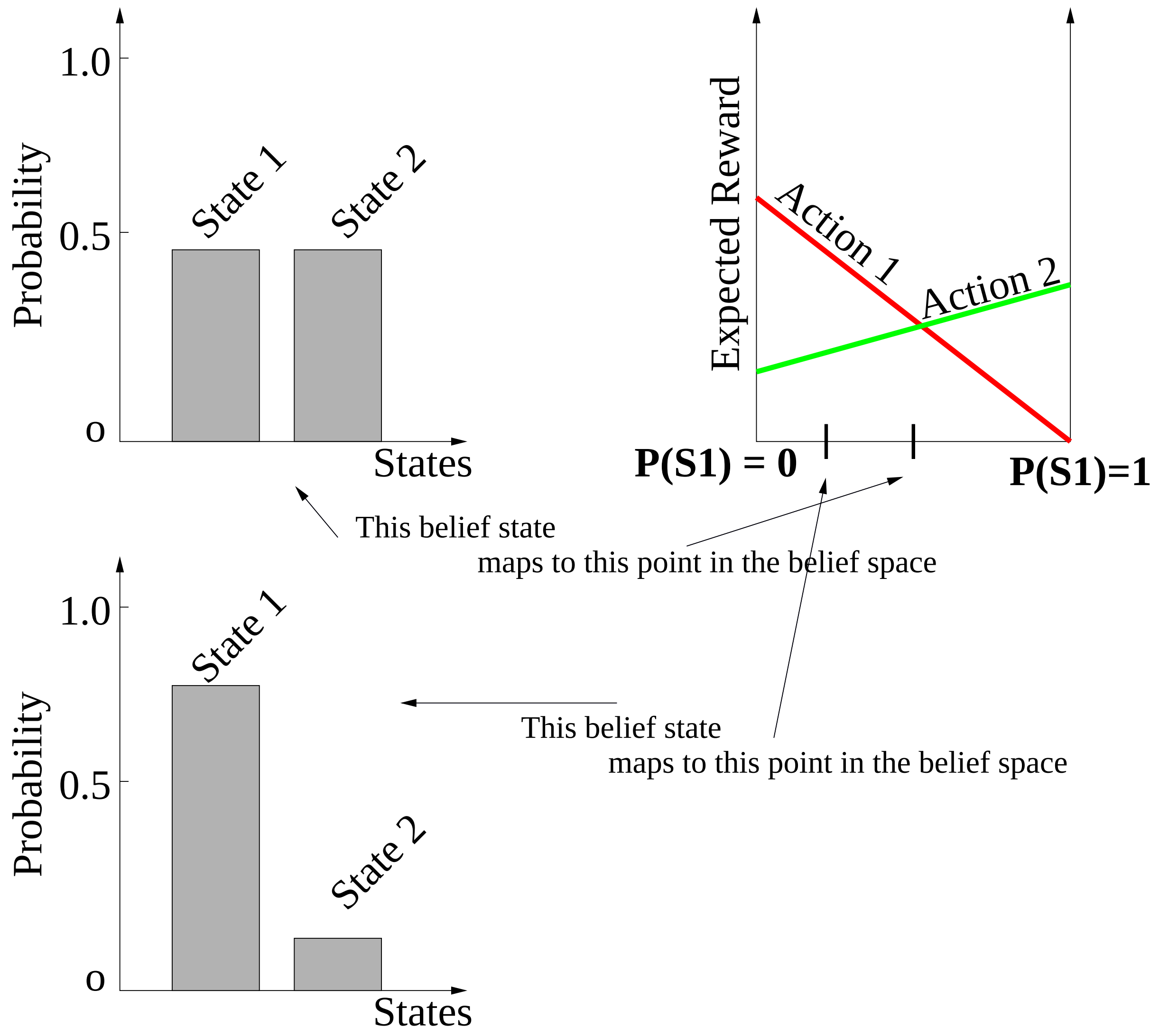
Finding Policies to POMDPs

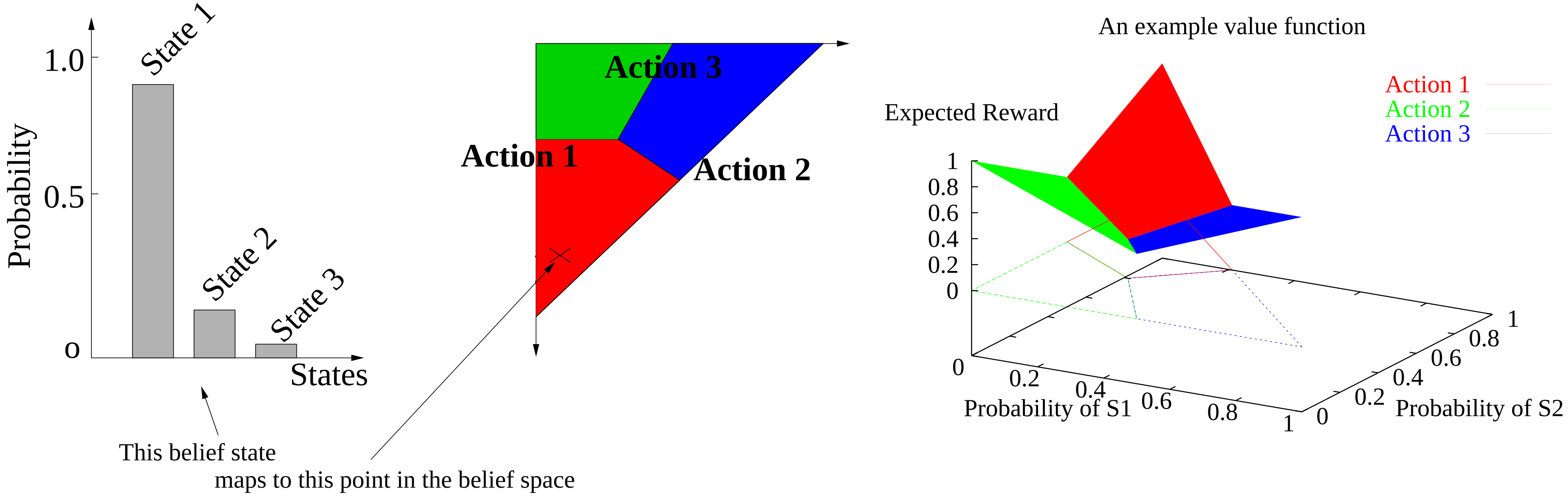
Finding Policies to POMDPs

- Finding an exact policy to an infinite horizon, stochastic, partially observable POMDP takes exponential time, and is not known to be decidable.
- Finding an exact policy to a finite horizon stochastic, partially observable POMDP is PSPACE-hard
- Finding an exact policy to a finite horizon, unobservable POMDP is NP-hard.
