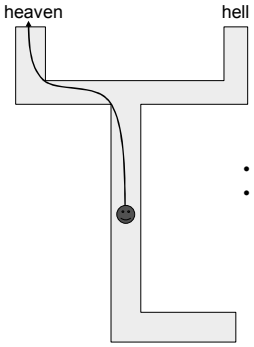
CSE-571
AI-based Mobile Robotics

Path Planning

Types of Planning Problems

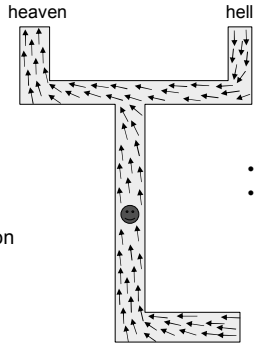
	State	Action Model
Classical Planning	observable	Deterministic, accurate
MDP, universal plans	observable	stochastic
POMDPs	partially observable	stochastic

Classical Situation



- World deterministic
- State observable

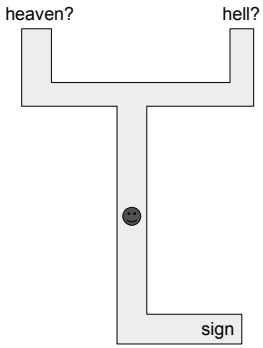
MDP Style Planning



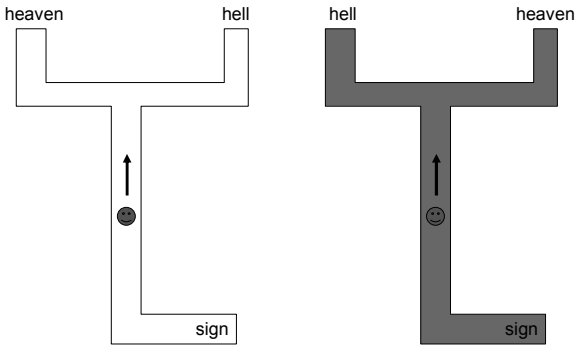
- Policy
- Universal Plan
- Navigation function

- World stochastic
- State observable

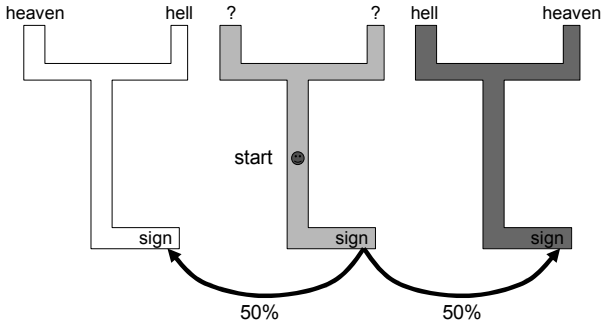
Stochastic, Partially Observable



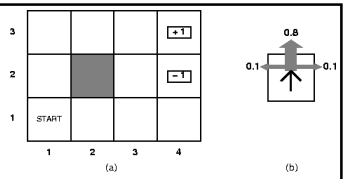
Stochastic, Partially Observable



Stochastic, Partially Observable

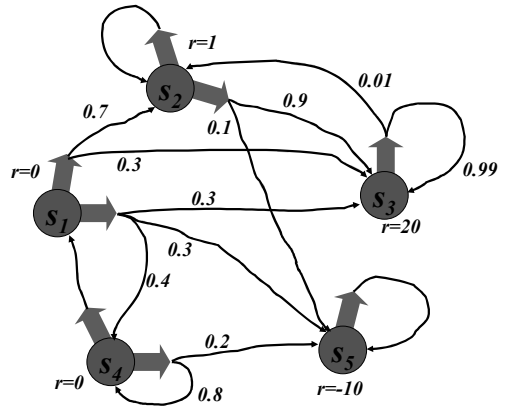


Markov Decision Process Example



- Robot must reach state (4, 3) (reward of +1) and avoid state (4, 1) (punishment of –1). Actions are North, South, West, East.
- **Deterministic version:** All actions always lead to the next square in the selected direction, except that moving into a wall results in no change in position.
- **Stochastic version:** Each action achieves the intended effect with probability 0.8, but the rest of the time, the agent moves at right angles to the intended direction.

