590DF
Introduction to AI-based Mobile Robotics

Path Planning

The Decision Making Problem

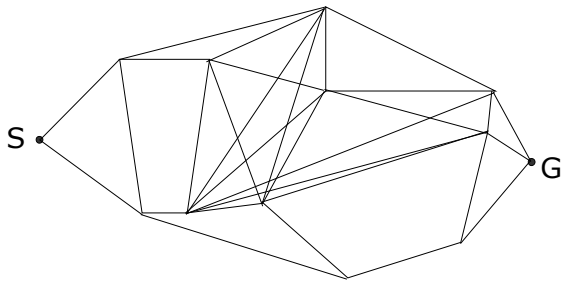
- Central Question:
  What should a robot do next?
- Embraces
  - *control* (short term, tight feedback)
  - *planning* (longer term, looser feedback)
- Probabilistic Paradigm: Considers uncertainty
  - current
  - future

Discretization of Continuous Maps

- Extract a discrete search space / graph from the continuous map.
  - Visibility graphs / roadmaps
  - Voronoi diagrams
  - Probabilistic roadmaps
  - Discrete grids

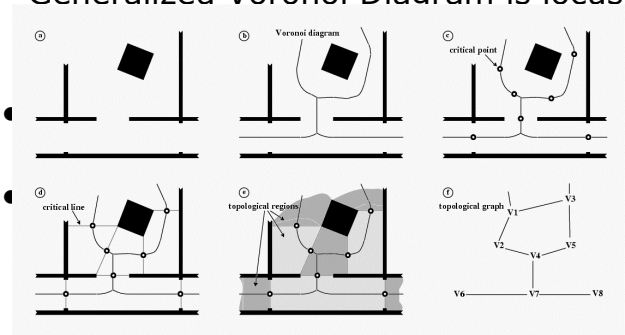
Visibility Graph / Roadmap

- Redcuce path planning to graph search

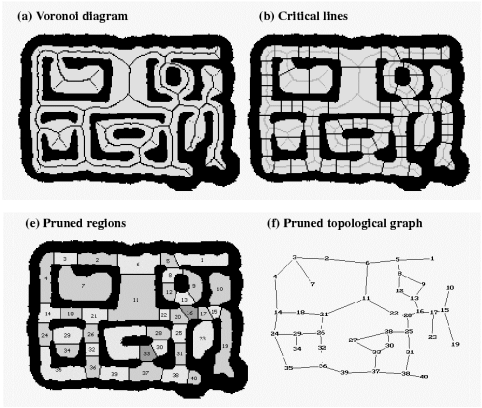


Voronoi Diagrams

- Generalized Voronoi Diagram is locus

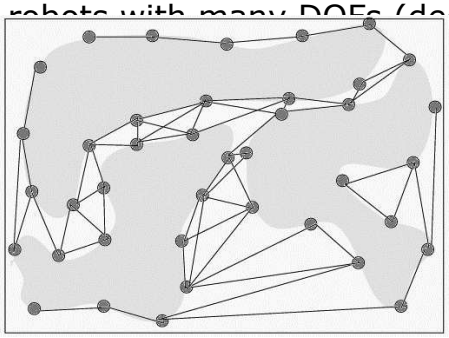


Voronoi Example



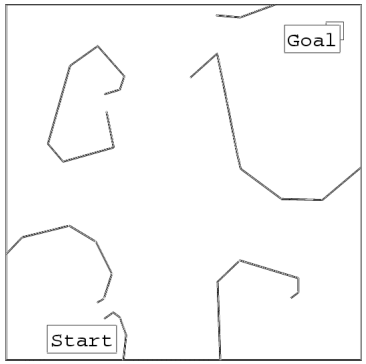
Probabilistic Roadmap

- For robots with many DOFs (degrees of f
- Sar etain tho
- Two h bet local, det



[Amato and Wu, 96]

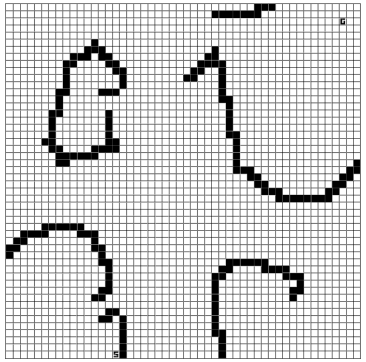
Continuous Environments



From: A Moore & C.G. Atkeson "The Parti-Game Algorithm for Variable Resolution Reinforcement Learning in Continuous State spaces," Machine Learning 1995