- Approximate solutions are required

Example Policy

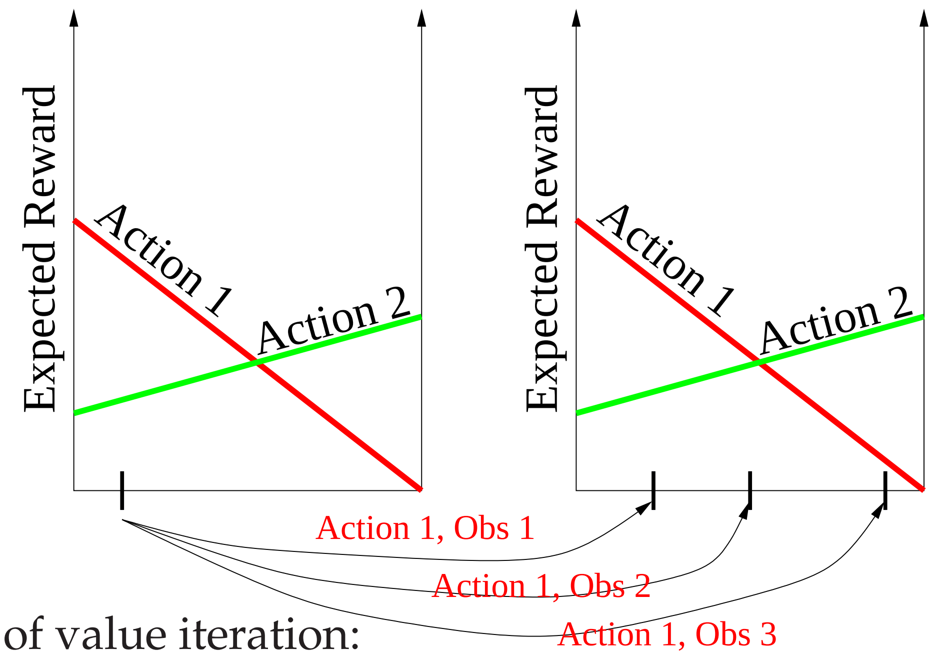


3 State Problem



- ❑ Value function is convex, piece-wise linear
- ❑ Intuitively, reward is highest at edges of belief space
  - ➤ Certainty of acquiring maximum reward is highest at edges
  - ➤ ... because ability to choose action is best at edges there
  - ➤ ... because ability to choose actions is best there
  - ➤ ... because knowledge of true underlying state is highest