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

- Optimal policy (discounted reward)

$$\pi^* = \underset{\pi}{\operatorname{argmax}} \; E\left[\sum_{t=0}^{\infty} \gamma^t \, R(s_t) | \pi\right]$$

- Value function of state s under policy π

$$V_\pi(s) = E\left[\sum_{t=0}^{\infty} \gamma^t \, R(s_t) | s_0 = s, \pi\right]$$

- Bellman equation for optimal value function

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

Value Iteration

- Value iteration: recursively estimating value function

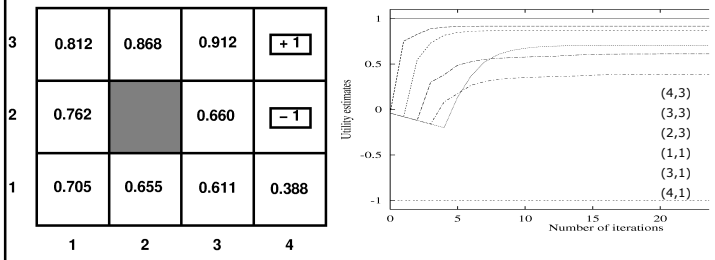
$$V(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} p(s'|s,a) V(s')$$

- MEU greedy policy:

$$\pi(s) = \underset{a}{\operatorname{argmax}} \; \gamma \sum_{s'} p(s'|s,a) \; V(s')$$

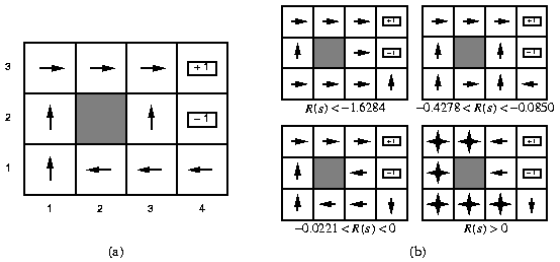
MDP Example

- Value function for R(s) = -0.04



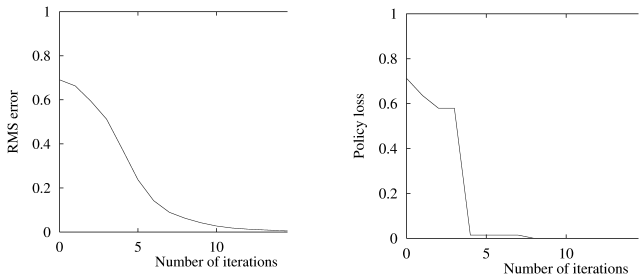
MDP Example

- Optimal policies



MDP Example

- Utility error and policy error

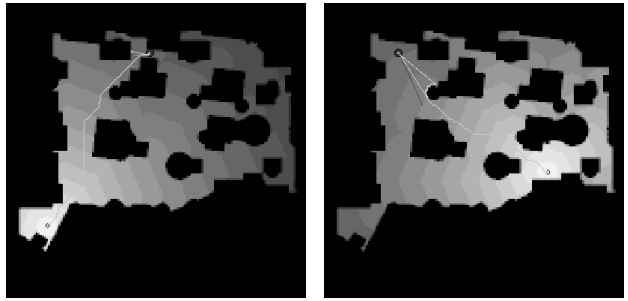


Policy Iteration

- Iterate between
- Policy evaluation:
  Given policy π_i, calculate utility $V_i = V_{\pi i}$ of each state if π_i were to be executed
- Policy improvement:
  Calculate a new MEU policy π_{i+1}

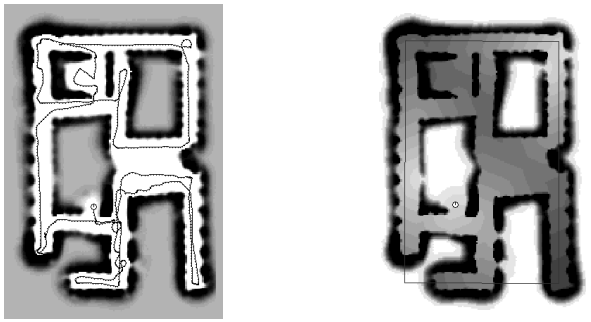
Value Iteration for Motion Planning

(assumes knowledge of robot's location)