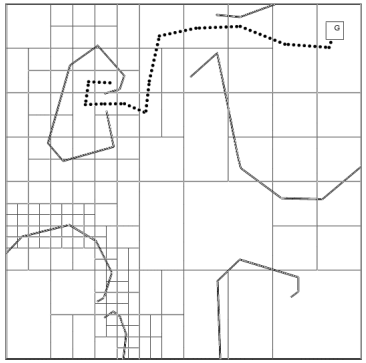
Approximate Cell Decomposition

[Latombe 91]



Parti-Game

[Moore 96]

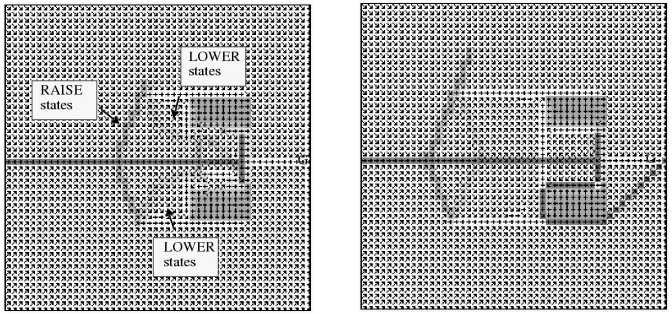


D* Path Planning

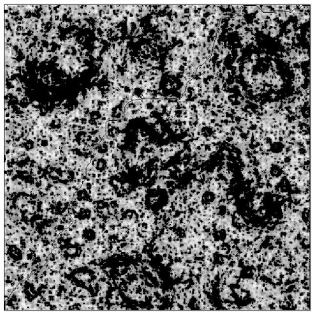
Stentz 94

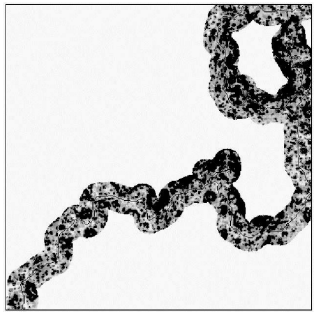
- Similar to A* but designed for efficient replanning
- Maintains an Open list consisting of Raise and Lower states.
- Raise states transfer information about increased cost.
- Lower states propagate information about decreased cost.
- Whenever robot detects new obstacle, a Raise wave front is sent backwards along backpointers.
- Whenever this wave front reaches alternative route, it generates a Lower wave front back to the robot.
- Optimal w.r.t limited map information.

D* Example



D* Example

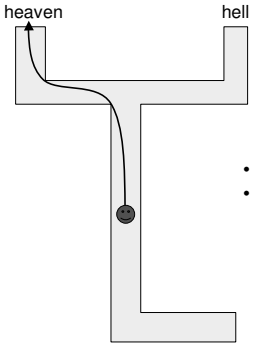




Types of Planning Problems

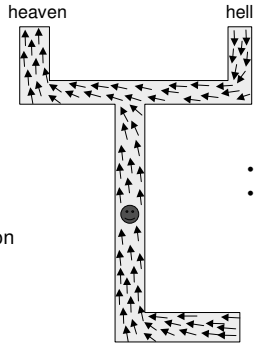
	State	Action Model
Classical Planning	observable	Deterministic, accurate
MDP, universal plans	observable	stochastic
POMDPs	partially observable	stochastic

Classical Situation



- World deterministic
- State observable

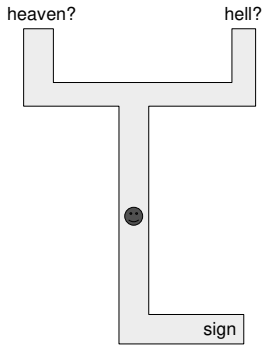
MDP-Style Planning



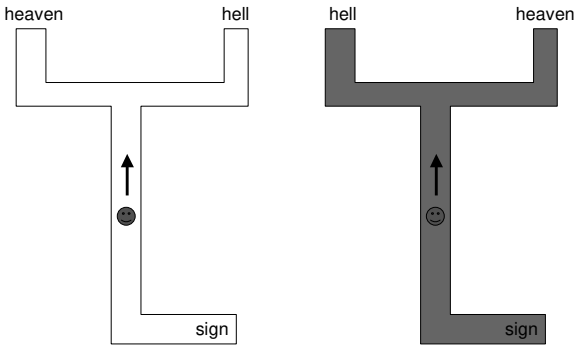
- Policy
- Universal Plan
- Navigation function

