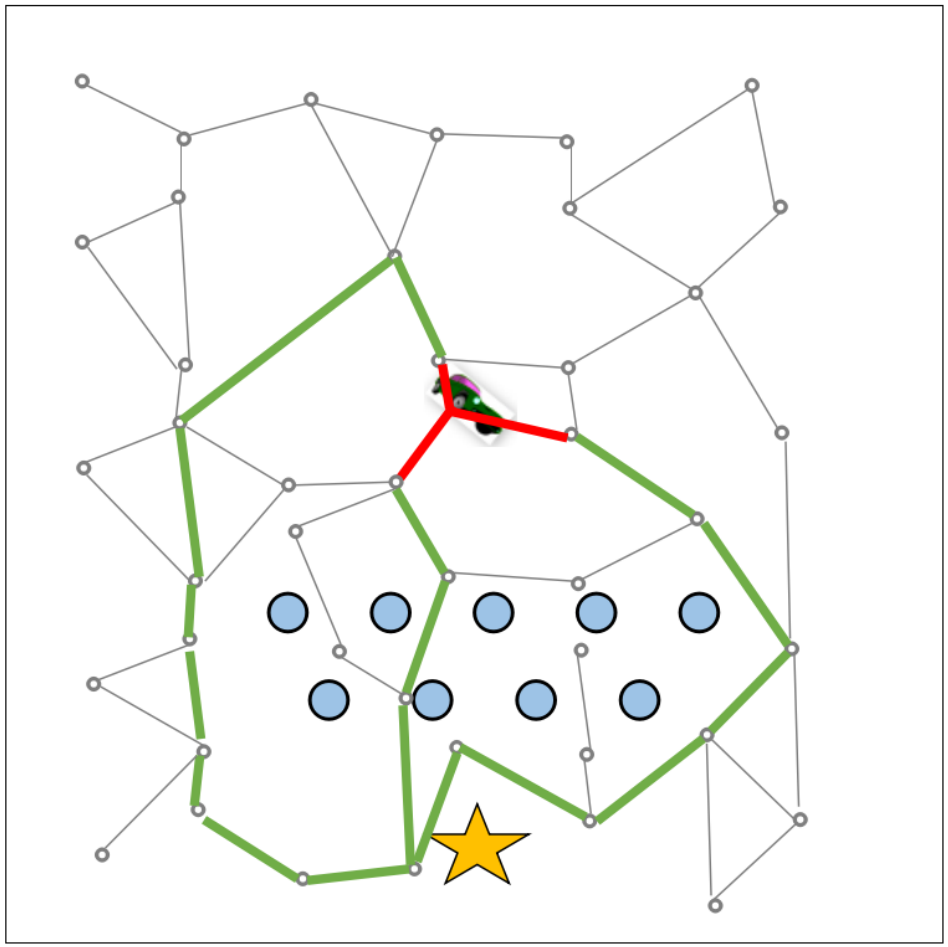
Initialize effort estimate

Effort-guided Planning: BEAST



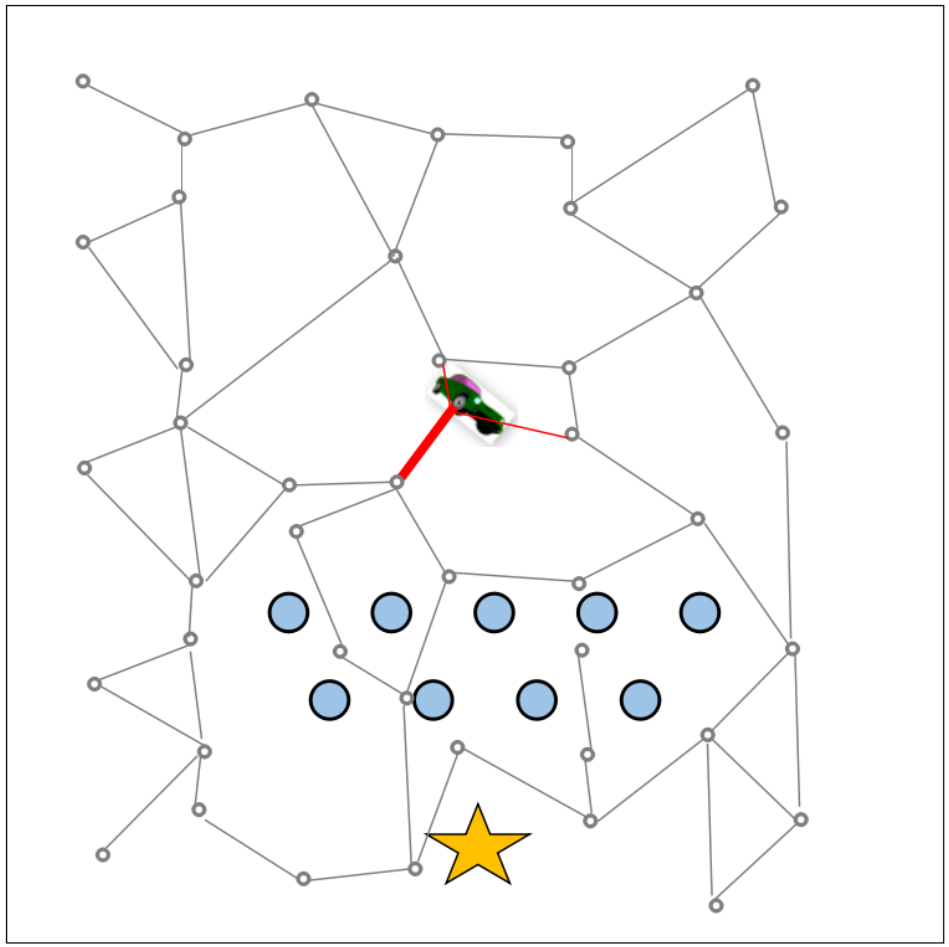
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



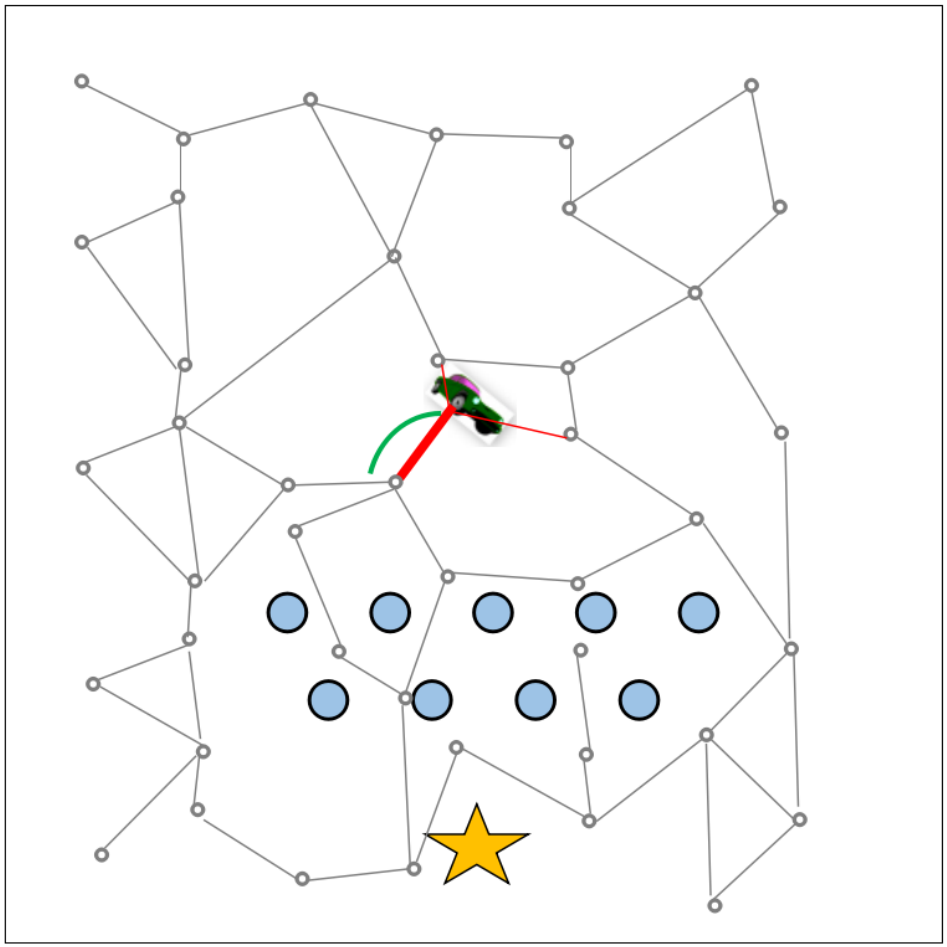
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



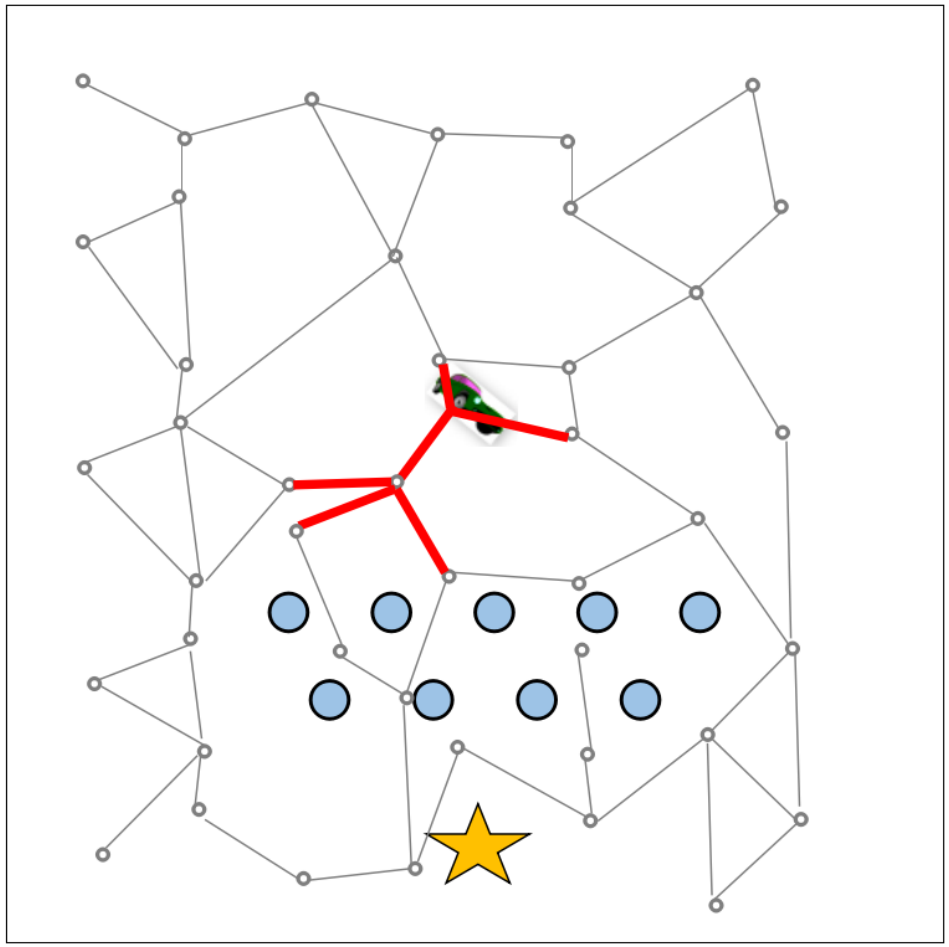
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



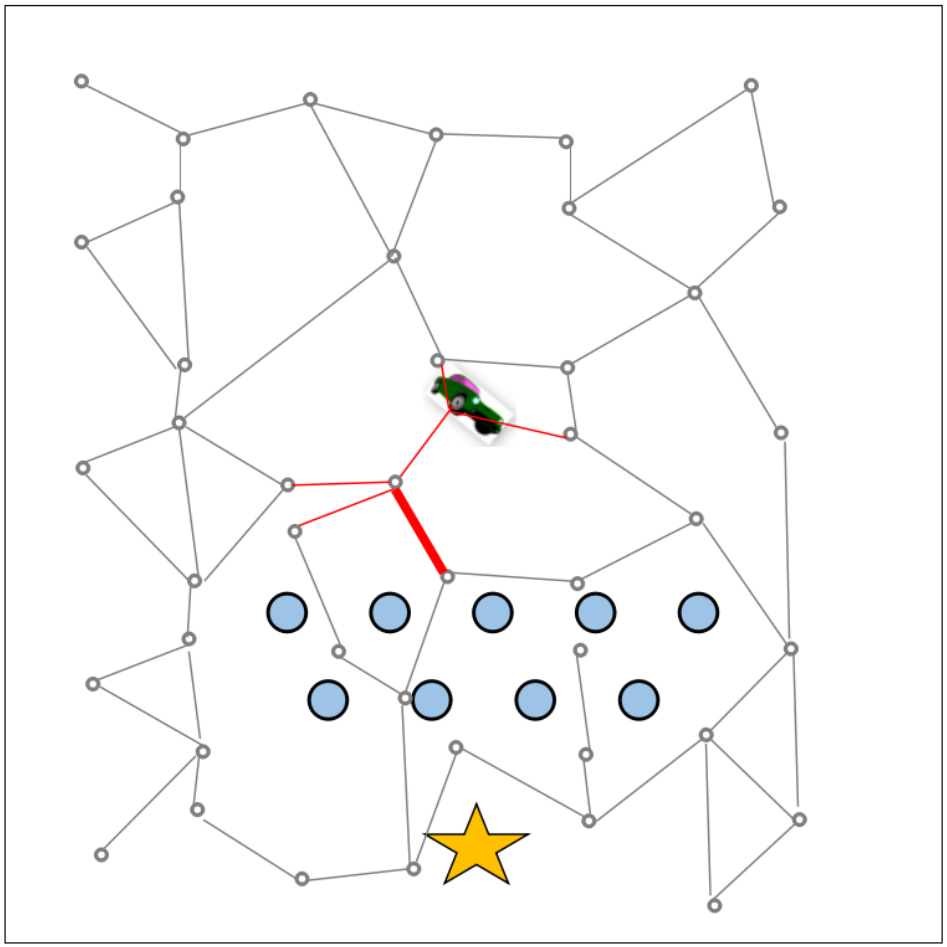
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



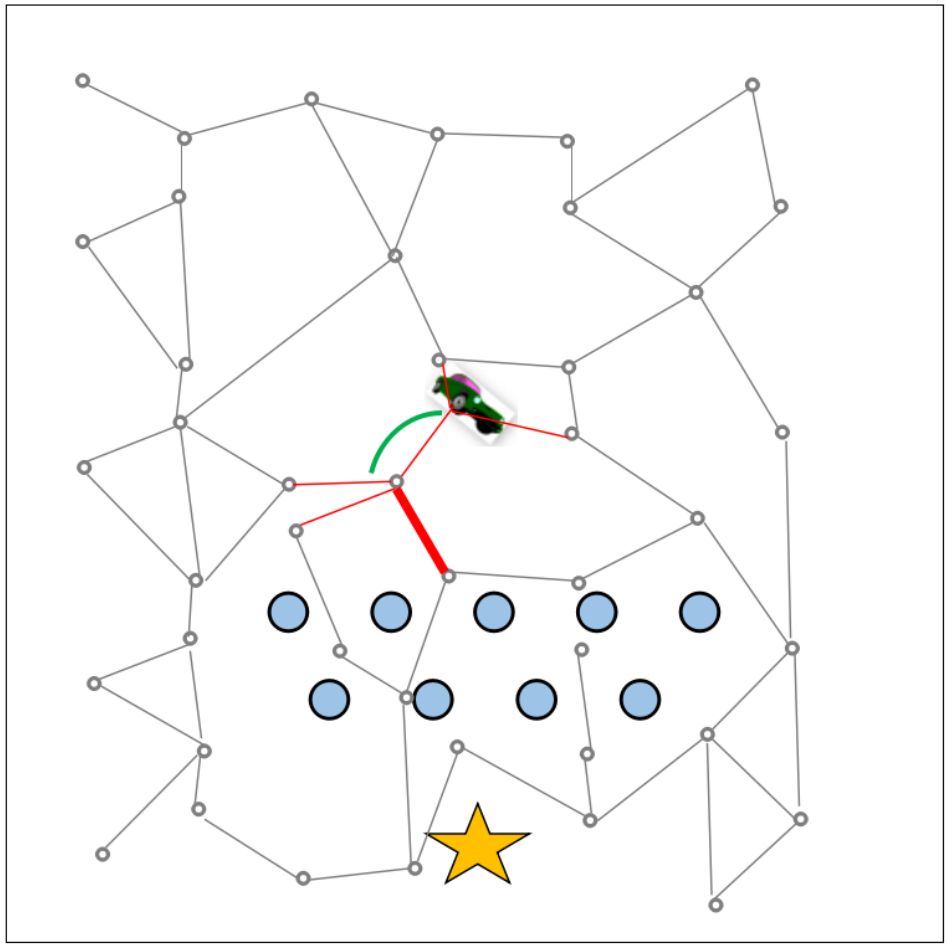
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



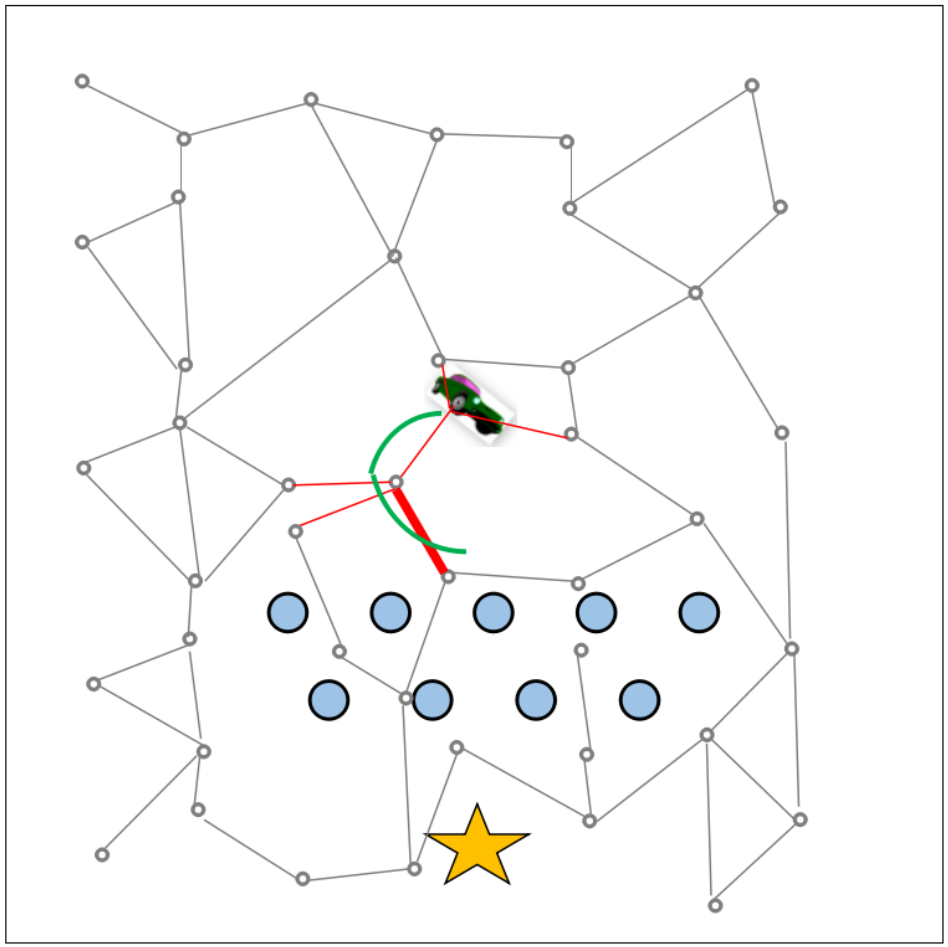
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