Value Iteration



❑ For one step of value iteration:

① For each initial state b:

② Compute final b'_n for each possible observation n

③ Compute weighted value $V(b)$ as:

$$V(b,a) = V(b,a) + \sum_{n=1}^{|O|} p(b'_n) V(b'_n)$$

Approximate Solutions

- Witness algorithm
- Assumptive planner
- Monte Carlo POMDPs
- Greedy heuristic
  - Most Likely State Strategy
  - Voting Strategy
  - Completely Observable after the First Step

Witness

- ❑ Starts with known belief state
- ❑ Constructs initial set of value function vectors
  - ➤ *Each vector represents an action-observation sequence*
- ❑ Searches each vector for belief state where the sequence is no longer dominant.
  - ➤ *With the terminal observation and belief state, we could have done better choosing a different action. This belief state is called a <u>witness</u> point.*
- ❑ Finds arbitrarily close solutions, given enough time
  - ➤ *Problem is of course the search. For every witness, (and there are a lot usually) we need to search a new vector.*

Most Likely State Strategy (Nourbakhsh et al., 1995)

- Maintain a state-set $U \subseteq S$
- Action a generates a new state-set $U' \subseteq S$
- Perception o generates a new state-set $U'' \subseteq U'$.
- Good for fast representation of coarse ambiguities

Assumptive Planner (Nourbakhsh et al., 1996)

- Maintain a state-set $U \subseteq S$
- Action a generates a new state-set $U' \subseteq S$
- Perception o generates a new state-set $U'' \subseteq U'$.
- Makes assumptions about set U by simplifying perceptions
- Generates complete path to goal set $U \subseteq G$ where G is the set of desired goal states

Voting Strategy (Simmons & Koenig, 1995)

- Maintain a state-set U
- Action a generates a new state-set U' (not necessarily subset of U
- Perception o generates a new state-set $U'' \subseteq U'$.
- Good where different states all lead to same action

Completely Observable after the First Step (Chrisman 1992)

- The value of a state is the combination of a single step of value backup, and the Q-value of the predicted posterior states computed from the underlying Markov model. Works in domains where heuristic can push belief state to edges of belief space, where POMDP most behaves like MDP
- Used by a number of heuristic planners: Koenig and Simmons 96, Geffner and Bonet's RTDP-Bel.
- RTDP-Bel works best when number of actions to resolve ambiguities is small.

Problems with Greedy Heuristics

- Information gathering occurs "on the fly" – do not predict when, where and what to sense
- Greedy heuristics work when execution of optimal policy resolves any ambiguities automatically
- Greedy heuristics break when the ambiguity does not go away without an explicit resolution action

  ➤ *The power of the heuristic path planners evaporates if active localisation is ever needed.*

Augmented MDP

- Inspiration from Coastal Navigation

  Found shortest, most reliable path trading off distance against cost of being uncertain. Cost related to information gained from predicted sensor readings at each point in space.

- Central idea – Instead of representing uncertainty as cost, represent uncertainty as explicit part of state space.

- Compress belief state into compact representation, given by

$$p(\mathbf{s}) \equiv < \operatorname*{argmax}_{\mathbf{s}} p(\mathbf{s}); H(p(\mathbf{s})) >$$

  where $H(p(s))$ is entropy of belief state.