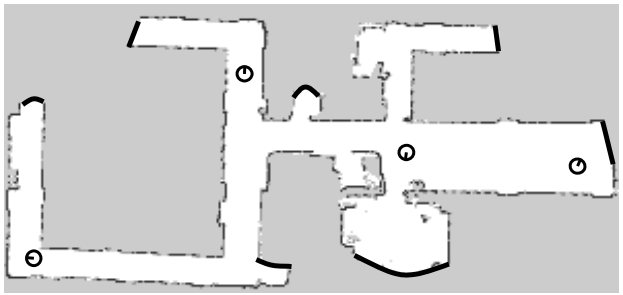


Frontier-based Exploration

- Every unknown location is a target point.



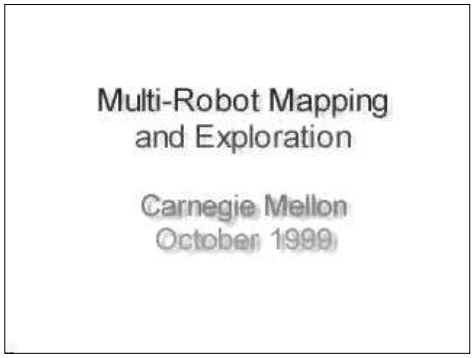
Multi-Robot Exploration



$$C(\theta) = \sum_{(i,j)\in\theta} \text{dist}(i,j) \qquad U(\theta) = \sum_{(i,j)\in\theta} \text{explore}(i,j)$$

$$\theta^* = \arg\max_{\theta} \left(U(\theta) - C(\theta)\right)$$

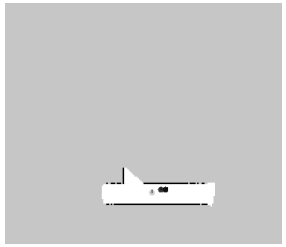
Coordinated Multi-Robot Exploration



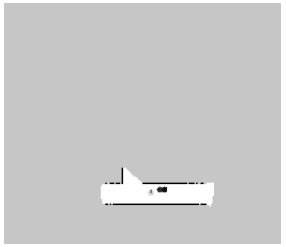
Thrun et al.:
DARPA TMR,
San Antonio 10/99

Coordinated Multi-Robot Exploration

Uncoordinated:



Coordinated:

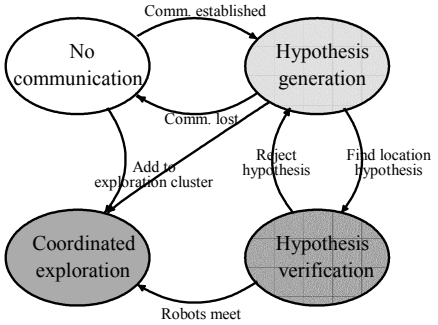


[Burgard et al. '00]

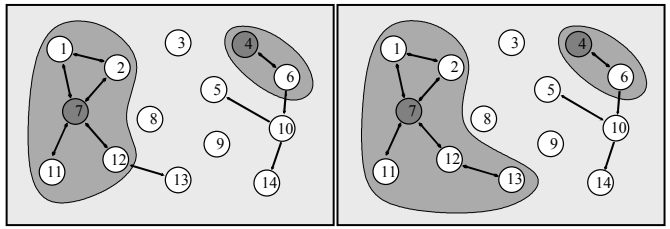
Distributed Exploration

- **Problems**
  - Limited communication
  - No centralized control / coordination
  - Robots don't know their start locations
- **Approach**
  - Initially robots explore individually
  - Whenever robots can communicate they …
    - … try to combine their maps
    - … trade off exploration and map merging
    - … coordinate exploration