- World stochastic
- State observable

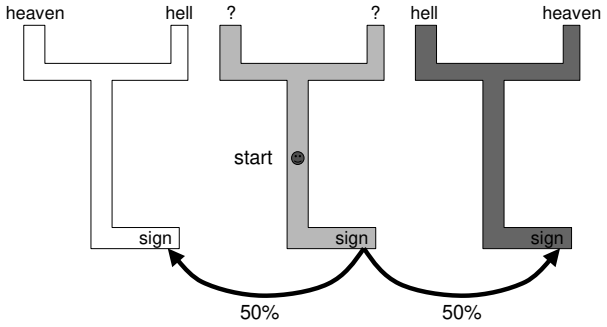
Stochastic, Partially Observable



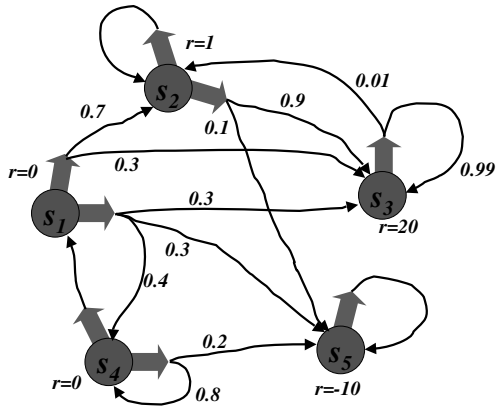
Stochastic, Partially Observable



Stochastic, Partially Observable



Markov Decision Process (MDP)



Markov Decision Process (MDP)

- **Given:**
- Finite set of states S
- Finite set of actions A
- Transition probabilities $p(s'|s,a)$
- Reward function $r(s)$

- **Wanted:**
- Policy $\pi(s)$ that maximizes the future expected reward

Value Iteration

- Value function of policy π

$$V_\pi(s) = E\left[\sum_{i=0}^{\infty} \gamma^i \, r(s_i) \,|\, s_0 = s, a_i = \pi(s_i)\right]$$

- Bellman equation for optimal value function

$$V(s) = r(s) + \max_a \gamma \sum_{s'} p(s'|s,a) V(s')$$

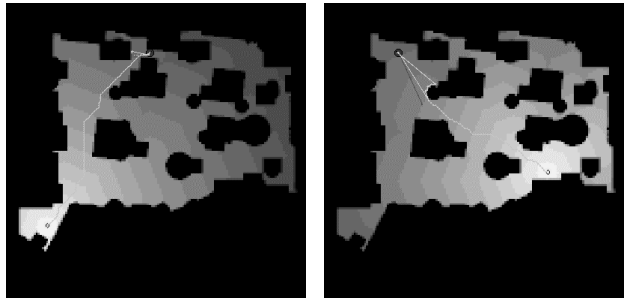
- Value iteration: recursively estimating value function

$$V(s) \leftarrow r(s) + \max_a \gamma \sum_{s'} p(s'|s,a) \, V(s')$$

- Greedy policy: $\pi(s) = \operatorname{argmax}_a \gamma \sum_{s'} p(s'|s,a) \; V(s')$

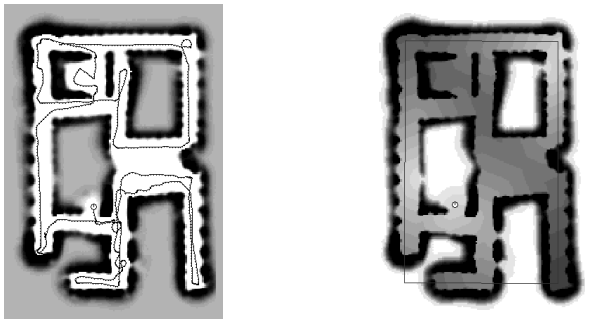
Value Iteration for Motion Planning

(assumes knowledge of robot's location)

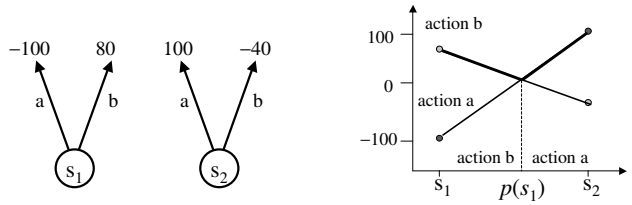


Exploration

- Every unknown location is a target point.