- Perform dynamic programming using compressed belief state to generate universal plan

Computing the Next State

- We use the motion and sensing equations from Markov Localisation:

$$p(\mathbf{s}|\,\mathbf{a}) = \int_{\mathbf{s}'} p(\mathbf{s}'|\mathbf{s},\mathbf{a})p(\mathbf{s})d\mathbf{s}$$

$$p(\mathbf{s}|\,\mathbf{o}) = \frac{p(\mathbf{o}|\,\mathbf{s})p(\mathbf{s})}{p(\mathbf{o})}$$

- But at every step of planning, we represent the belief $p(\mathbf{s})$ compactly as belief state s_i:

$$p(\mathbf{s}) = < \operatorname*{argmax}_{\mathbf{s}} p(\mathbf{s}); H(p(\mathbf{s})) > = \mathbf{s}_i$$

- We can then use value iteration

$$J(\mathbf{s}_i) = \max_a [R(\mathbf{s}_i) + C(\mathbf{s},a) + \sum_{j=1}^{N} p(\mathbf{s}_j|\mathbf{s}_i,a) \cdot J(\mathbf{s}_j)]$$

$$\pi(\mathbf{s}_i) = \operatorname*{argmax}_a [R(\mathbf{s}_i) + C(\mathbf{s},a) + \sum_{j=1}^{N} p(\mathbf{s}_j|\mathbf{s}_i,a) \cdot J(\mathbf{s}_j)]$$

Robot Navigation Example

To see the how the state advances as we plan motion and sensing:

① Reconstruct $p(\mathrm{s})$ under the assumption of a unimodal Gaussian with entropy $H(p(\mathbf{s}))$

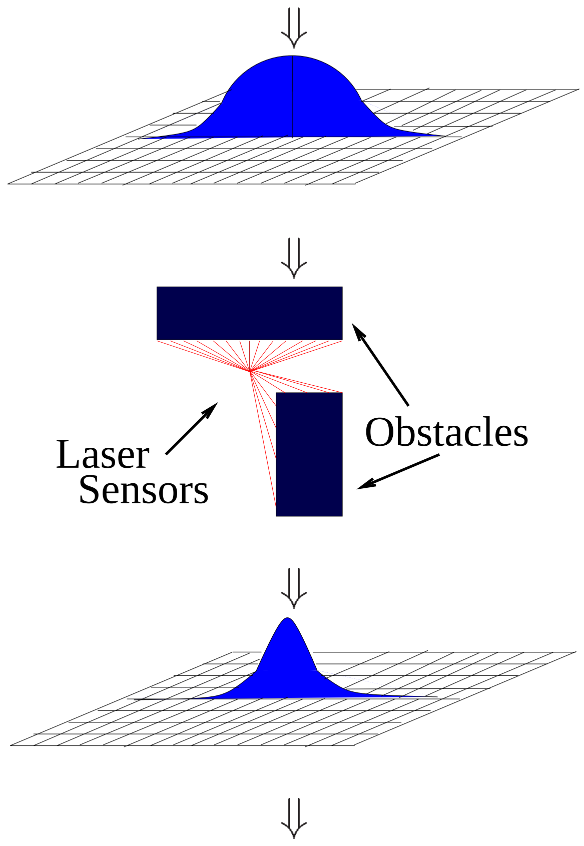
② Apply motion command $\mathbf{a}$ to get $p(\mathbf{s}|\mathbf{a})$

③ Predict sensor measurement $\mathbf{o}$ at position $\mathrm{argmax}_{\mathbf{s}}\, p(\mathbf{s})$