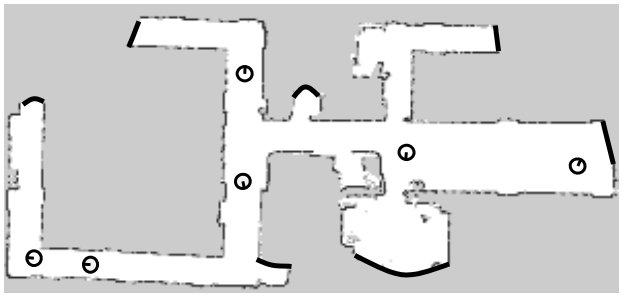
Interactions Between Robots



Dynamic Team Coordination



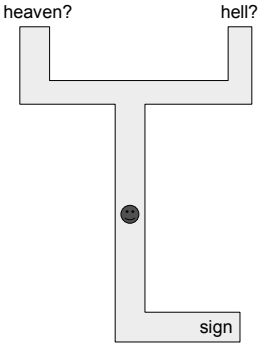
Coordination for Map Merging



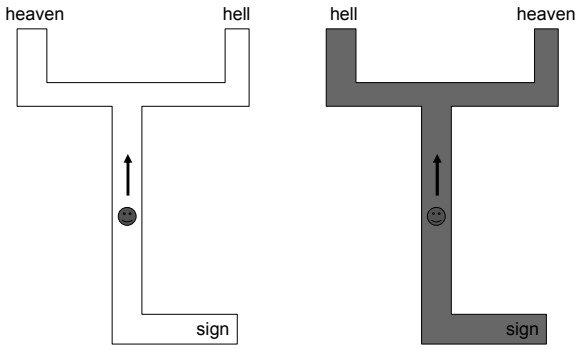
$$C(\theta) = \sum_{(i,j)\in\theta} \begin{cases} \text{dist}(i,j) & \text{if } j \text{ is frontier} \\ \text{dist}(i,j) + \text{meet}(i,j) & \text{if } j \text{ is hypothesis} \end{cases} \qquad U(\theta) = \sum_{(i,j)\in\theta} \begin{cases} \text{explore}(i,j) & \text{if } j \text{ is frontier} \\ p(j)\,\text{merge}(i,j) & \text{if } j \text{ is hypothesis} \end{cases}$$

$$\theta^* = \arg\max_{\theta} \left(U(\theta) - C(\theta) \right)$$

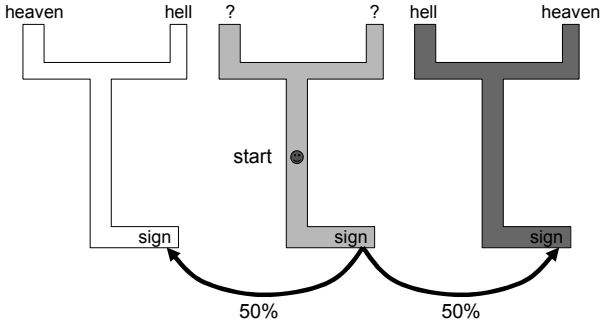
Stochastic, Partially Observable



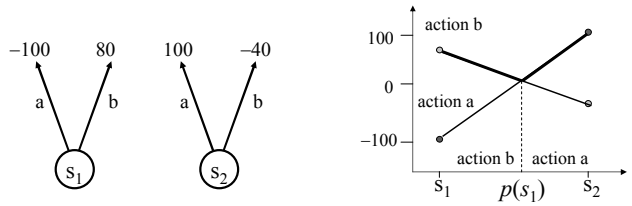
Stochastic, Partially Observable



Stochastic, Partially Observable



Idea of POMDPs



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

Value Iteration in POMDPs

Substitute b for s

- Value function of policy π

$$V_\pi(b) = E\left[\sum_{t=0}^{\infty} \gamma^t R(b_t) \mid b_0 = b, \pi\right]$$

- Bellman equation for optimal value function

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

- Value iteration: recursively estimating value function

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

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

Augmented MDPs: Coastal Navigation

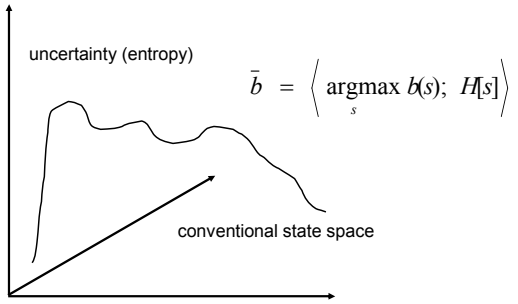
Trade-off between

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

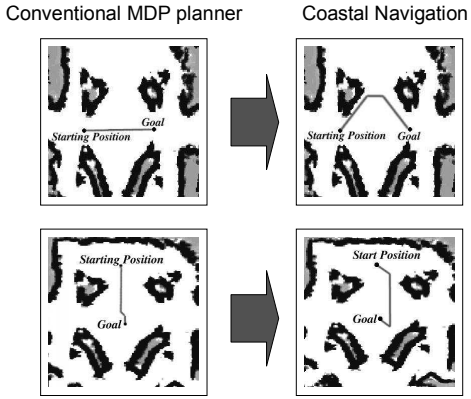
Uncertainty of a belief state is measured by entropy:

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

Augmented MDPs:

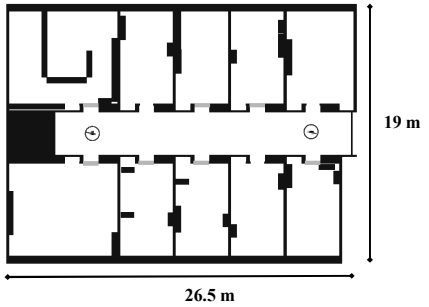


Paths Generated with Coastal Navigation

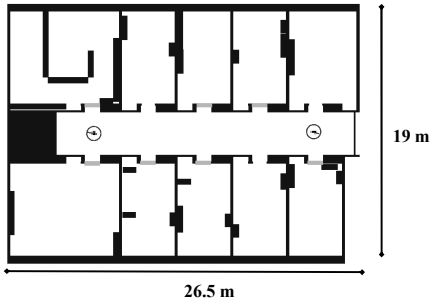


Approximation of POMDPs: Active Localization

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



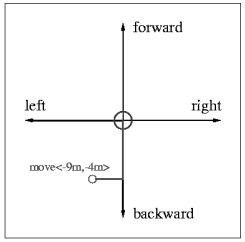
Active Localization: Idea

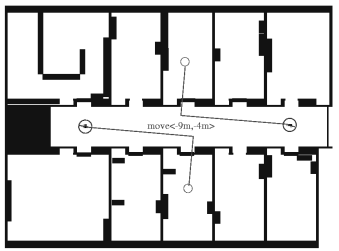


Efficient, autonomous localization by **active** disambiguation

Actions

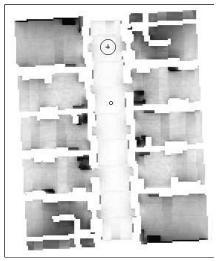
- Target point **relative** to robot
- Two-dimensional search space
- Choose action based on **utility** and **cost**





Utilities

- Given by change in uncertainty
- Uncertainty measured by entropy



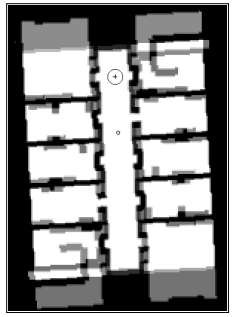
$$H(X) = -\sum_x Bel(x) \log Bel(x)$$

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

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

Costs: Occupancy Probabilities

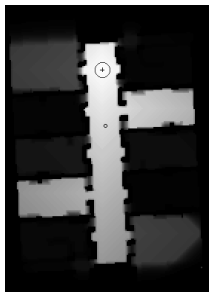
- Costs are based on occupancy probabilities



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

Costs: Optimal Path

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



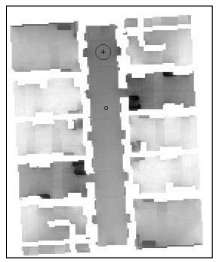
$$C(a) \leftarrow p_{occ}(a) + \min_b [C(b)]$$

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

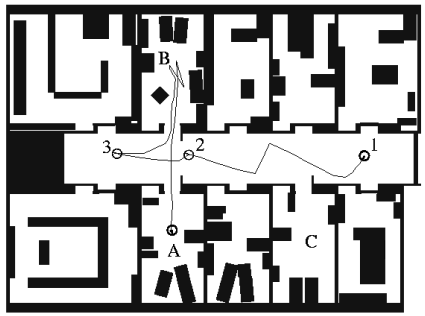


Efficient Implementation

- **Complexity**: $O(|X| \text{ x } |A|^T \text{ x } |Z|^N)$
  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