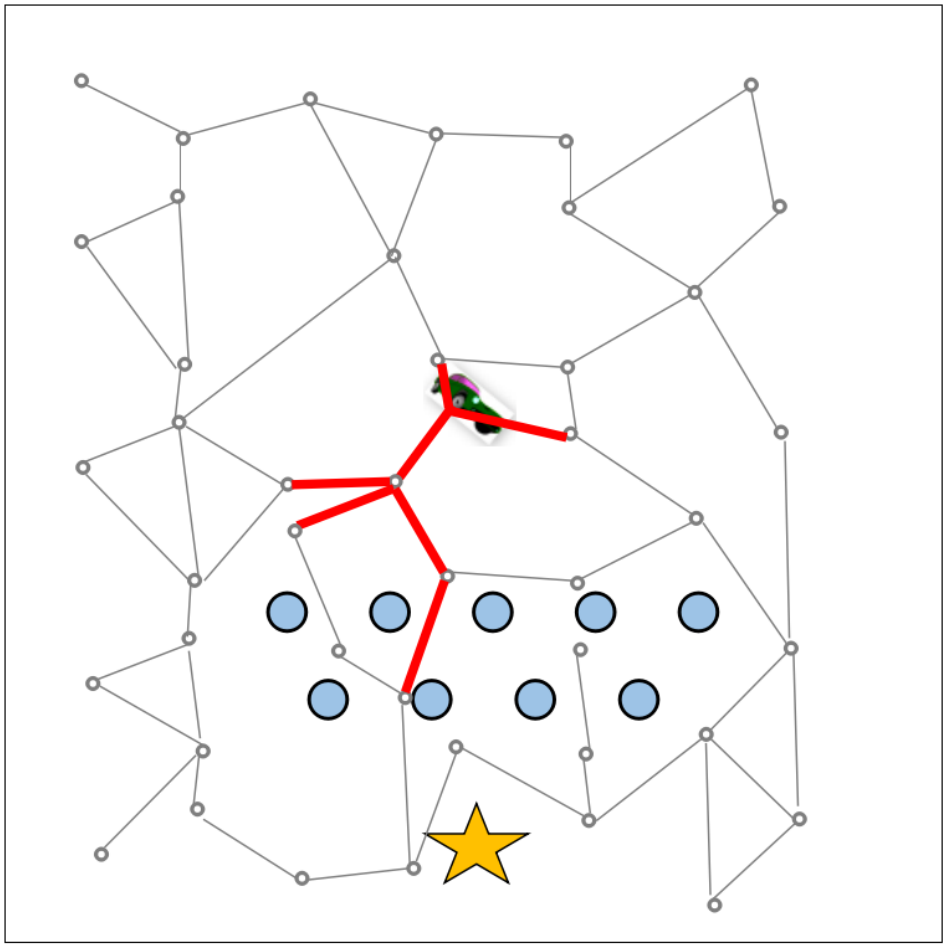
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



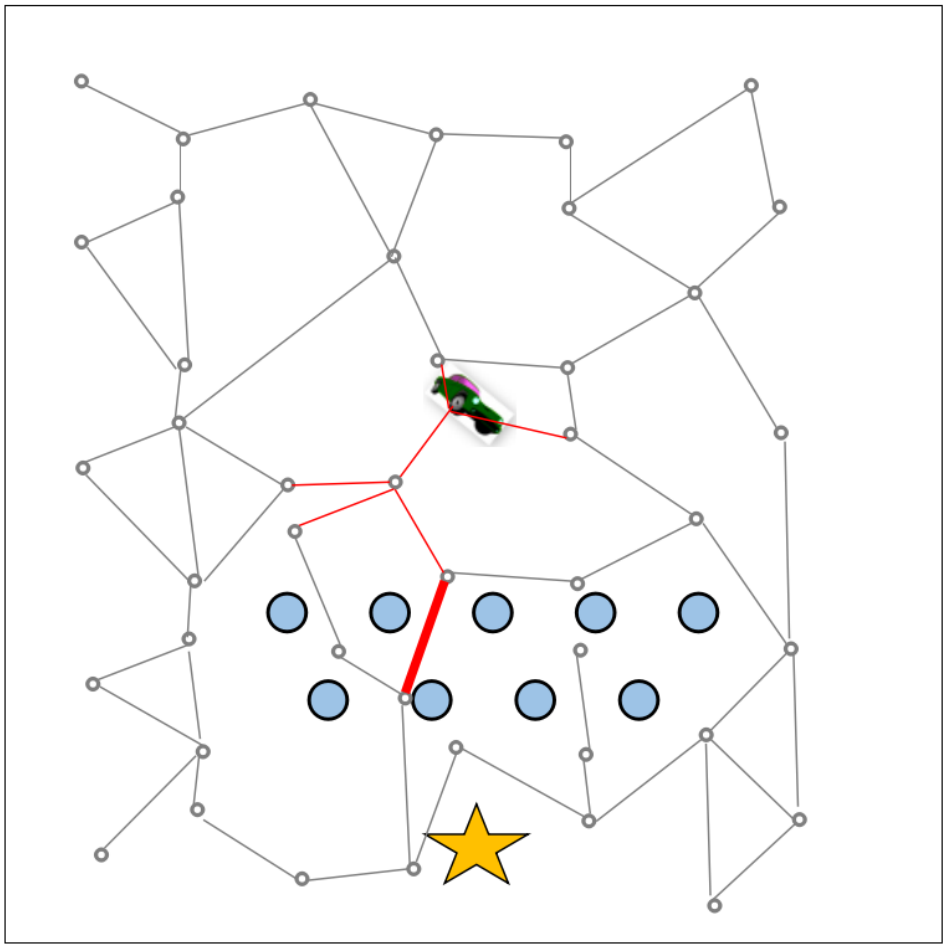
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



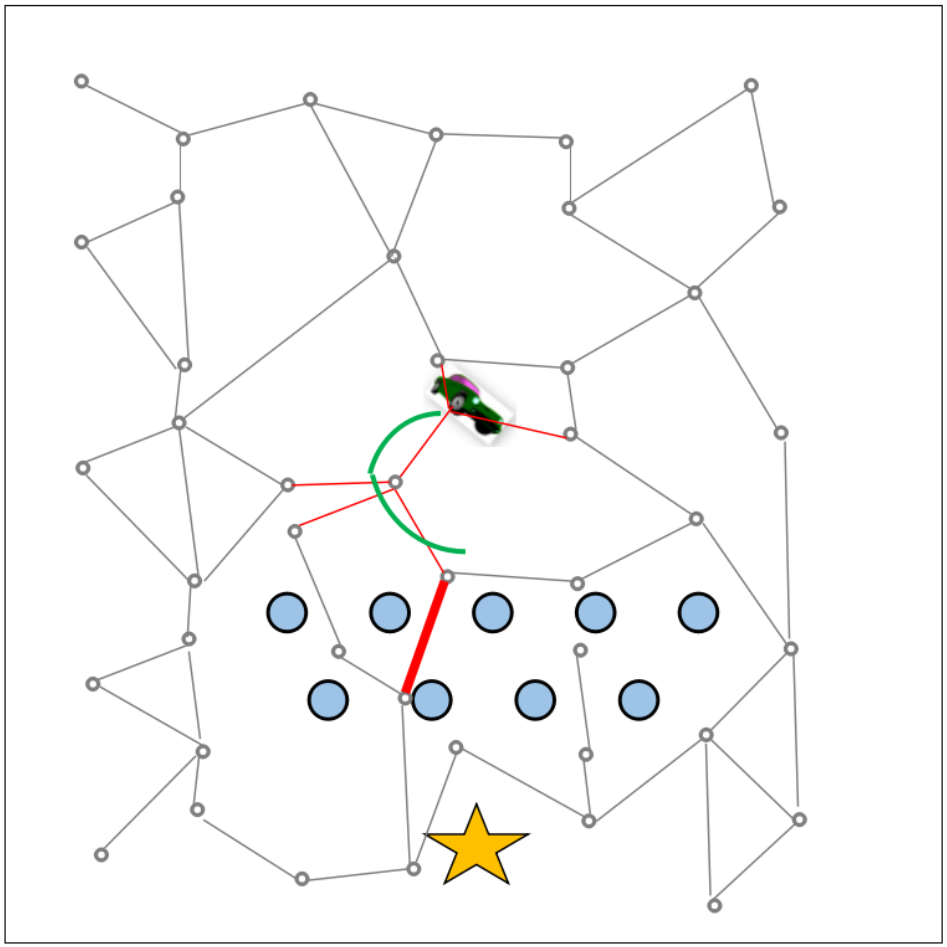
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



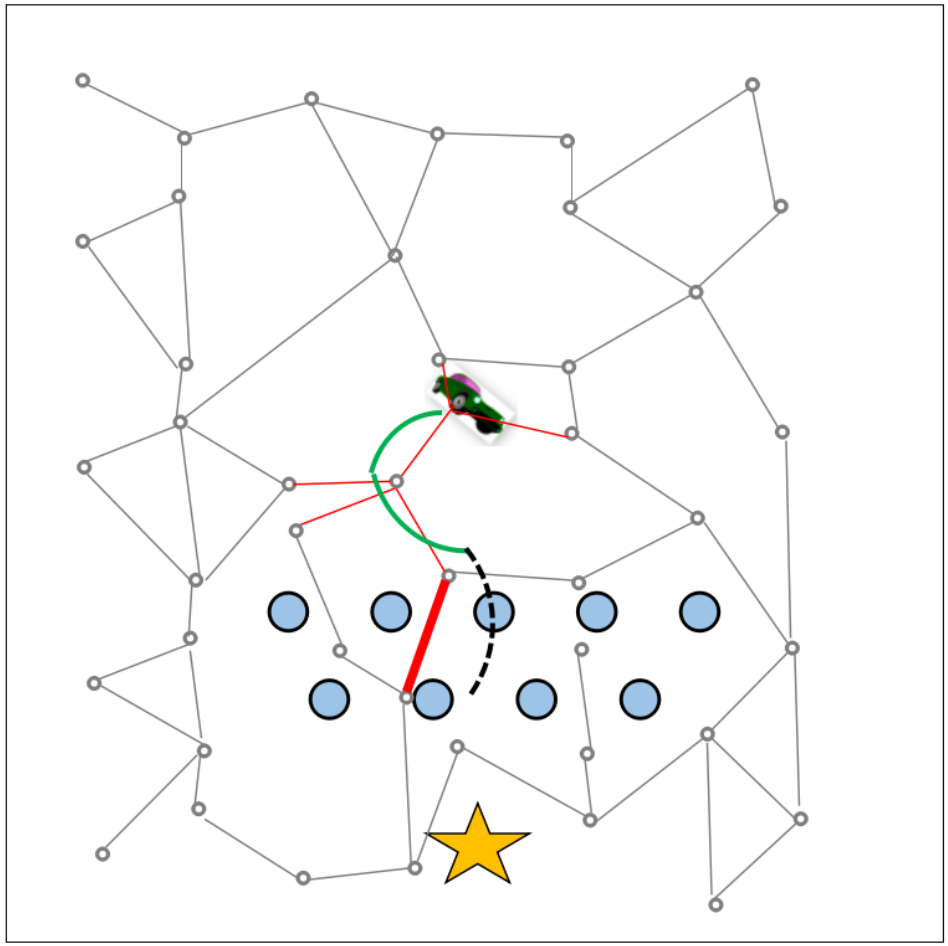
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



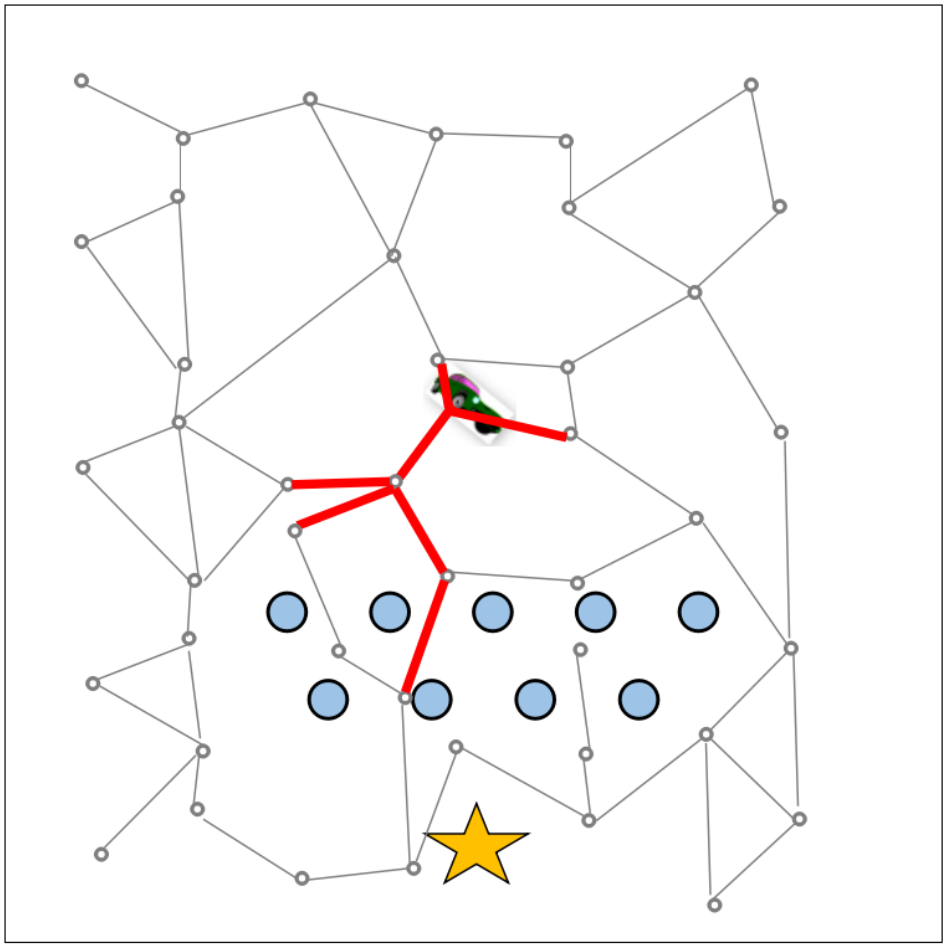
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



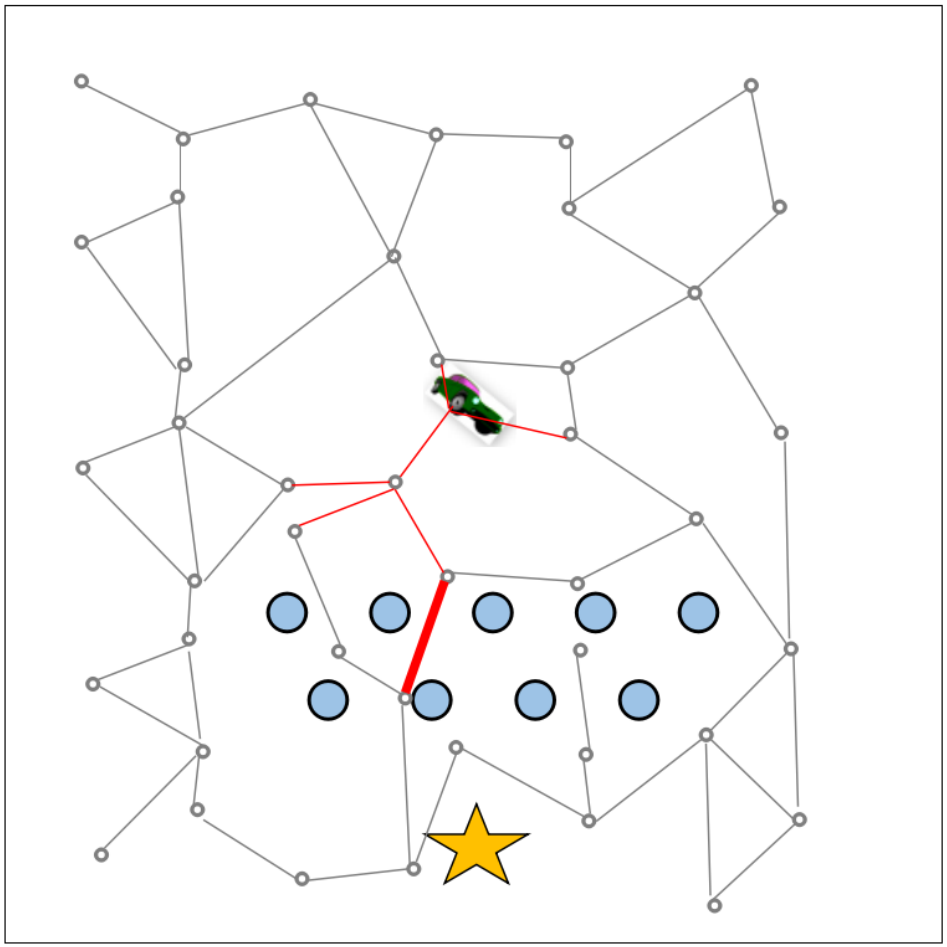
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