④ Apply prediction $\mathbf{o}$ to get $p(\mathbf{s}|\mathbf{o})$

⑤ Compress final $p(\mathrm{s})$ to low-dimensional statistic

$$< \mathrm{argmax}_{\mathbf{s}}\, p(\mathbf{s}); H(p(\mathbf{s})) >$$



$$< \mathrm{argmax}_{\mathbf{s}}\, p(\mathbf{s}); H(p(\mathbf{s})) >$$

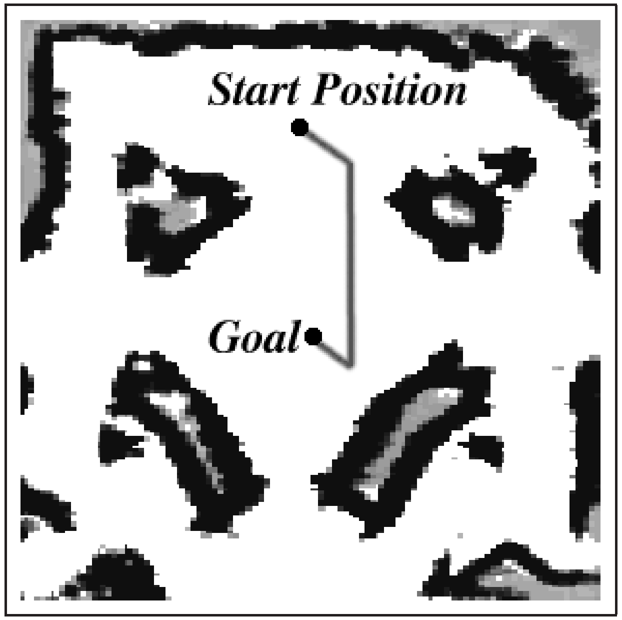
An example trajectory

① The robot localises

② Travels across the open space

③ Relocalises at the other side

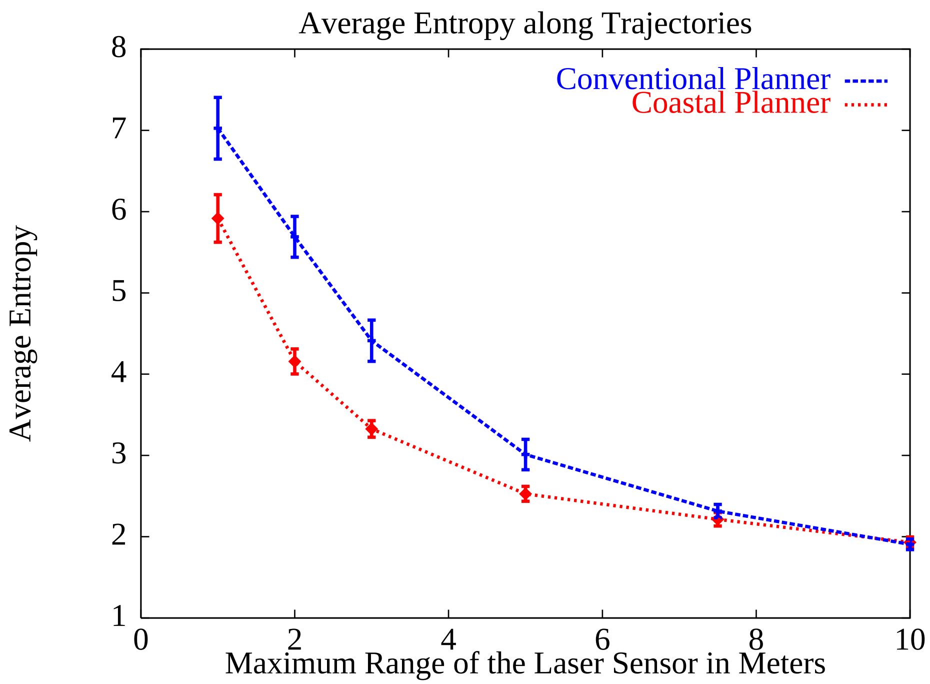
④ Travels to the goal

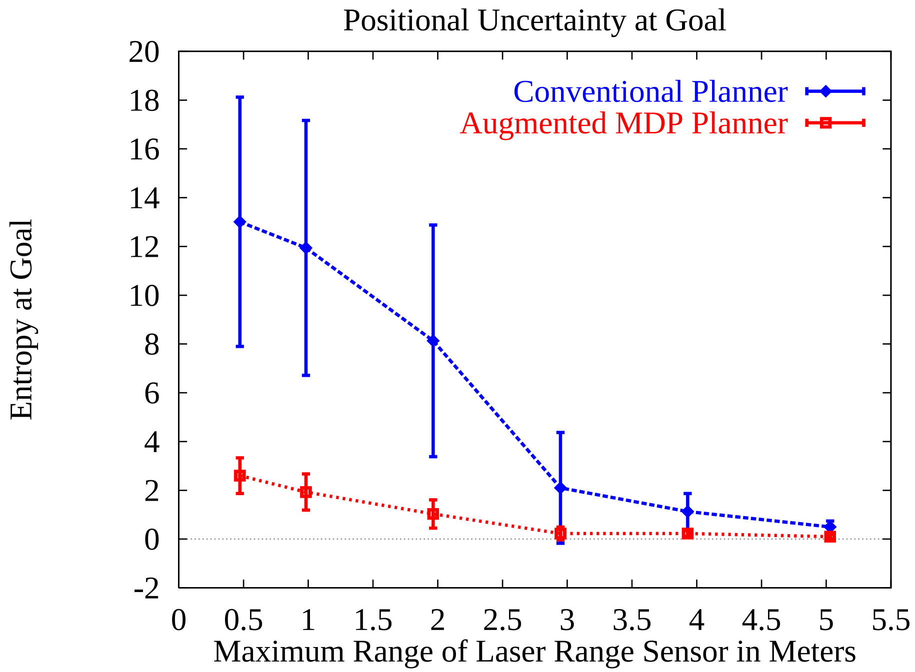


➤ Uncertainty can grow arbitrarily large, and then shrink arbitrarily quickly

➤ Distance travelled to preserve positional certainty governed by tradeoff between distance costs and reward structure. Robot can be made to travel arbitrarily far by adjusting costs and rewards.