Idea of POMDPs



⇒ Value function (finite horizon): Piecewise linear, convex

Value Iteration in POMDPs

Substitute b for s

- Value function of policy π

$$V_\pi(b) = E\left[\sum_{\tau=0}^{\infty} \gamma^\tau \, r(b_\tau) \,|\, b_0 = b, a_i = \pi(b_i)\right]$$

- Bellman equation for optimal value function

$$V(b) = r(b) + \max_a \gamma \int_{b'} p(b'|b,a)\, V(b')\, db'$$

- Value iteration: recursively estimating value function

$$V(b) \leftarrow r(b) + \max_a \gamma \int_{b'} p(b'|b,a) V(b')\, db'$$

- Greedy policy: $\pi(b) = \operatorname{argmax}_a \gamma \int_{b'} p(b'|b,a) V(b')\, db'$

Augmented MDPs: Coastal Navigation

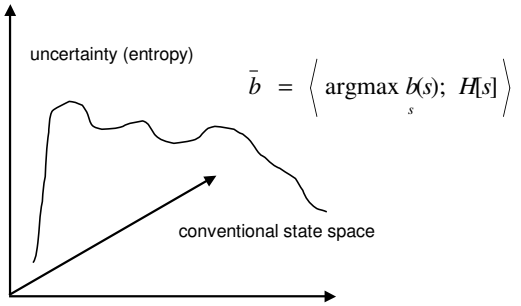
Trade-off between

1. future expected uncertainty about the position of the robot
2. distance to the target location

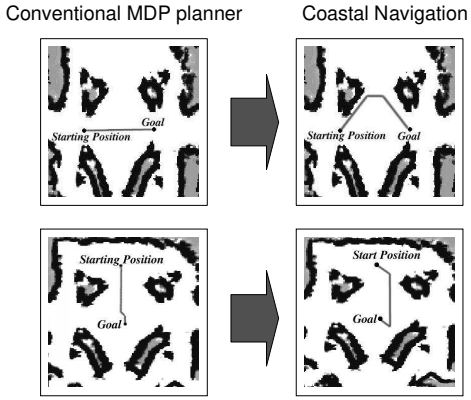
Uncertainty of a belief state is measured by entropy:

$$H(X) = -\sum_X bel(x) \log bel(x)$$

Augmented MDPs:

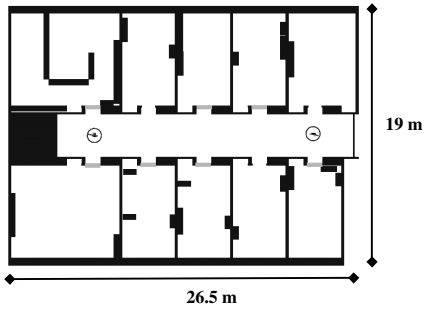


Paths Generated with Coastal Navigation

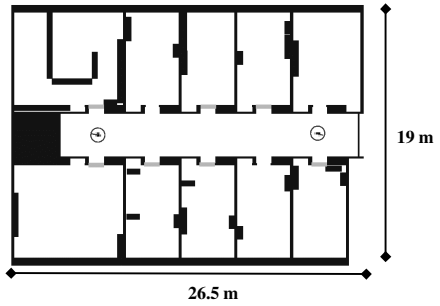


Approximation of POMDPs: Active Localization

Localization so far: **passive** integration of sensor information



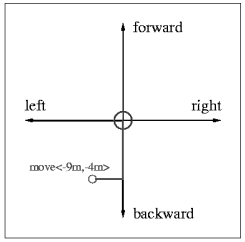
Active Localization: Idea

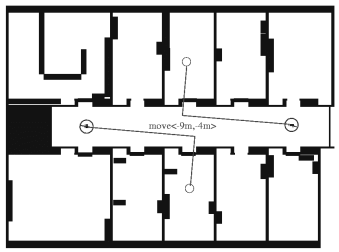


Efficient, autonomous localization by **active** disambiguation

Actions

- Target point **relative** to robot
- Two-dimensional search space
- Choose action based on **utility** and **cost**