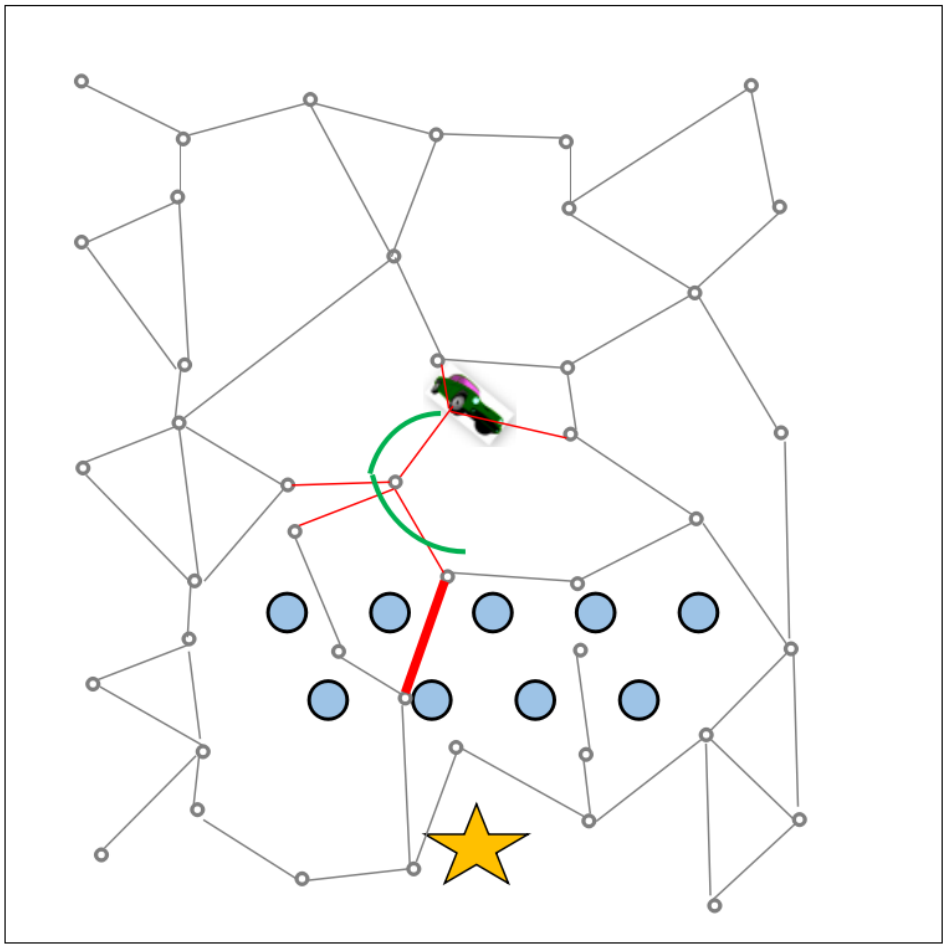
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



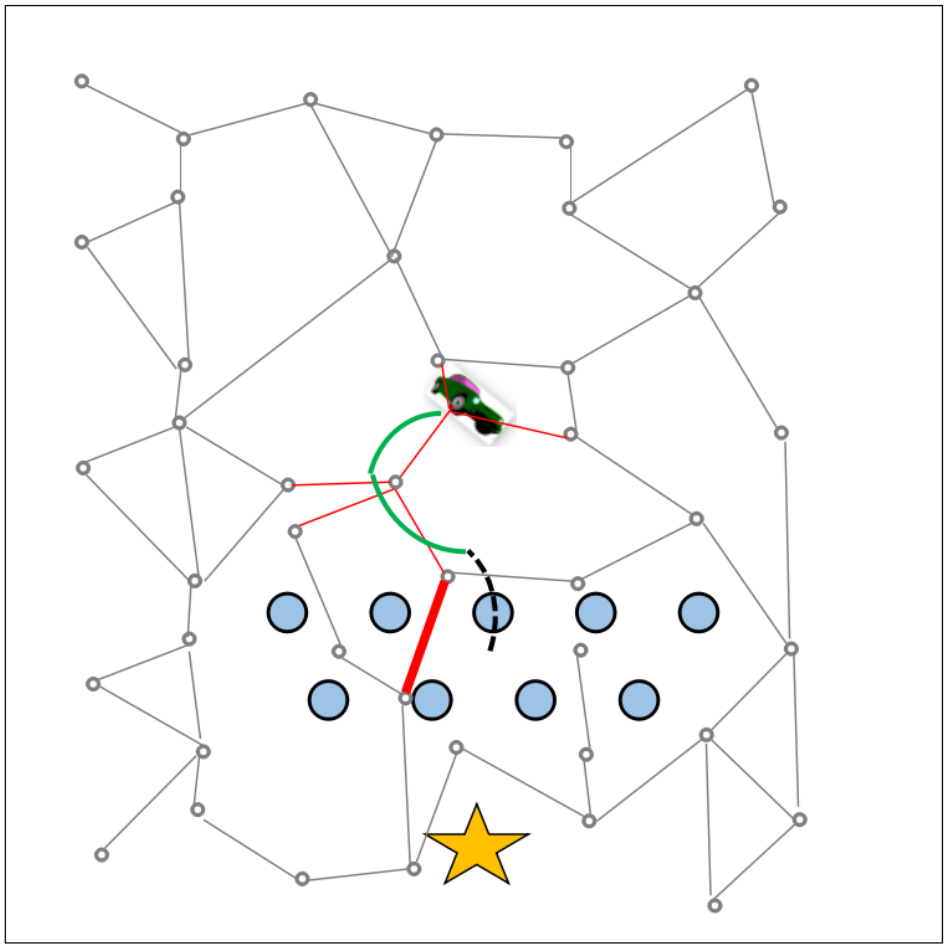
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



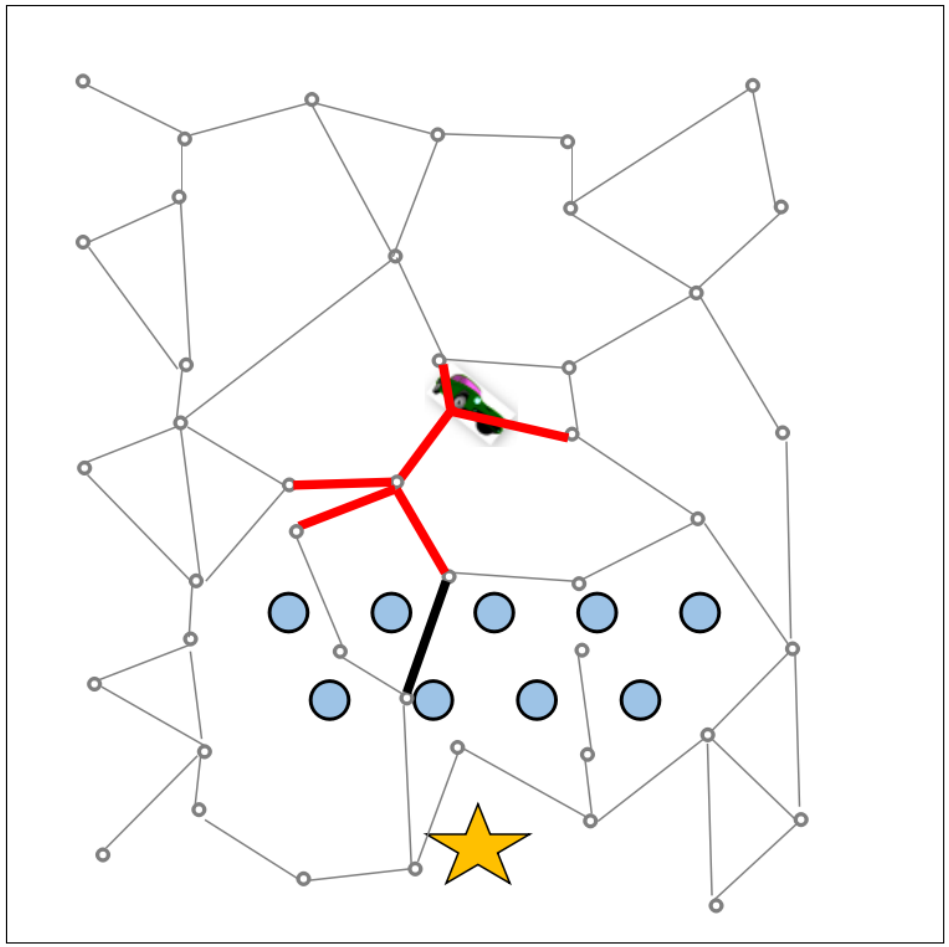
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



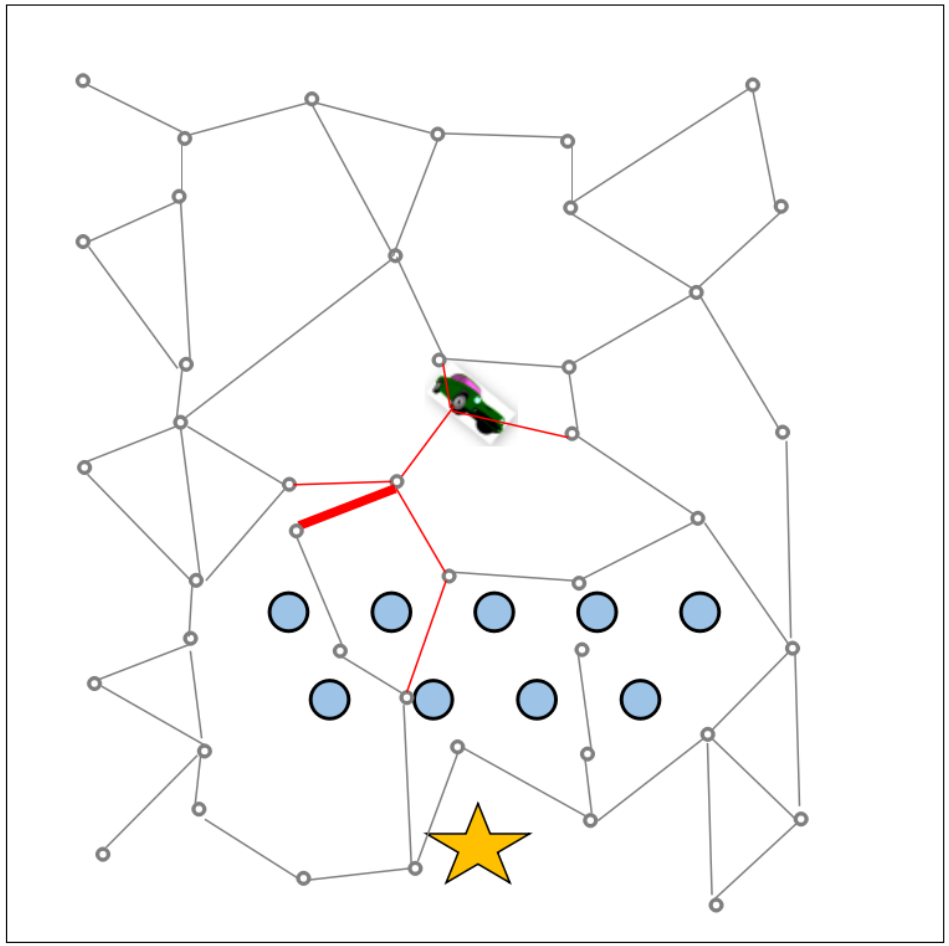
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



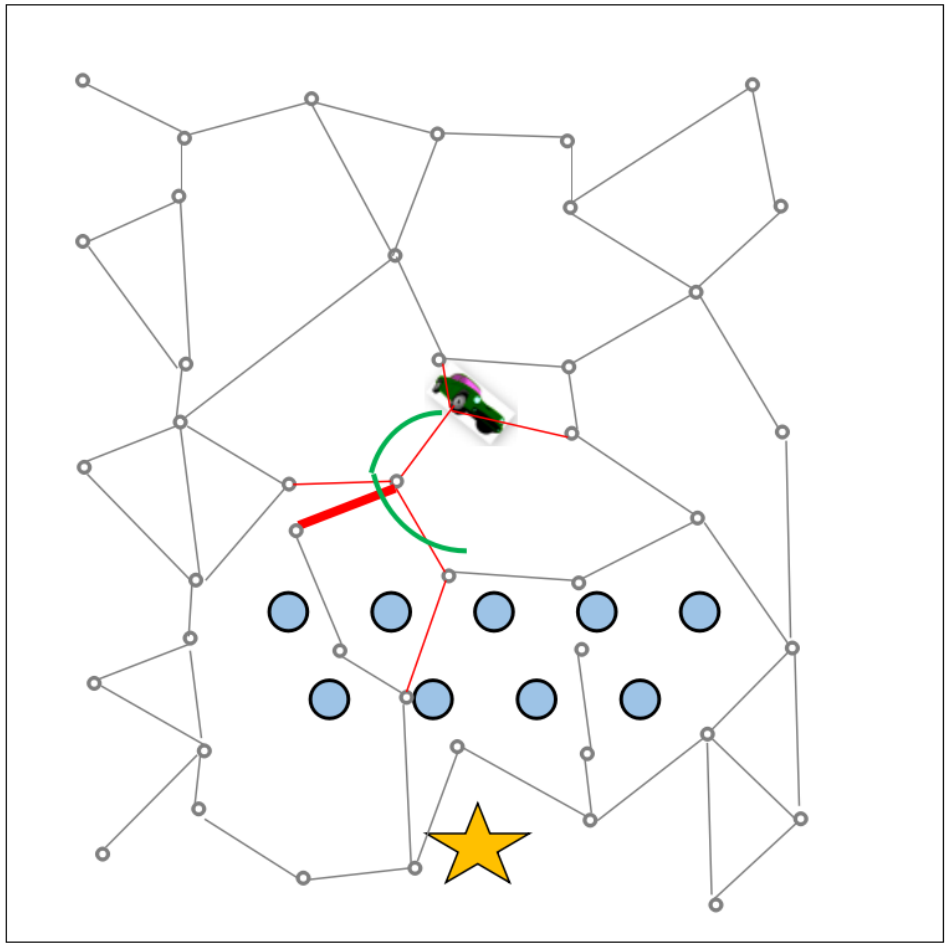
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



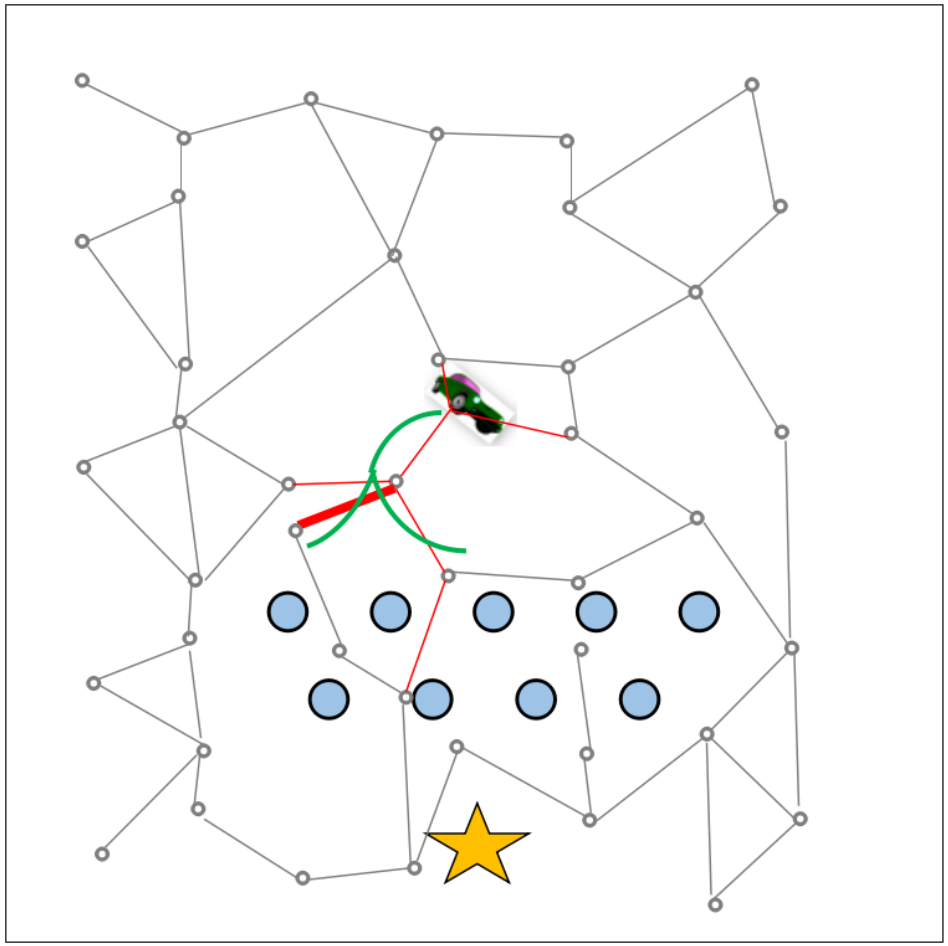
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



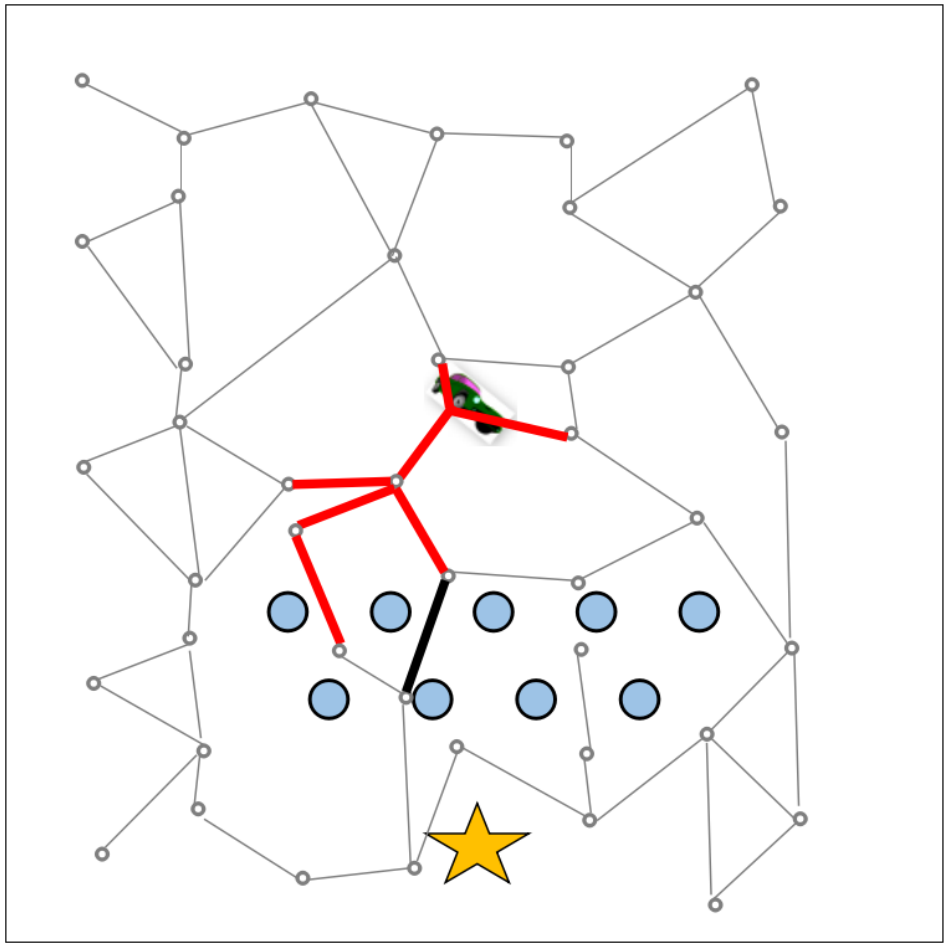
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