Comparison of Coastal to Augmented Planner





Speech Domain

- ❑ Use the Augmented Planner in a different domain – dialog management
- ❑ State of human user is inherently poorly observable
- ❑ Domain shares very few features with robot navigation
  - ➤ Obvious topology missing
  - ➤ Much wider set of actions, with no natural neighbour restrictions
  - ➤ Observation probability model very different, not even static

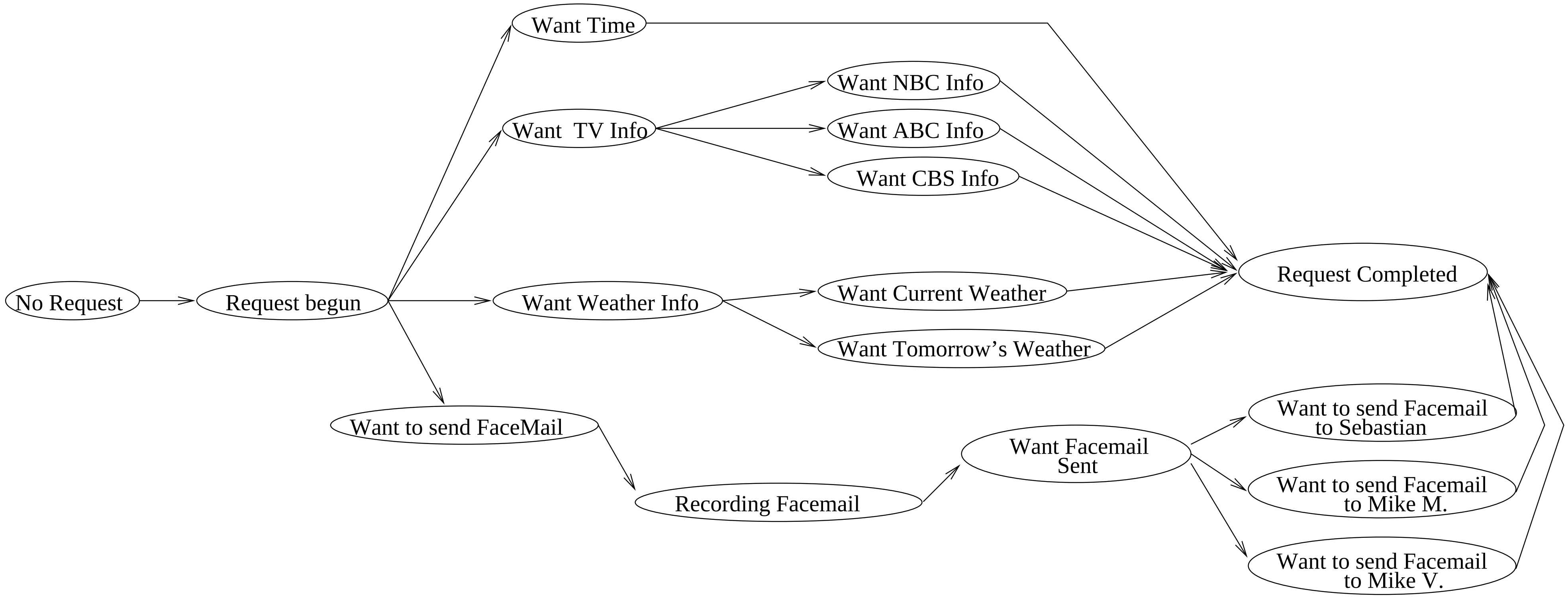
Dialog Management

- State: Represents desire of user
  e.g. `want_tv`, `want_meds`
- Observations: Utterances from speech recogniser
  e.g. "I want to take my pills now"
- Actions: output from speech synthesiser, display changes, or robot actions
  e.g. say "You should take your Prozac now.", or the robot moves to the kitchen.
- Action/observation equations are the same as in navigation

$$
\begin{aligned}
p(\mathbf{s}|\,\mathbf{a}) &= \int_{\mathbf{s}'} p(\mathbf{s}'|\mathbf{s},\mathbf{a})p(\mathbf{s})ds \\
p(\mathbf{s}|\,\mathbf{o}) &= \frac{p(\mathbf{o}|\,\mathbf{s})p(\mathbf{s})}{p(\mathbf{o})}
\end{aligned}
$$