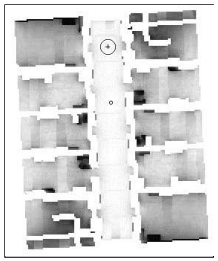




Utilities

- Given by change in uncertainty
- Uncertainty measured by entropy

$$H(X) = -\sum_x Bel(x) \log Bel(x)$$



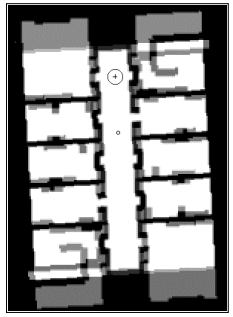
$$U(a) = H(X) - E_a[H(X)]$$

$$= H(X) + \sum_{z,a} p(z \mid x)\, Bel(x \mid a) \log \frac{p(z \mid x)\, Bel(x \mid a)}{p(z \mid a)}$$

Costs: Occupancy Probabilities

- Costs are based on occupancy probabilities

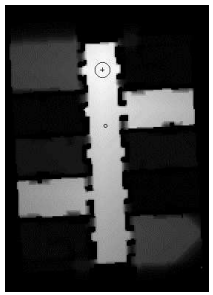
$$p_{occ}(a) = \sum_x Bel(x)\, p_{occ}(f_a(x))$$



Costs: Optimal Path

- Given by cost-optimal path to the target
- Cost-optimal path determined through value iteration

$$C(a) \leftarrow p_{occ}(a) + \min_b [C(b)]$$

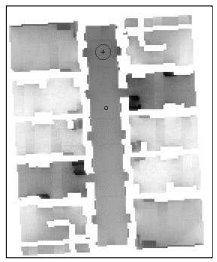


Action Selection

- Choose action based on expected utility and costs

$$a^* = \arg\max_a (U(a) - C(a))$$

- Execution:
  - cost-optimal path
  - reactive collision avoidance

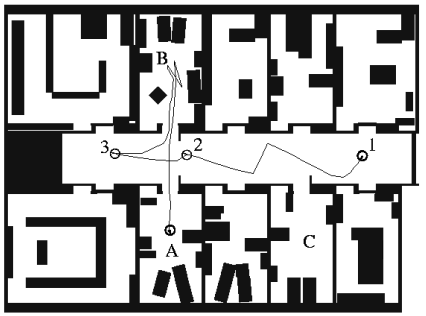


Efficient Implementation

- **Complexity**: $O(|X| \times |A|^T \times |Z|^N)$
  here: $|X| > 10^6$, $|A| = 8$, $|Z| = 64$, $N = 24$

- Local maxima: $|X| = 10$
- Treat sensors independently: $N = 1$
- Only next action: $T = 1$
- Targets are arbitrarily far away: $|A| = |X|$

Experimental Results



- Random navigation failed in 9 out of 10 test runs
- Active localization succeeded in all 20 test runs

Robot Planning Frameworks

	Classical AI/robot planning	Value Iteration in MDPs	Parti-Game	POMDP	Augmented MDP
State/actions	discrete & continuous	discrete	Continuous	discrete	discrete
State	observable	observable	observable	partially observable	partially observable
Environment	deterministic	stochastic	stochastic	stochastic	stochastic
Plans	Sequences of actions	policy	policy	policy	policy
Completeness	Yes	Yes	Yes	Yes	No
Optimality	Rarely	Yes	No	Yes	No
State space size	Huge, often continuous, 6 dimensions	millions	n/a	dozens	thousands
Computational Complexity	varies	quadratic	n/a	exponential	polynomial

Decision Making: Lessons Learned

- Different sources of uncertainty
  - Environment unpredictable
  - Sensor limitations
  - Models inaccurate
- Two implications
  - Need policy instead of simple (open-loop) plan
  - Policy must be conditioned on belief state
- Approaches
  - MDP: Only works with perfect sensors, models
  - POMDPs: general framework, but scaling limitations
  - Augmented MDPs: lower computation, but approximate