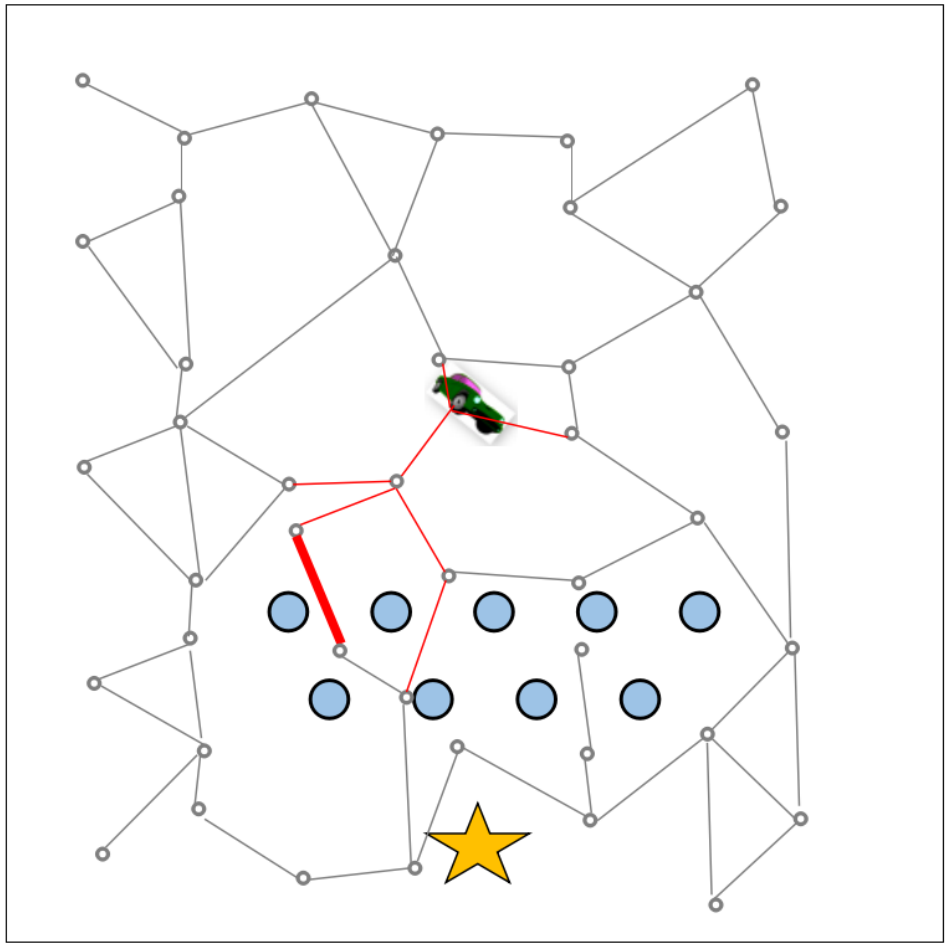
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



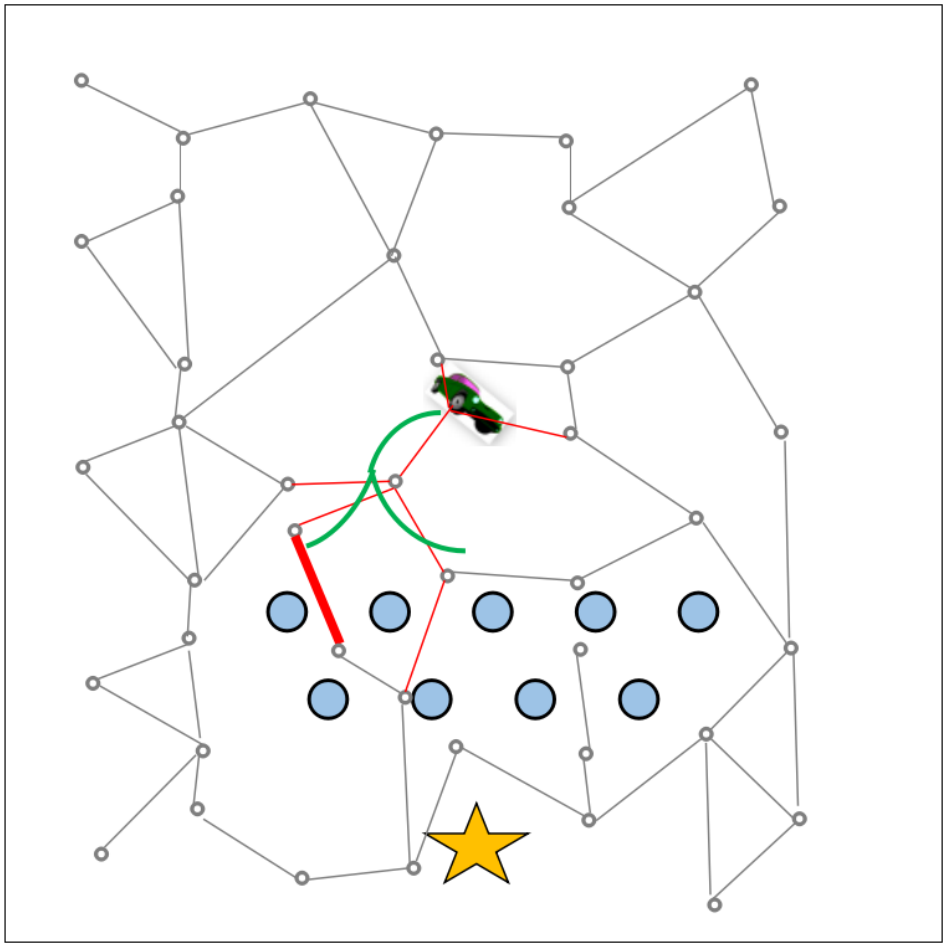
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



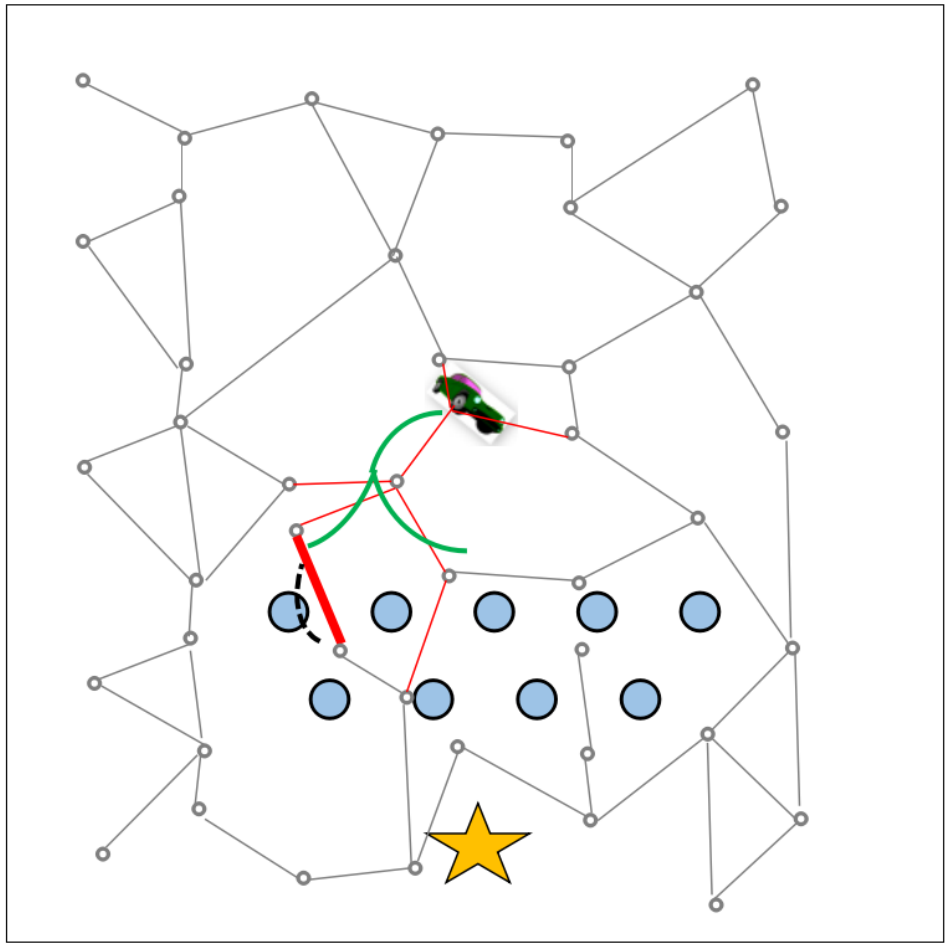
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



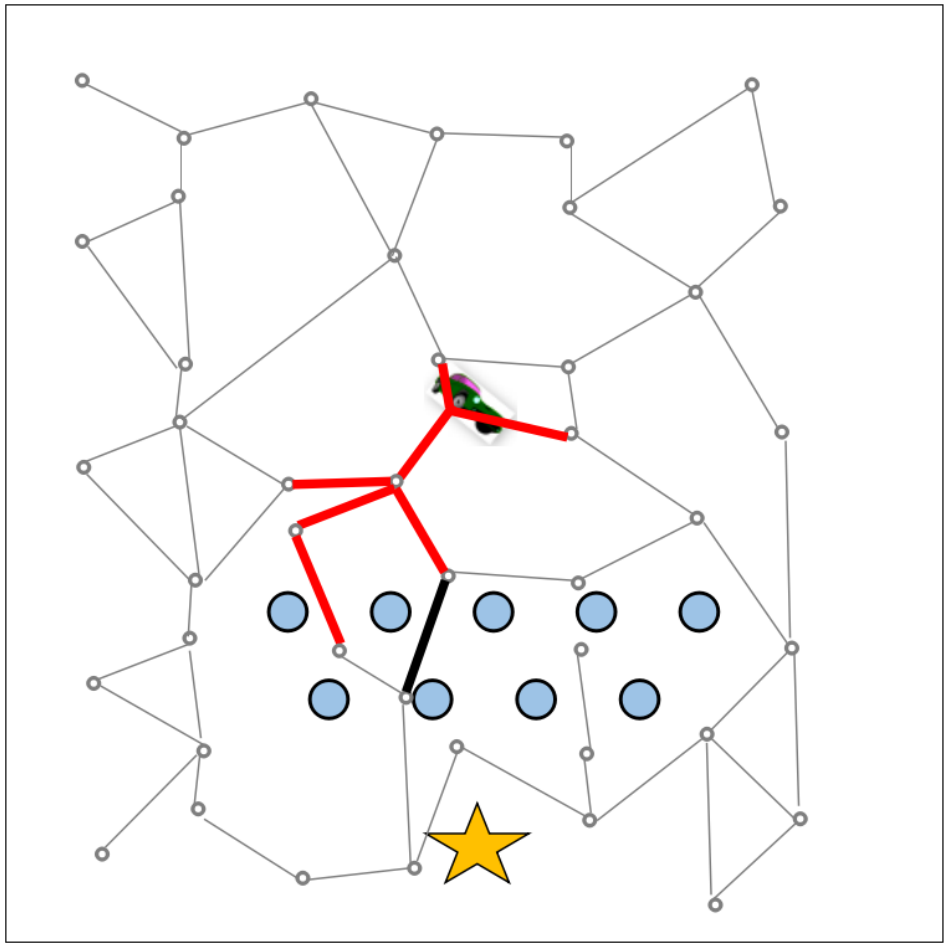
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



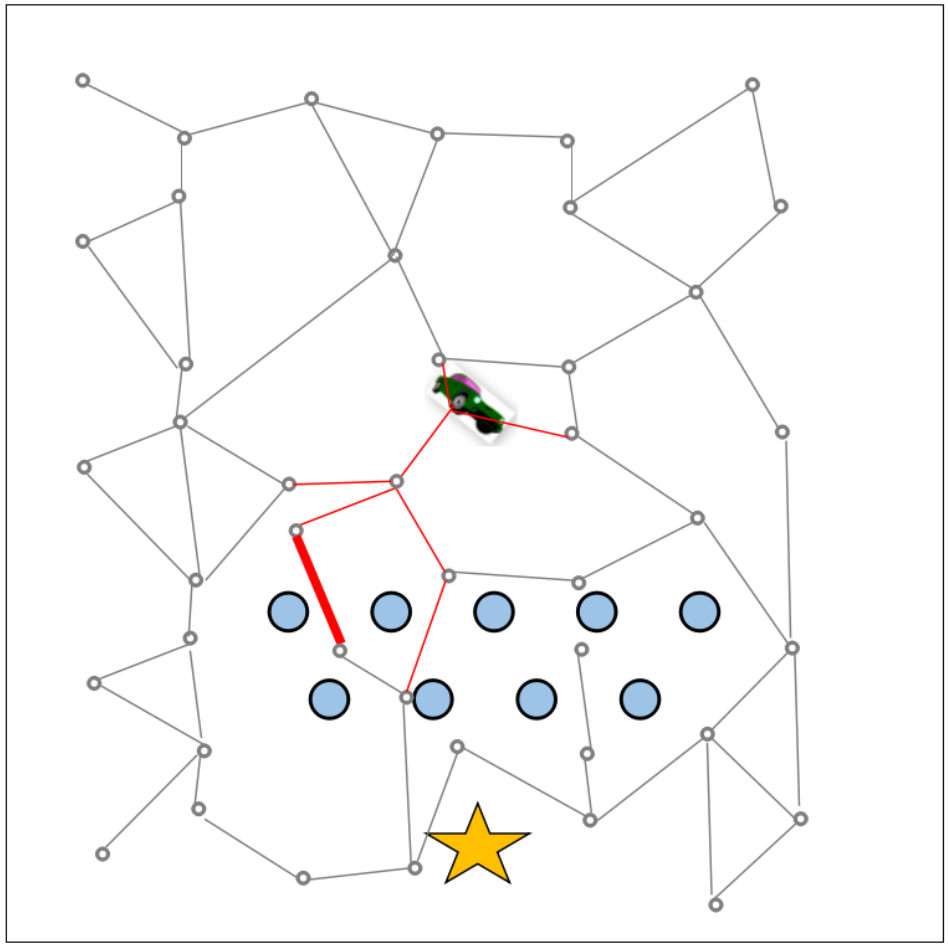
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



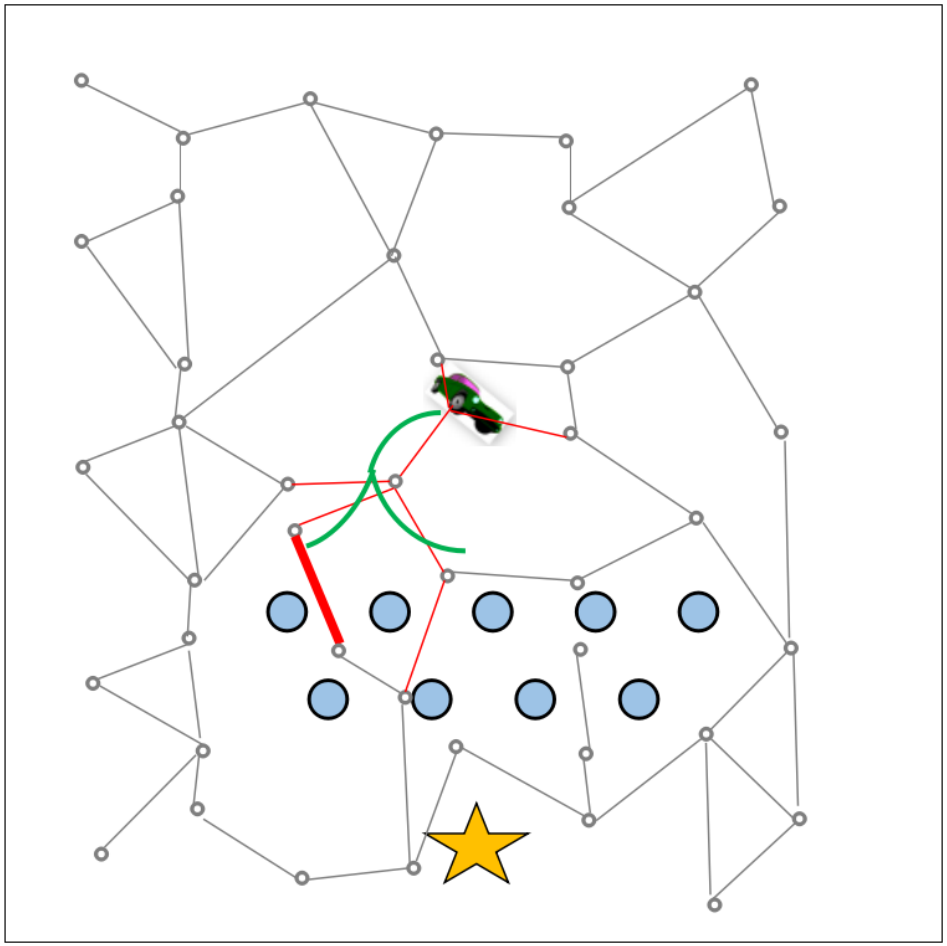
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



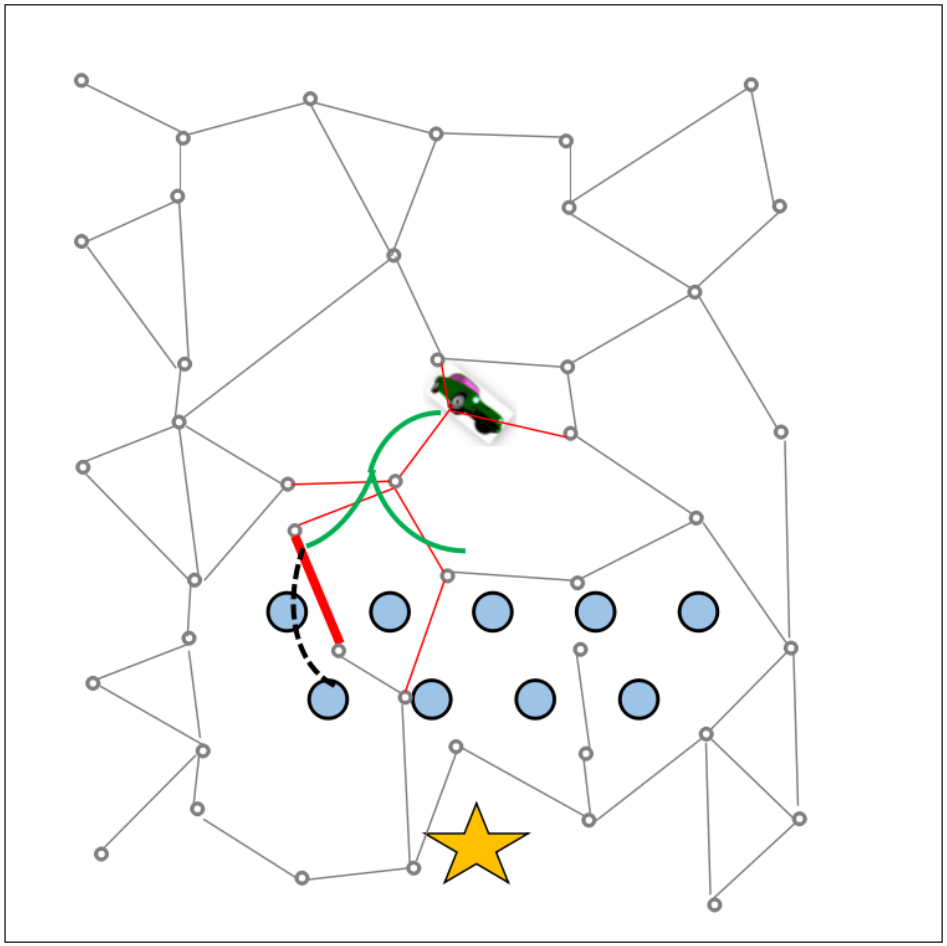
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