Dialog Management

- Major difference from navigation is in individual models $p(\mathbf{o}|\mathbf{s})$ and $p(\mathbf{s}'|\mathbf{s},\mathbf{a})$
- Action model is given by hand



Dialog Management

- Observation model is given by hand for planning.
- During execution, belief state is updated using observation probabilities output by Sphinx
  - *This leads to possibly suboptimal policies, but in general it leads to better estimates of the user's state.*
- Build in confirmation actions, that serve to clarify the state in highly ambiguous situations
  - Observe: *"Go to Sebastian's Office"*
  - Action: *"Did you say you want me to go to Sebastian Thrun's office?"*
  - Observe: *"Yes."*
  - Action: `robot starts moving towards Sebastian's office.`

Computing the Next State

The state advances in the same manner:

① Reconstruct $p(\mathbf{s})$ under the assumption of a unimodal Gaussian with entropy $H(p(\mathbf{s}))$

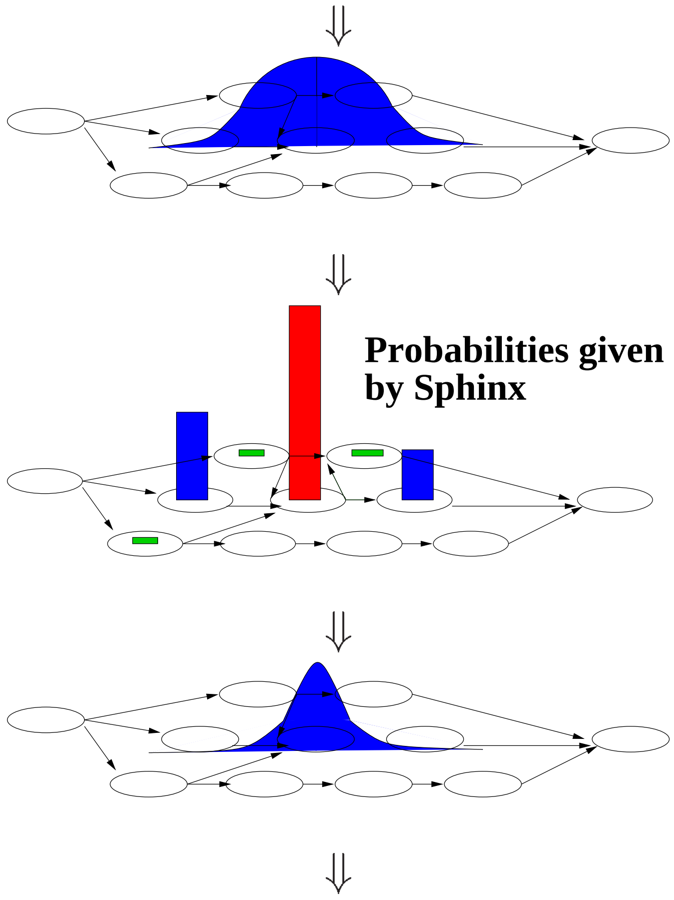
② Apply action $\mathbf{a}$ to get $p(\mathbf{s}|\mathbf{a})$

③ Predict speech utterance $\mathbf{o}$ at position $\text{argmax}_\mathbf{s}\, p(\mathbf{s})$