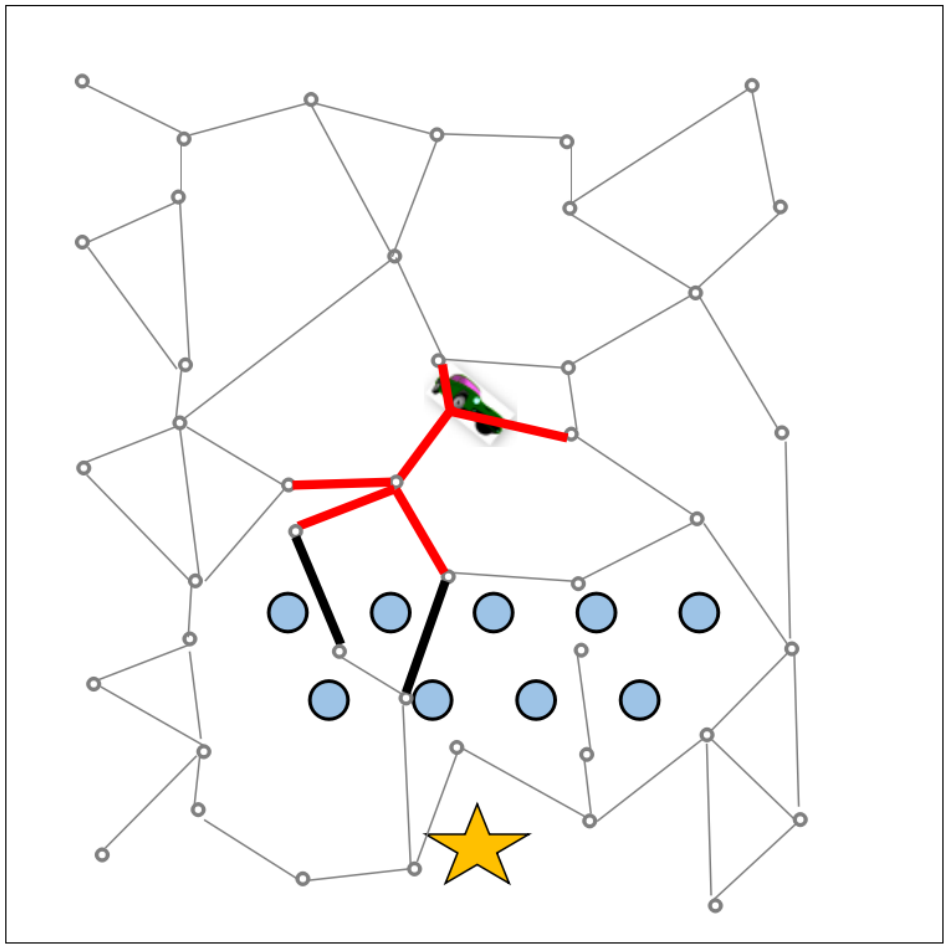
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



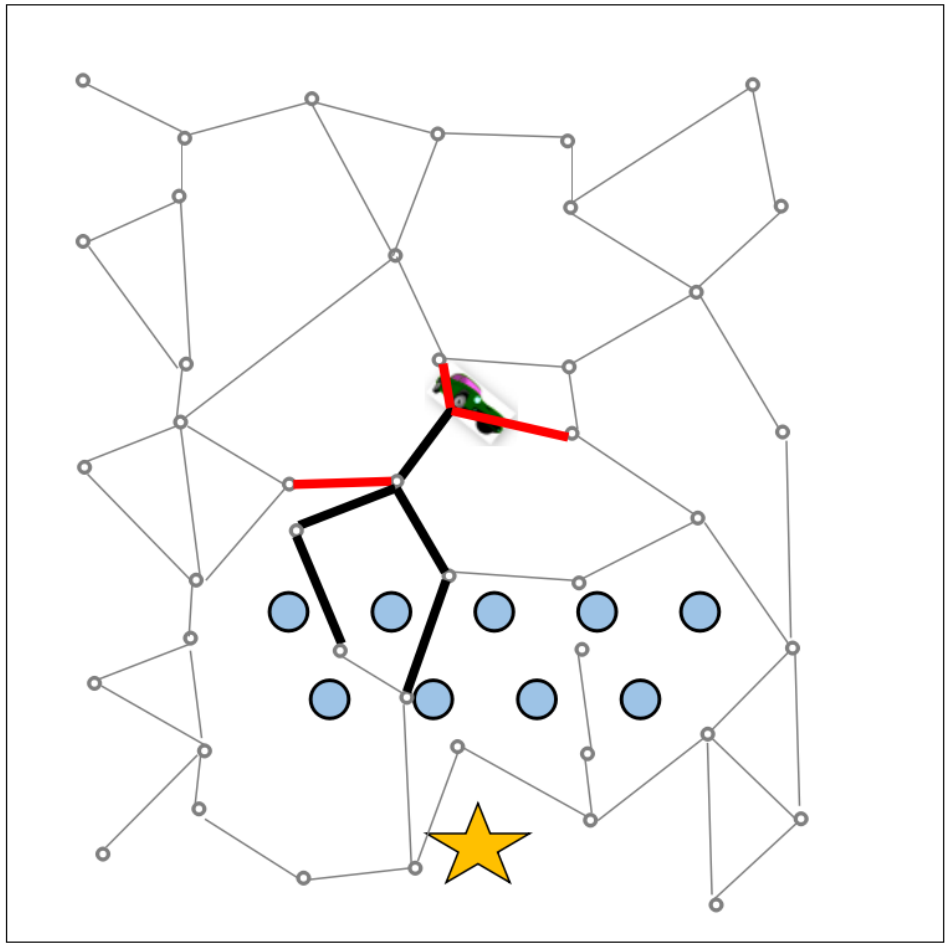
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



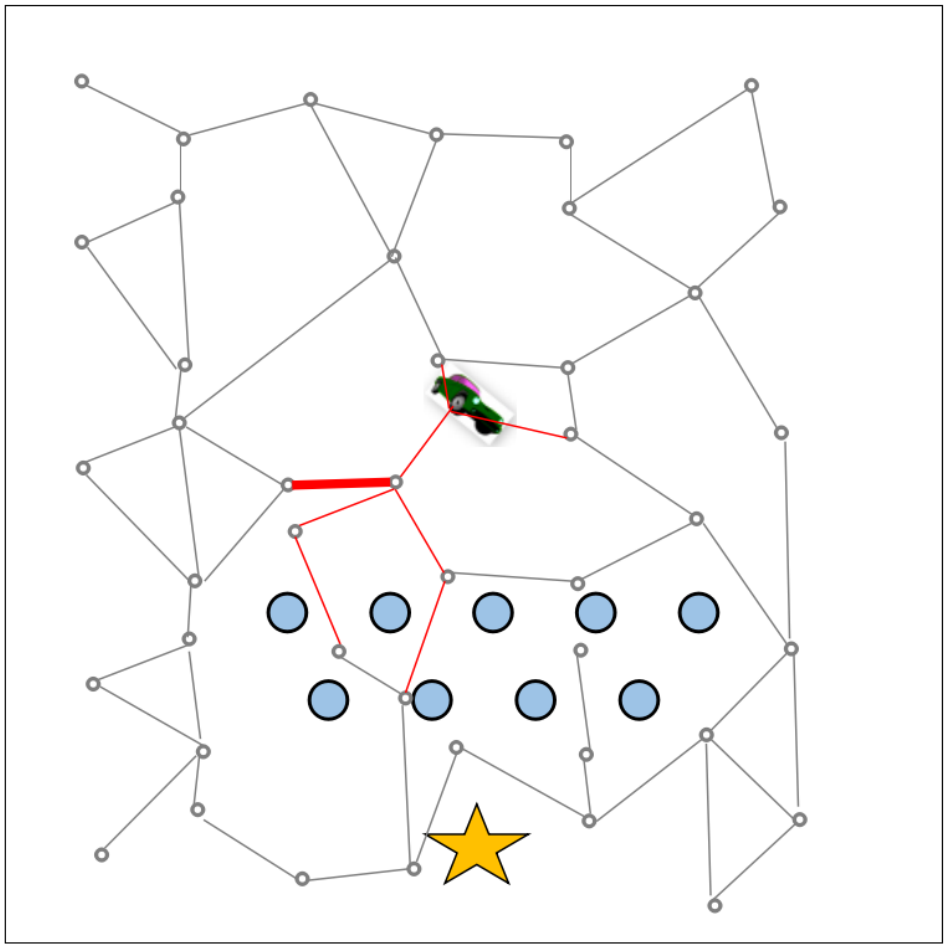
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



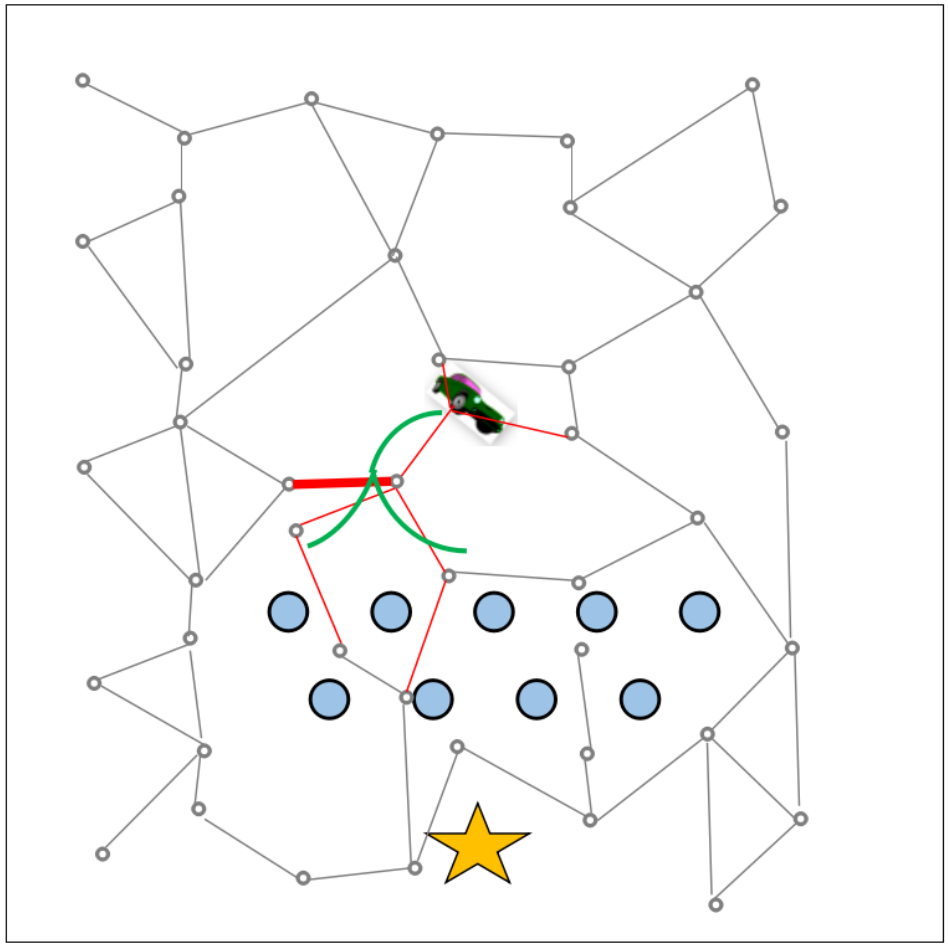
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



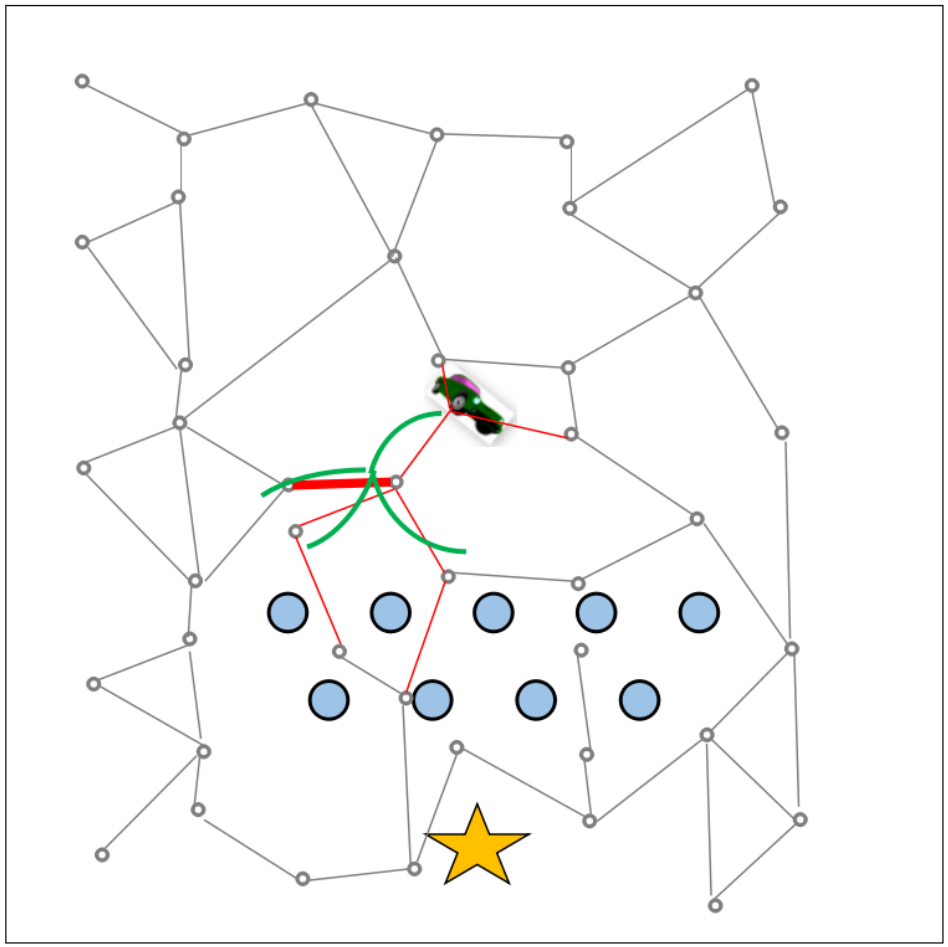
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



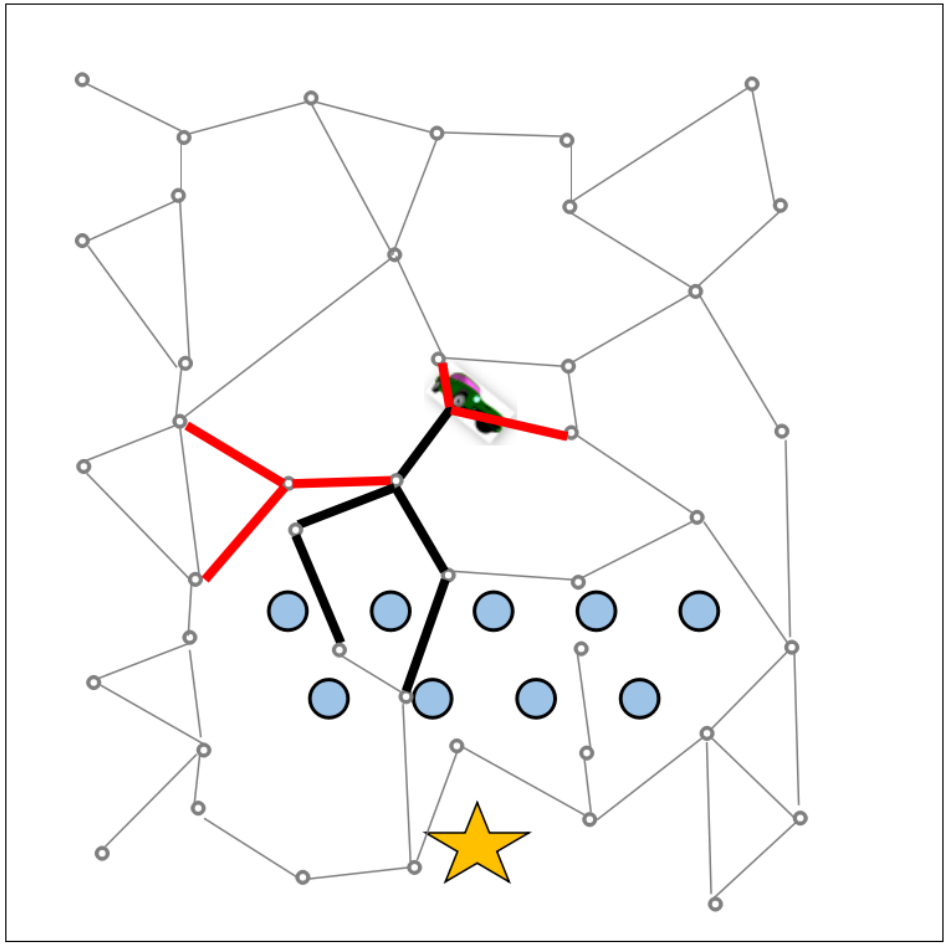
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



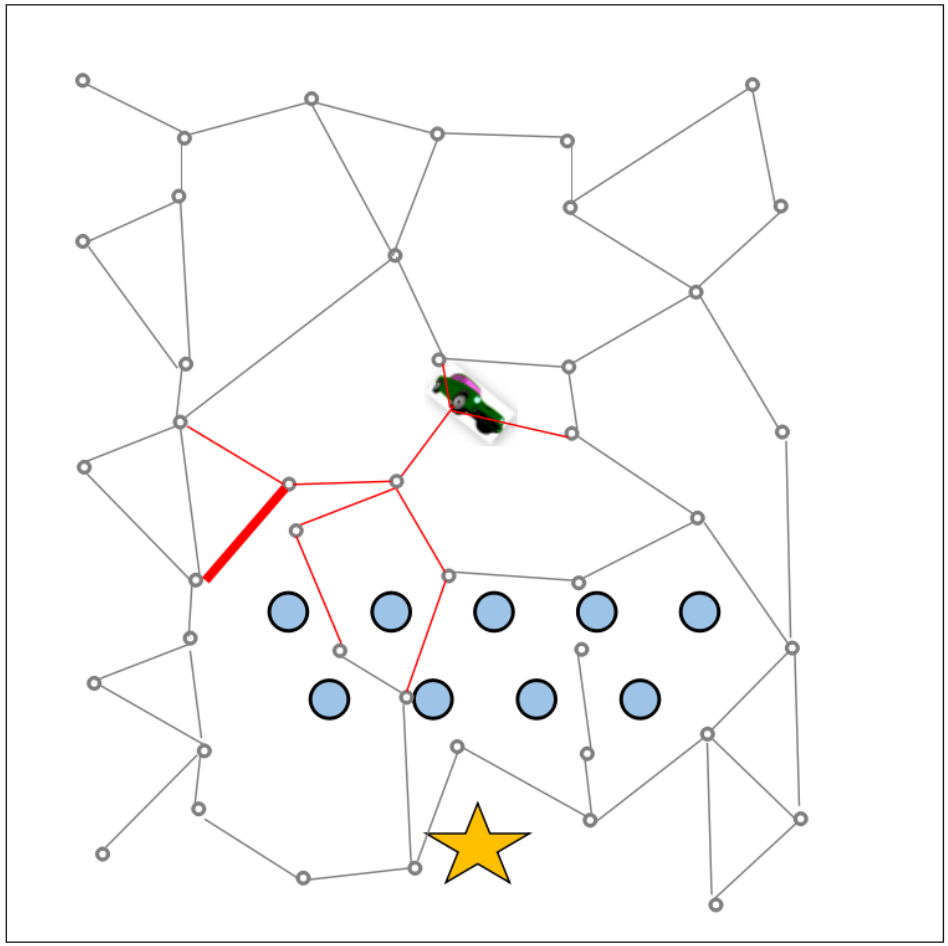
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



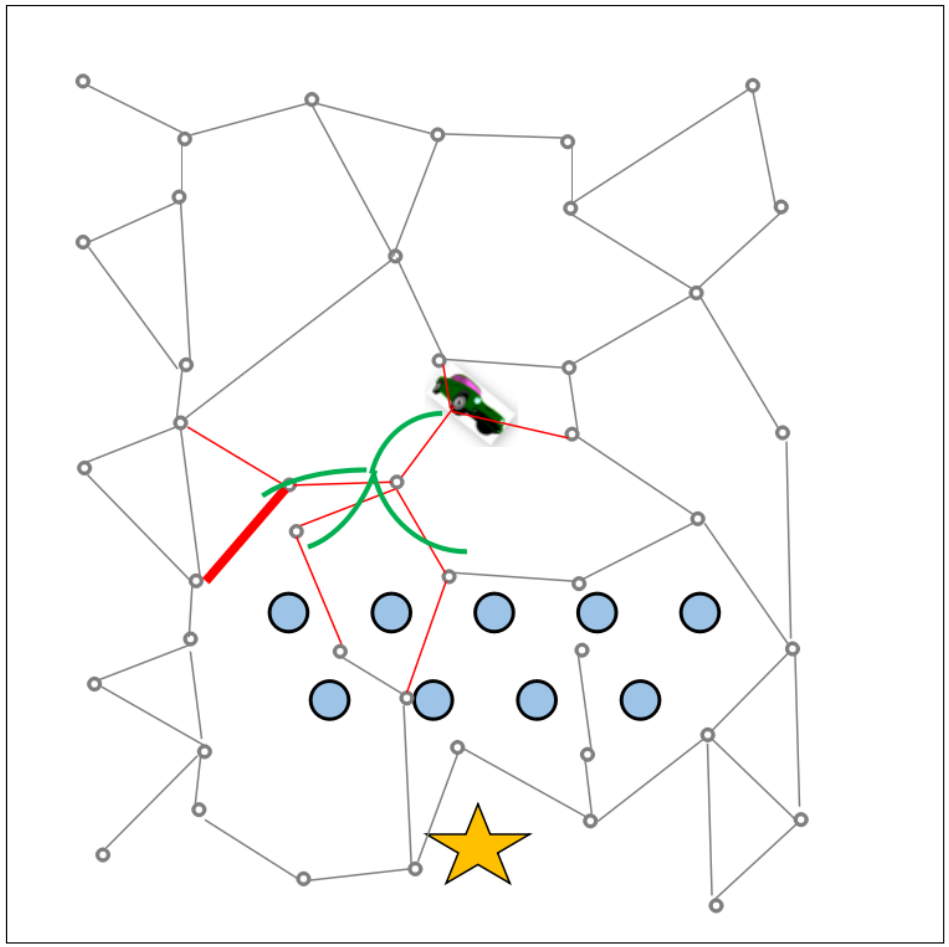
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



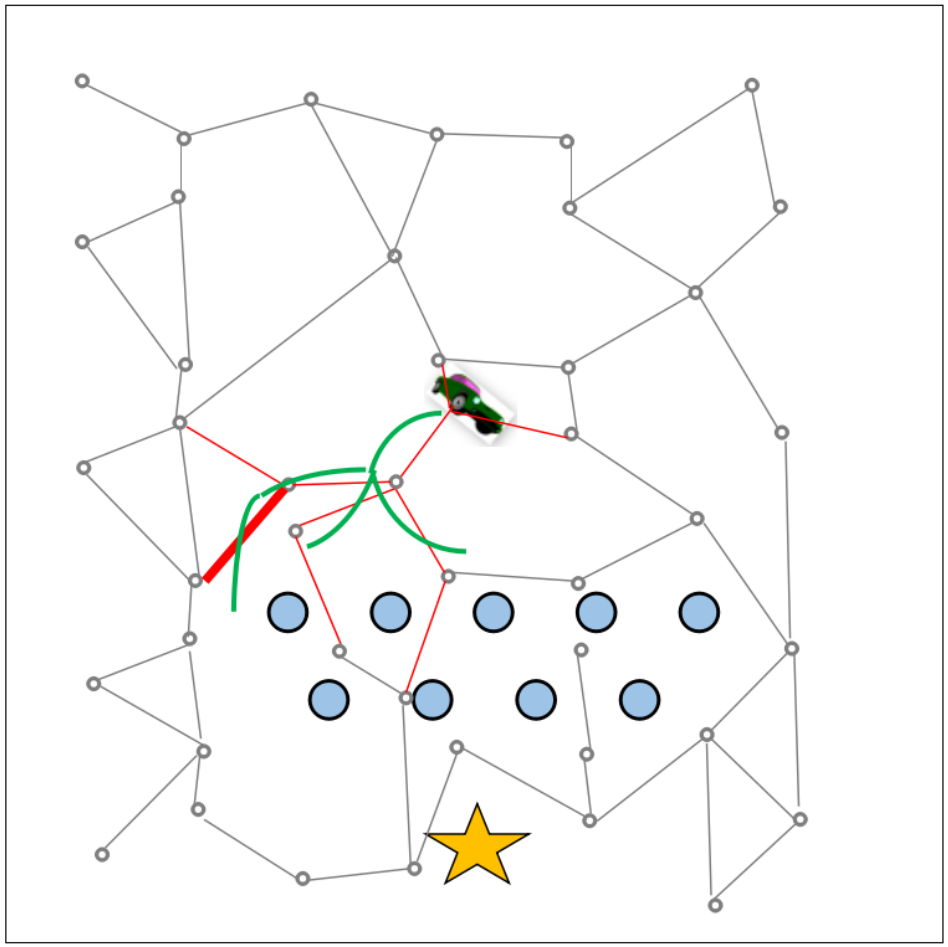
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



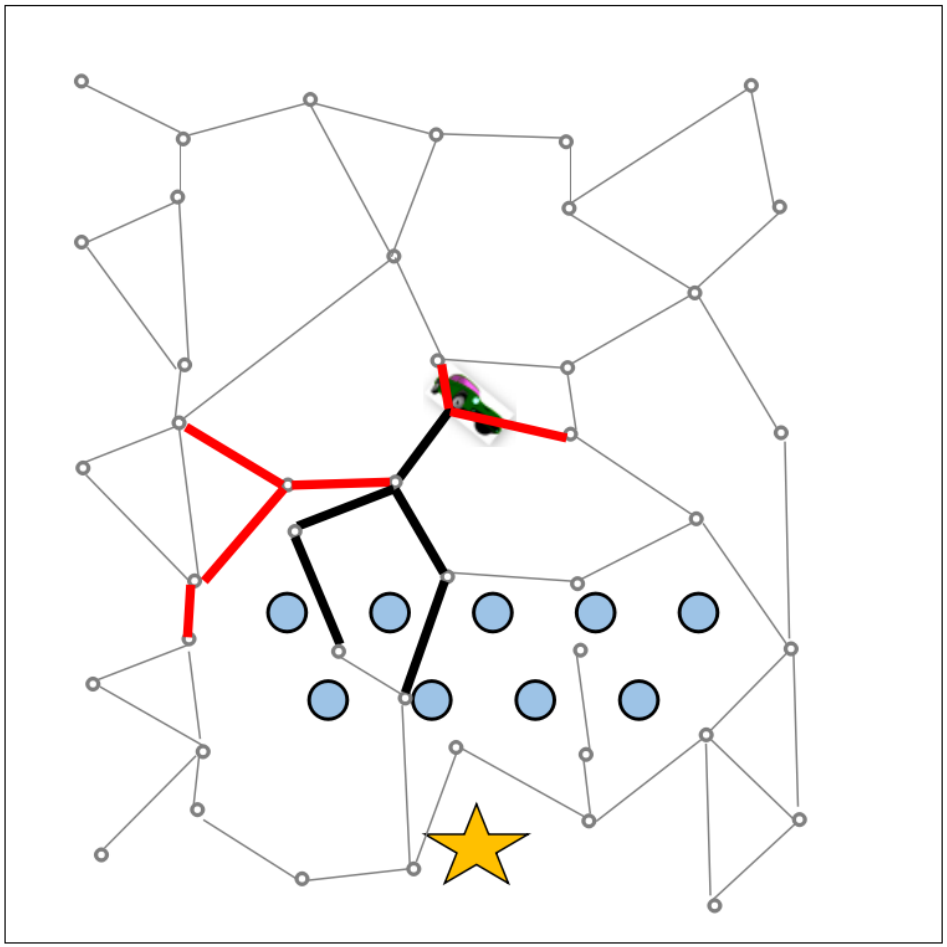
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



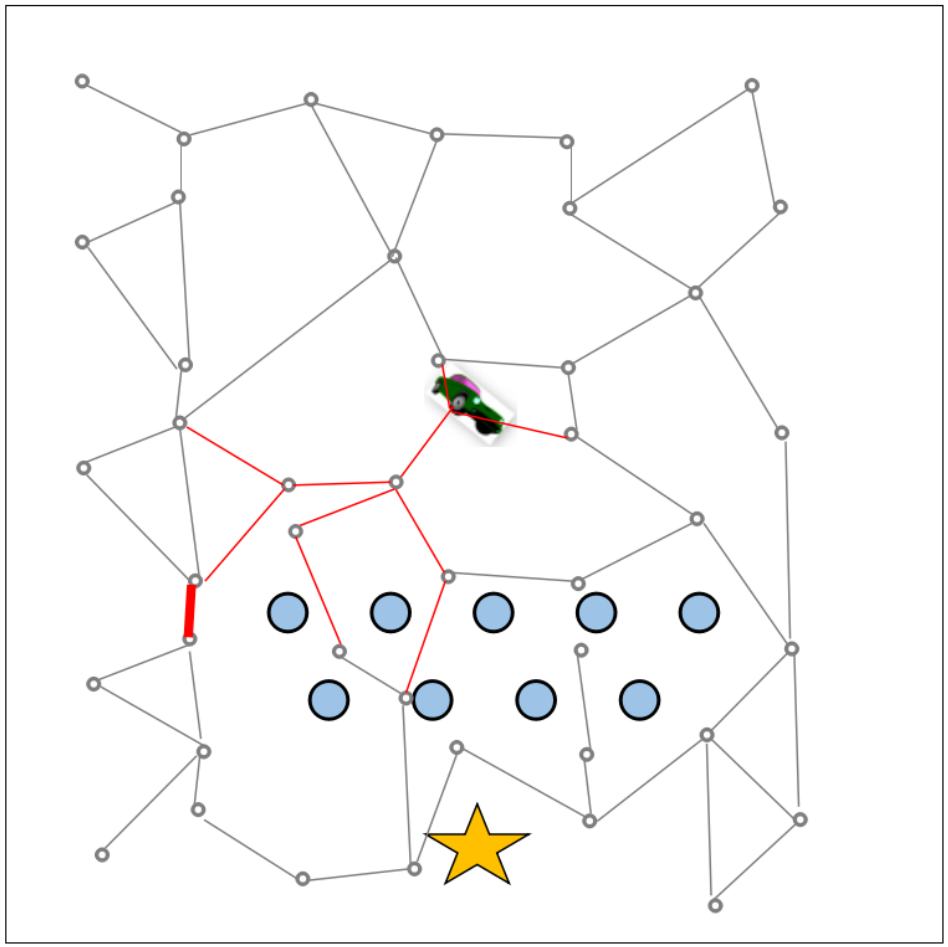
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



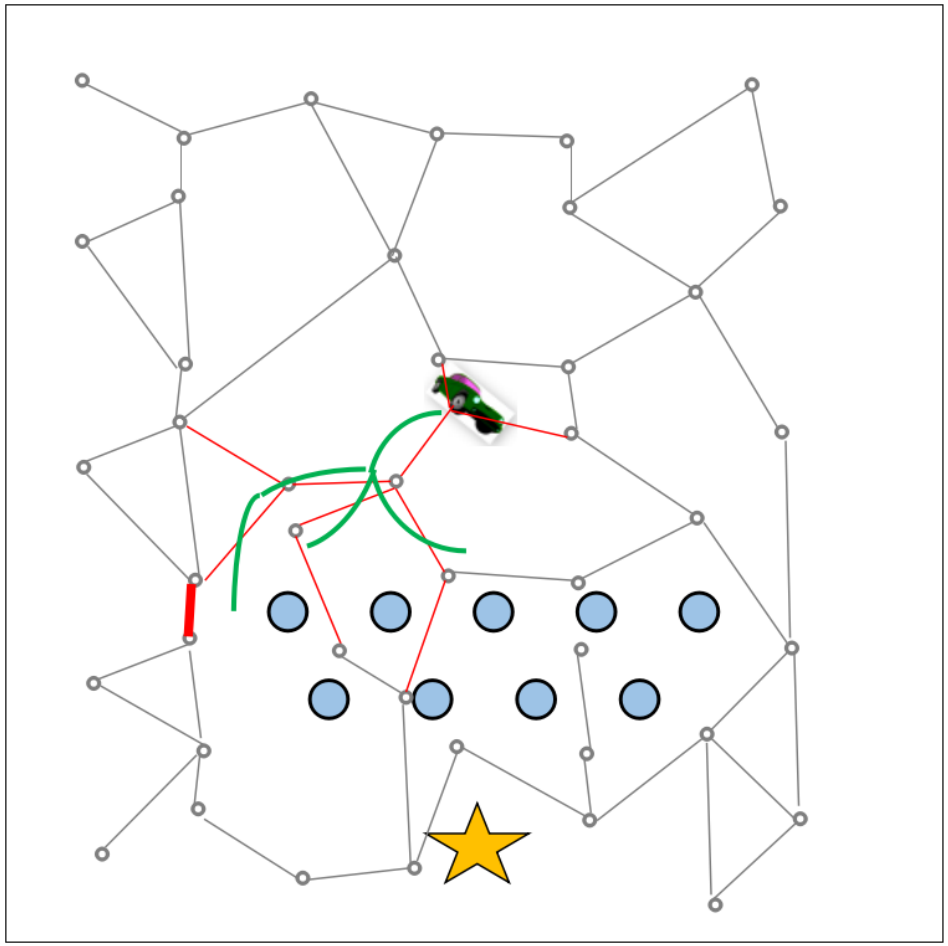
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



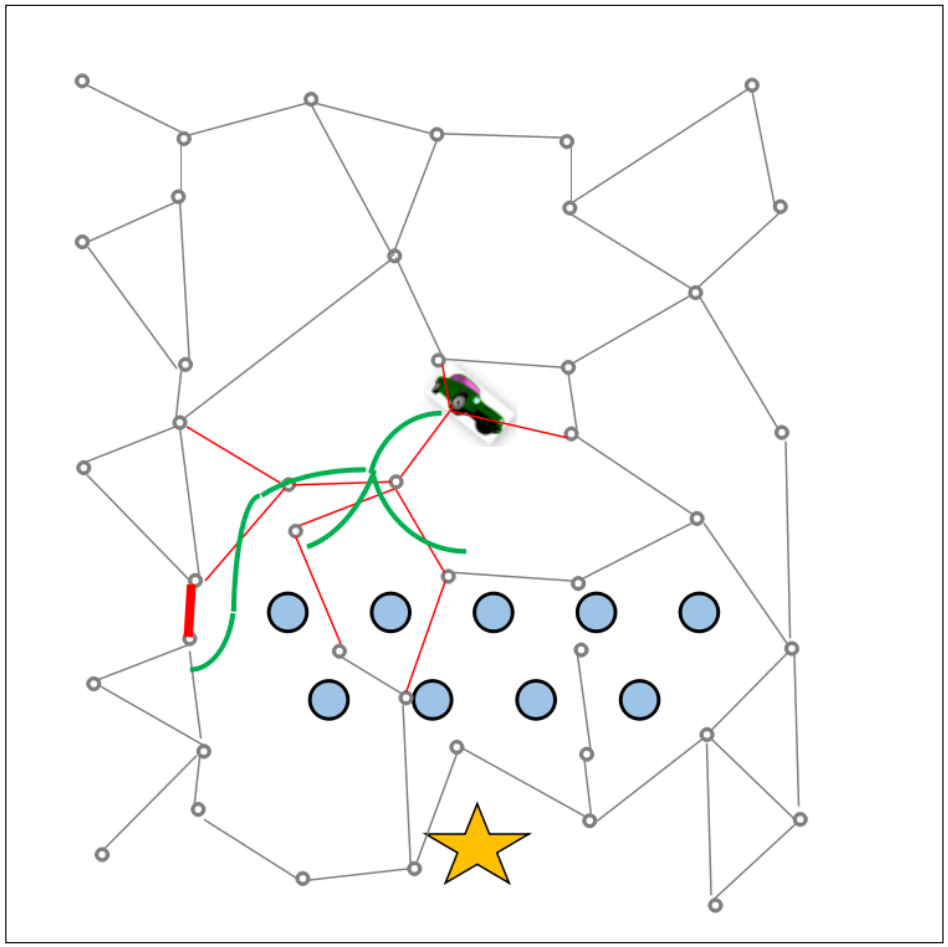
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



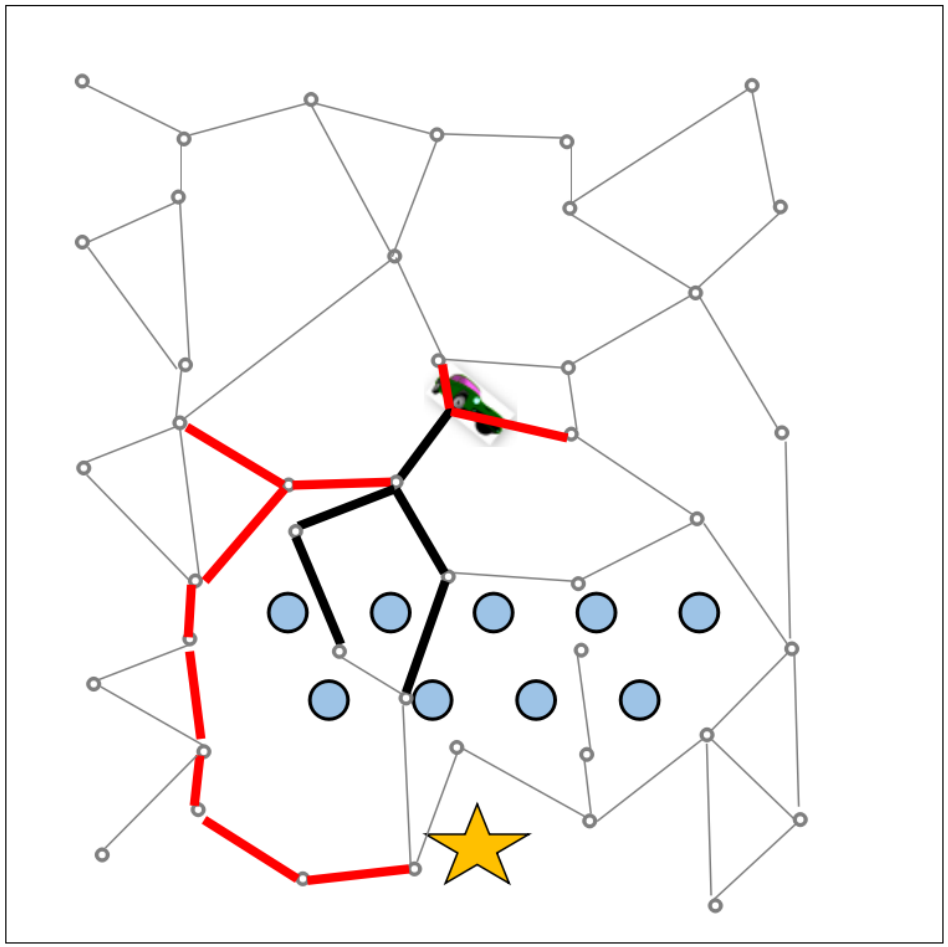
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



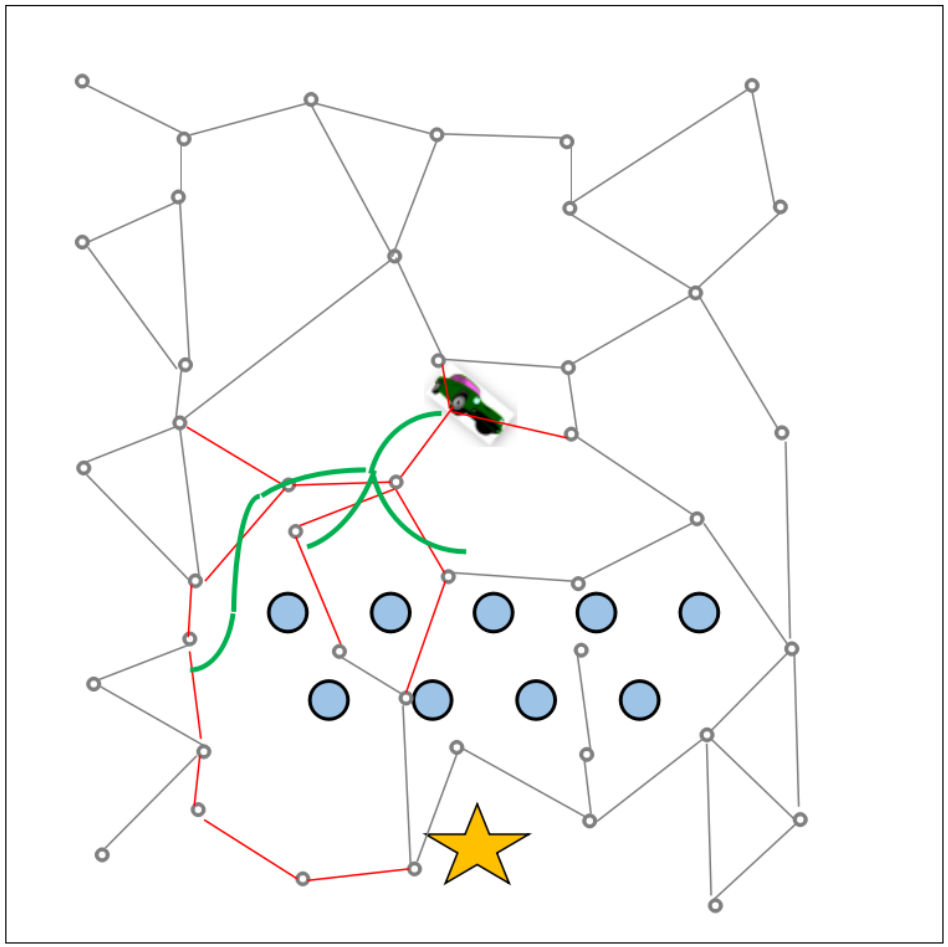
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



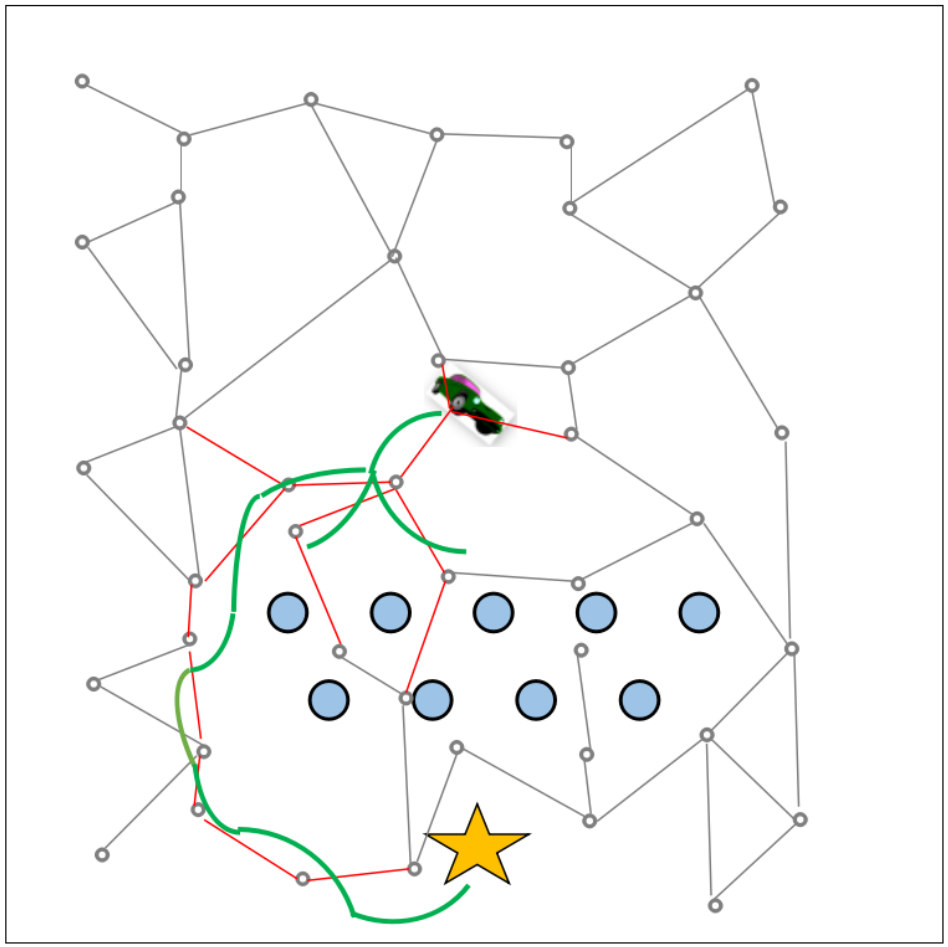
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



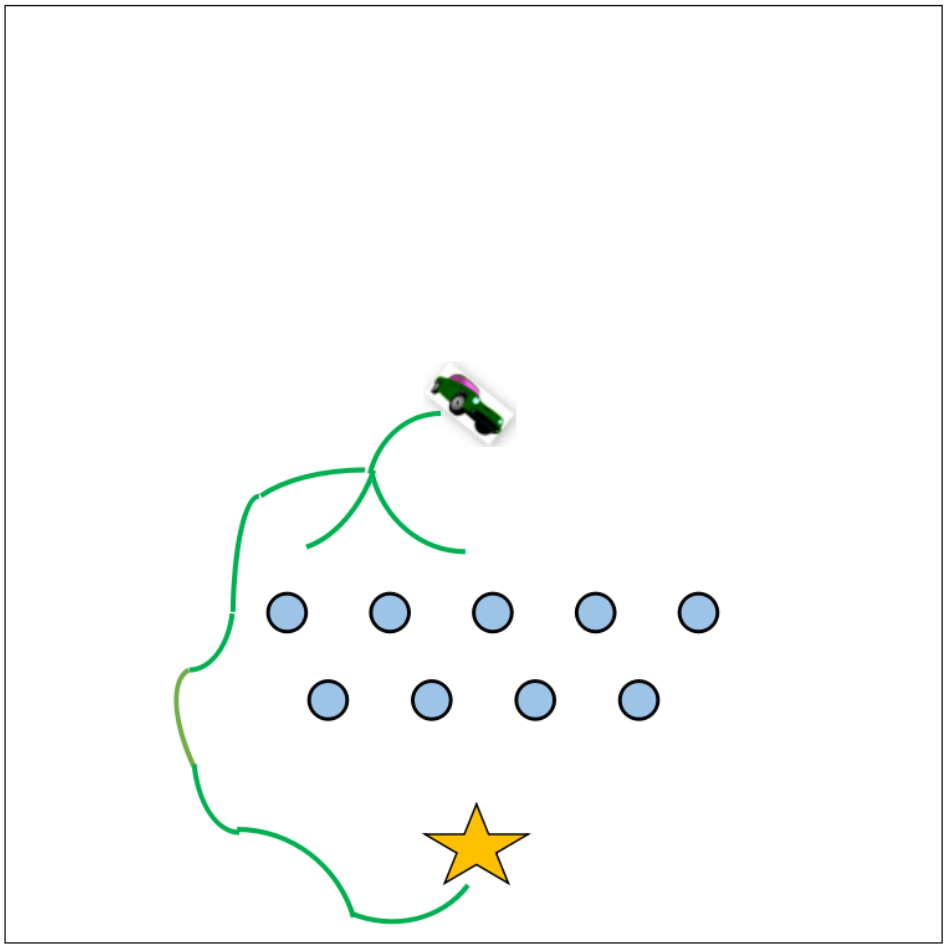
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



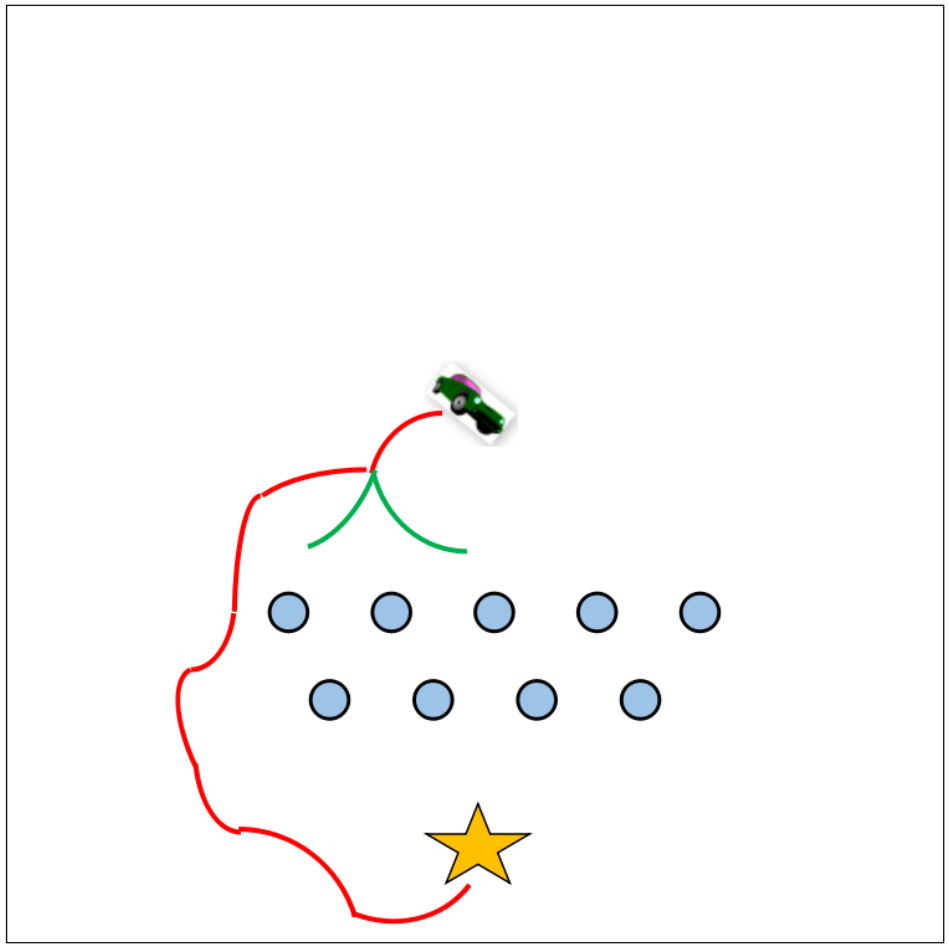
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



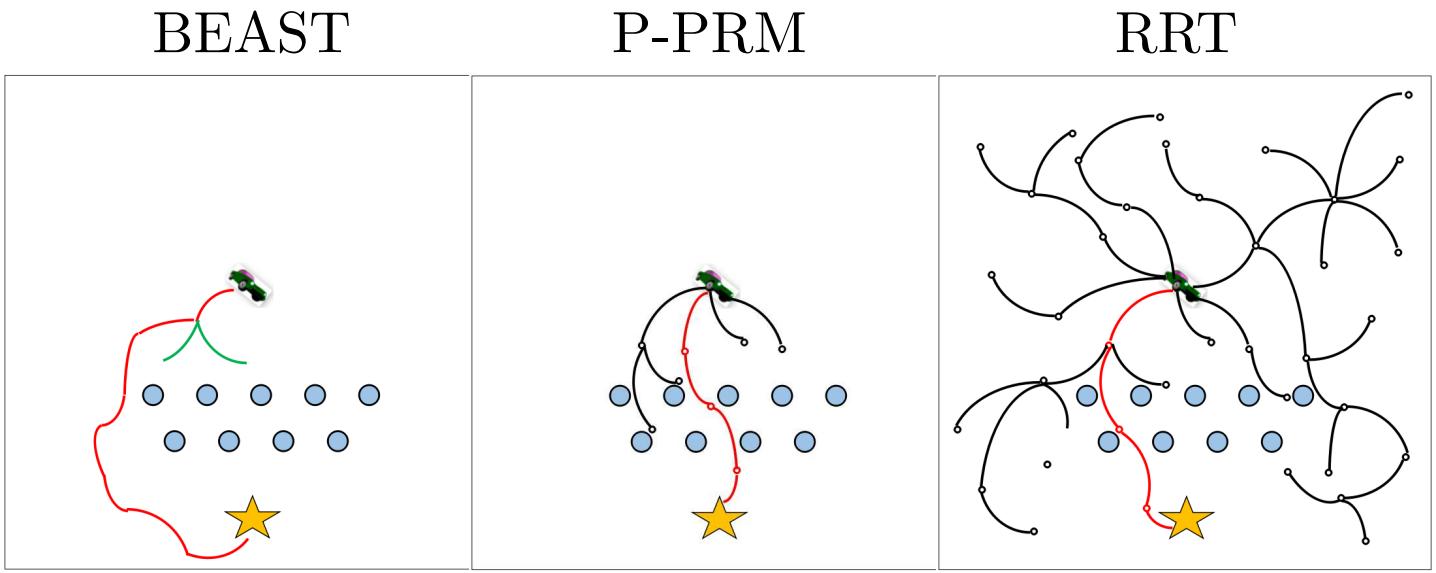
Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



Estimate effort → Guide motion tree growth toward easy way

Effort-guided Planning: BEAST



Beast find solution faster than P-PRM and RRT

Experiments

Environments and Set up

- Open Motion Planning Library (OMPL)
  ompl.kavrakilab.org
  RRT, KPIECE
  Dynamic Car, Blimp, Quadrotor
- We implemented
  P-PRM
  Hovercraft
- 5 start-goal pairs
- 50 random seeds

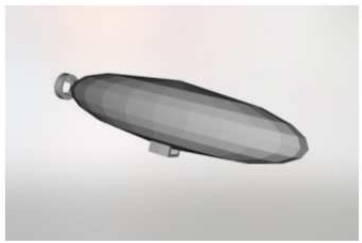
(a) car and hovercraft

(b) open area

(c) 3 ladder

(d) single wall

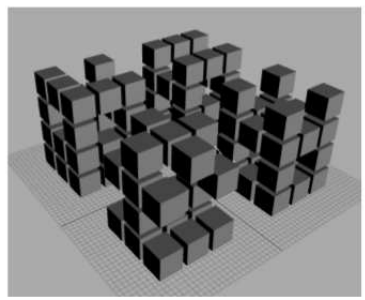
(e) 2D forest



(f) blimp

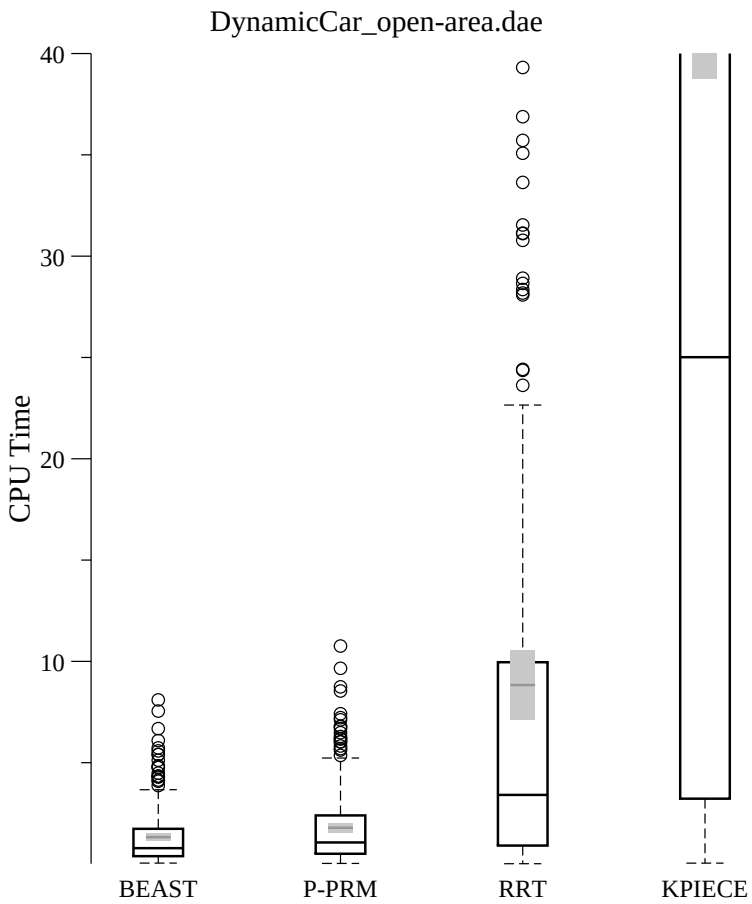
(g) quadrotor

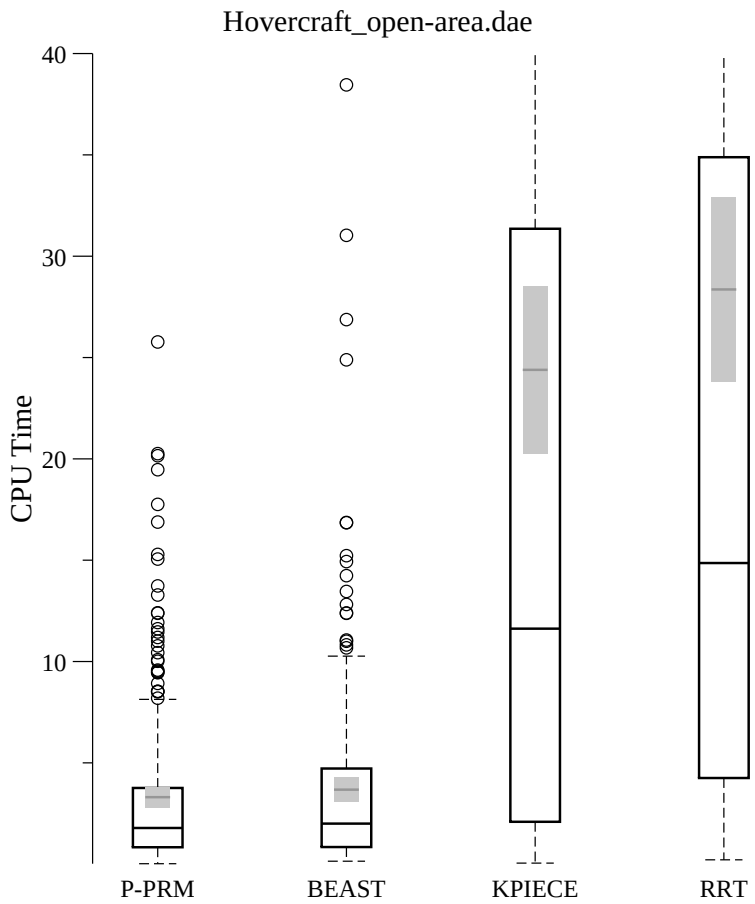
(h) 3D forest

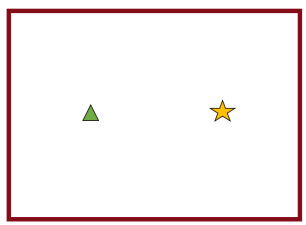


(i) fifthelement

Results: Open Area with Dynamic car & Hovercraft

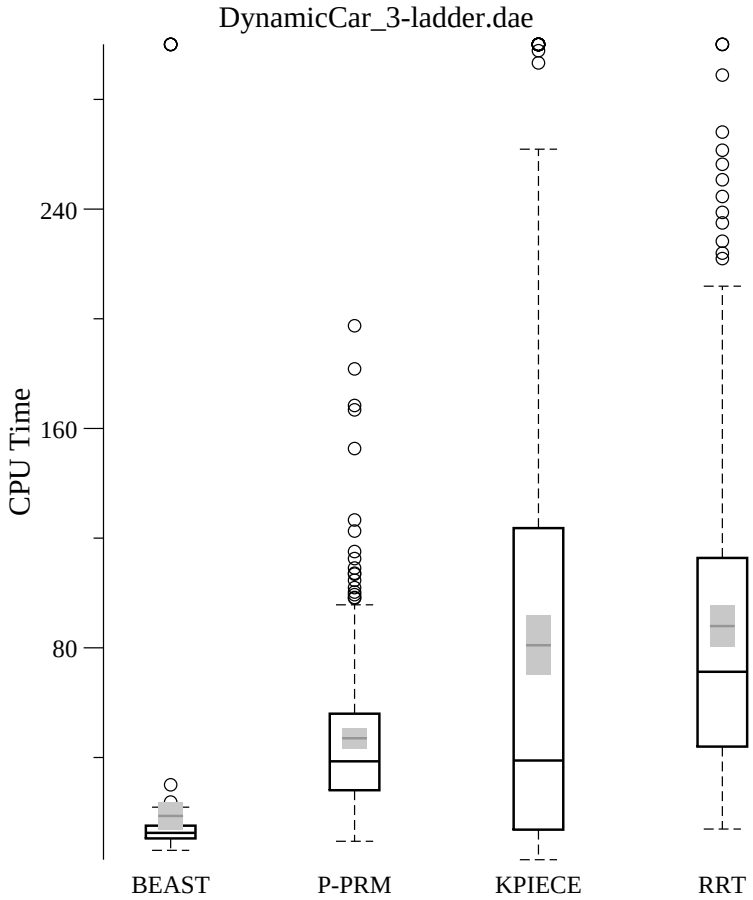


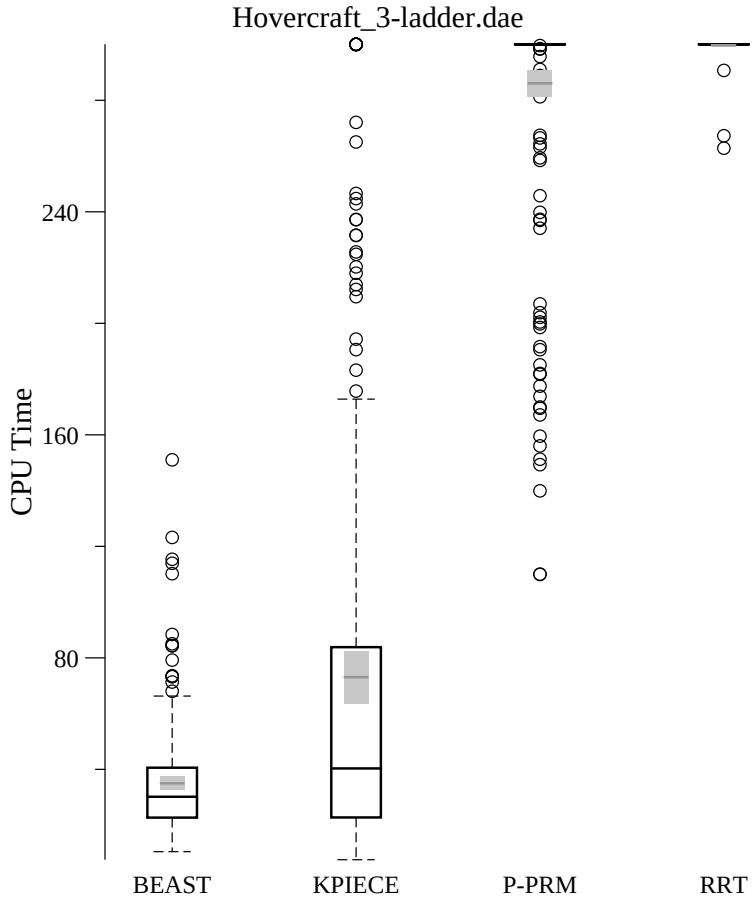


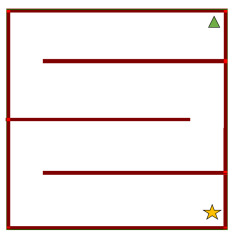


BEAST's overhead seems modest

Results: 3-ladder with Dynamic car & Hovercraft

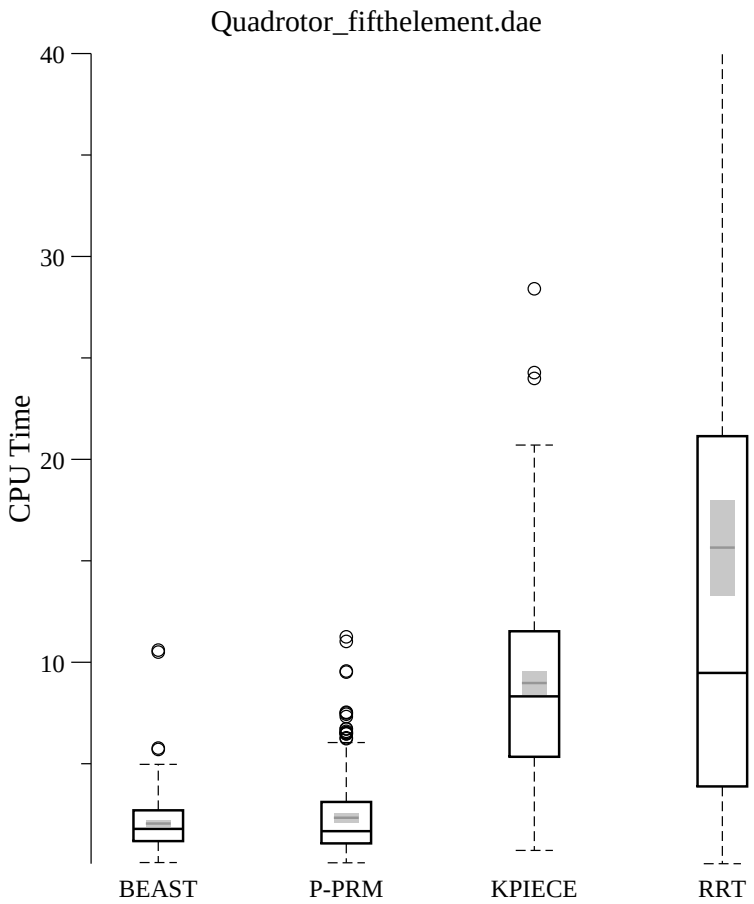


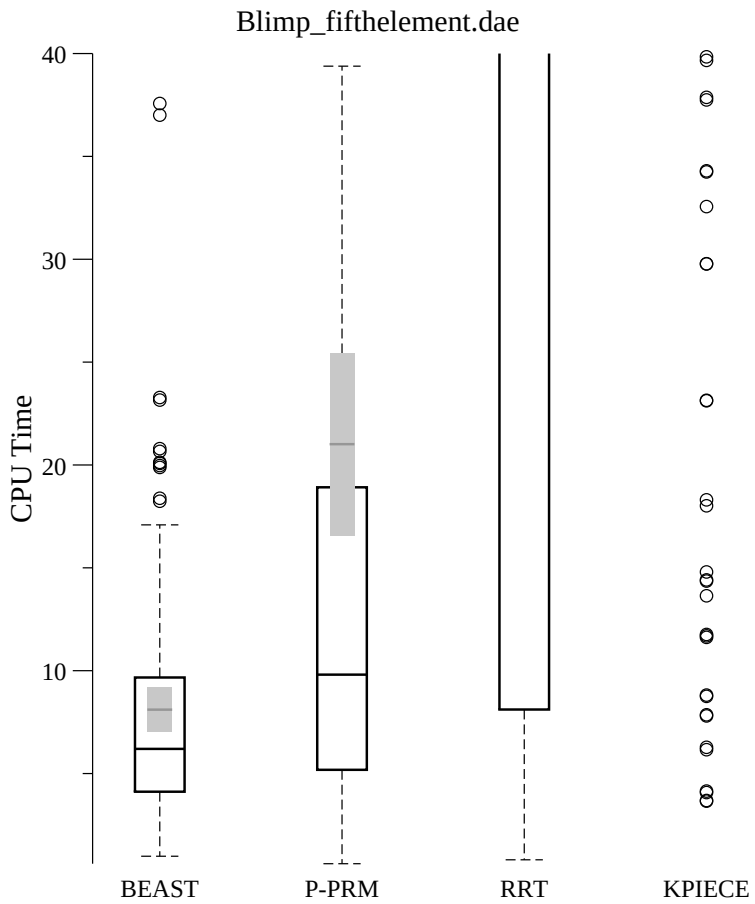


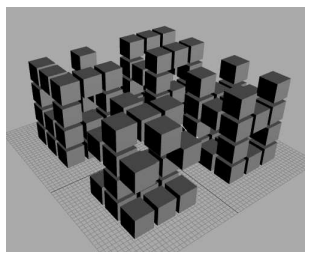


BEAST find complex path quickly

Results: Fifth-element with Quadrotor & Blimp

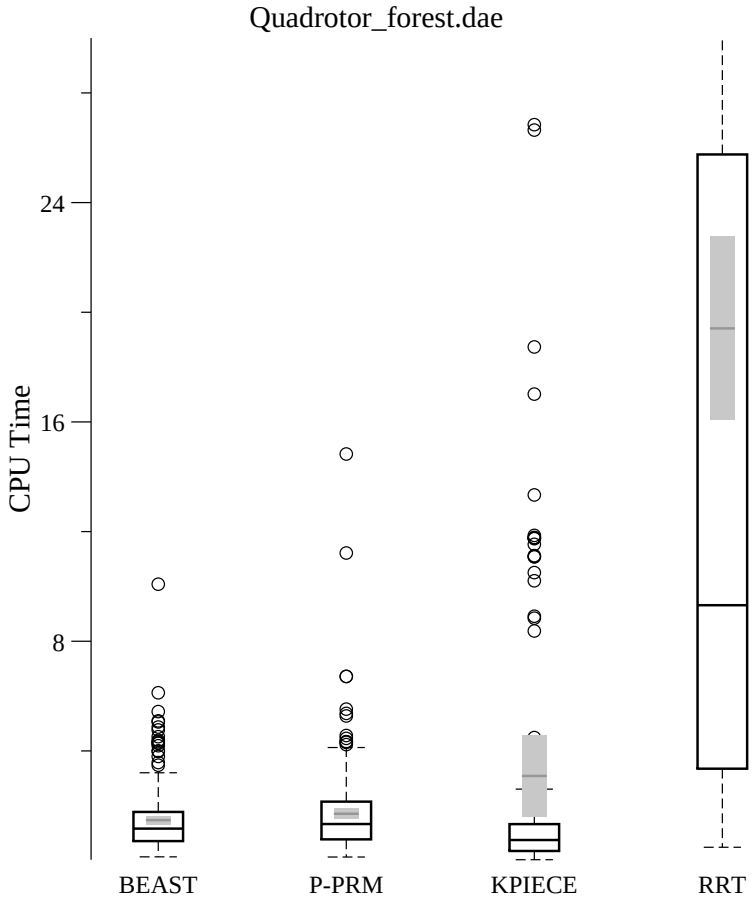


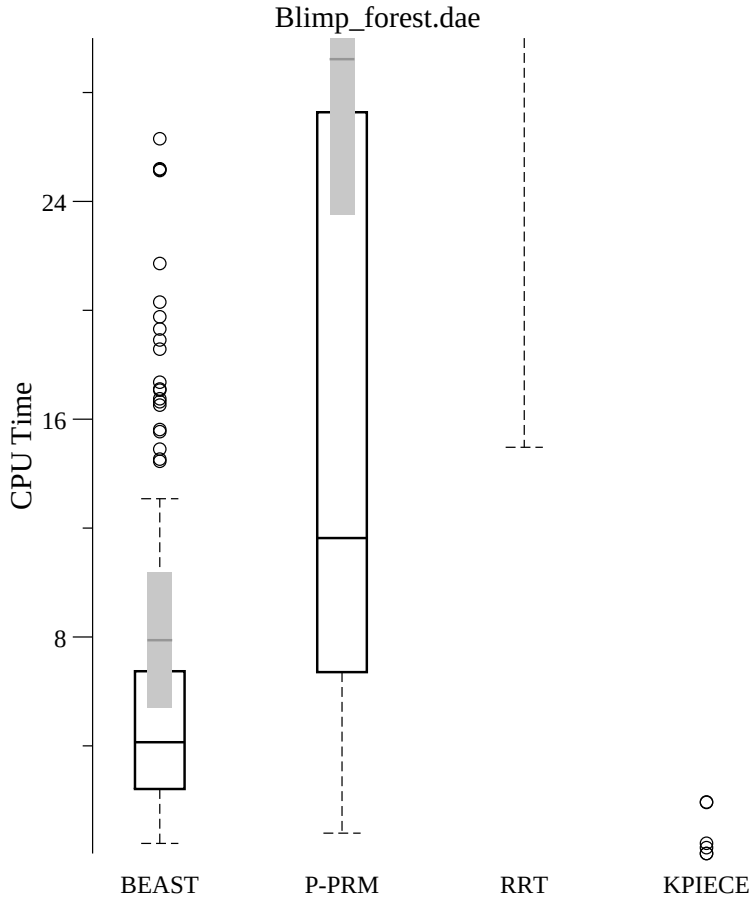


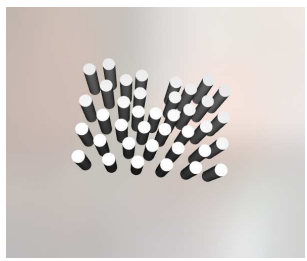


BEAST also performs well in 6-dof and 4-dof

Results: 3D Forest with Quadrotor & Blimp







BEAST deals well with clutter.

Results

Does fast planning yield high-cost plans?

Goal achievement time = planning time + trajectory time

(As factor of BEAST, 95% confidence intervals for the median)

map	vehicle	P-PRM	KPIECE	RRT
open area	car	1.0–1.1	1.8–2.3	1.0–1.2
	hover.	1.0–1.1	1.6–1.9	1.4–1.8
single wall	car	1.0–1.1	1.2–1.4	1.0–1.1
	hover.	∞–∞	1.1–1.3	∞–∞
3 ladder	car	1.0–1.1	1.2–1.3	1.1–1.2
	hover.	∞–∞	1.0–1.1	∞–∞
2D forest	car	0.9–1.1	∞–∞	1.4–1.8
	hover.	0.8–0.9	2.8–∞	∞–∞
3D forest	quad.	0.9–1.0	1.0–1.2	1.1–1.4
	blimp	1.0–1.1	∞–∞	1.9–2.4
fifthelement	quad.	0.8–1.0	0.9–1.0	1.3–1.6
	blimp	0.9–0.9	∞–∞	1.0–1.3

GAT of BEAST is similar to P-PRM and better than KPIECE and RRT

Conclusion

Summary

1. Explicit reasoning about planning effort
2. Find solutions faster than cost-guided planning
3. Continue the transfer of ideas from heuristic graph search to sampling-based motion planning:
   - Abstraction-based heuristics
   - Explicit estimates of effort
   - Online learning for metareasoning

Questions?

Back-up Slides

Limitation

- If abstract misses important aspects of the problem, BEAST may not provide much speed-up
- If the problem is very simple, the overhead of forming and maintaining the abstraction may not be worth the possible decrease in state propagation and collision checking
- Ignore solution cost

Internal Sampling

Benefit of internal sampling? Add more samples to the destination region so that increase the chance it can further propagate outward.

$$te(e) = ee(e) + \min_{e_2 \in e.out} \frac{e_2.\alpha + e_2.\beta + 1/n}{e_2.\alpha + 1/n} + te(e_2.dest)$$