④ Apply prediction $\mathbf{o}$ to get $p(\mathbf{s}|\mathbf{o})$

⑤ Compress final $p(\mathbf{s})$ to low-dimensional statistic

$$< \text{argmax}_\mathbf{s}\, p(\mathbf{s}); H(p(\mathbf{s})) >$$

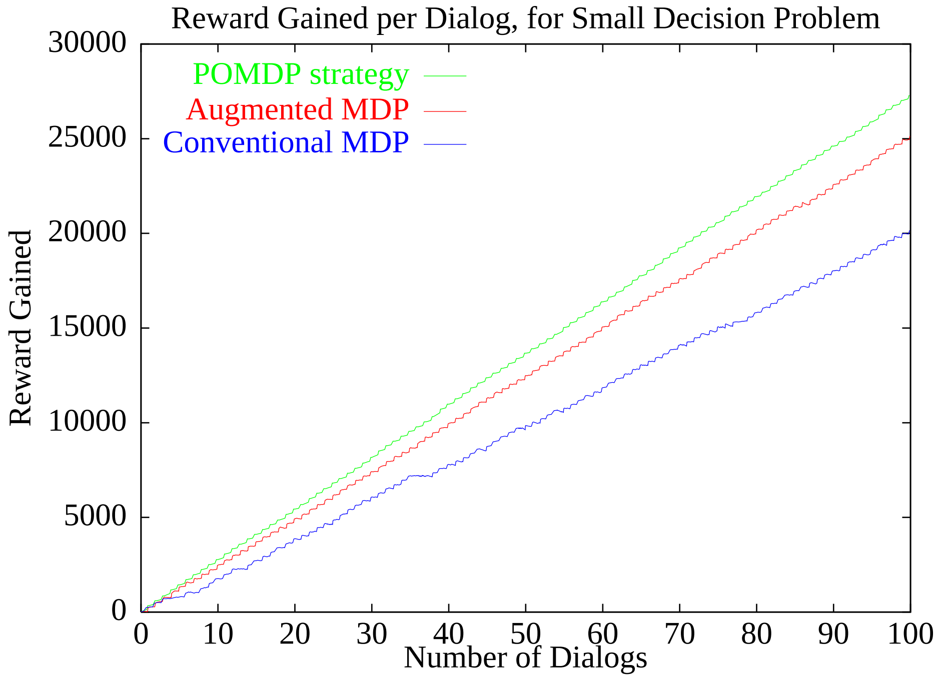


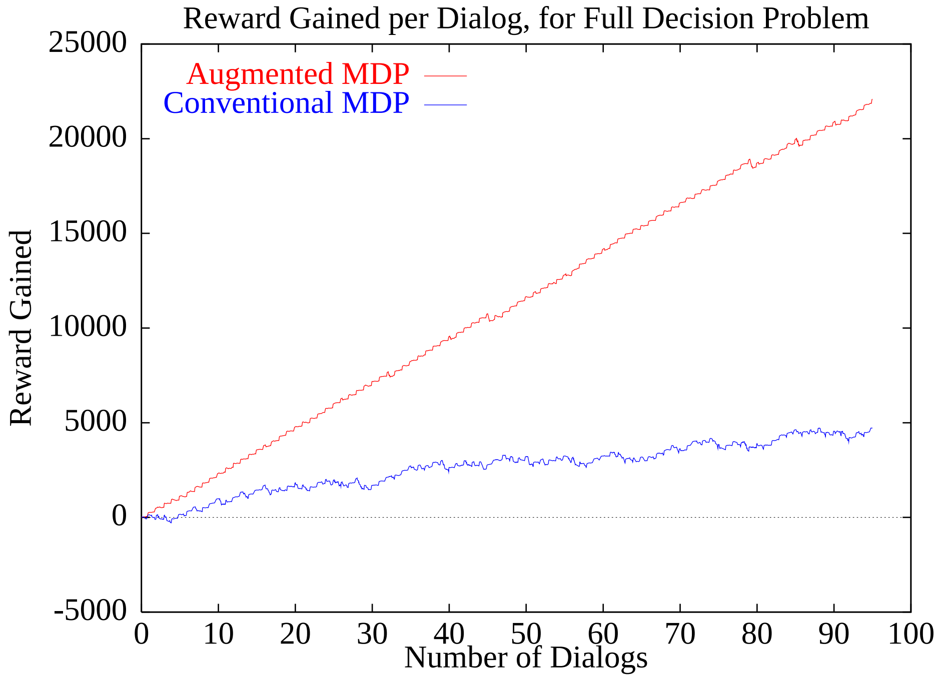
$$< \text{argmax}_\mathbf{s}\, p(\mathbf{s}); H(p(\mathbf{s})) >$$

An Example Dialog

Observation	Belief Entropy	Action	True State
<silence>	0	do_nothing	none
flo hello	2	ask_repeat	request_begun
flo hello	0	say_hello	request_begun
was on a	20	ask_repeat	want_facemail
flo time is	20	ask_repeat	want_facemail
flo i want to deliver please	6	confirm_facemail	want_facemail
flo i want to deliver facemail	0	say_begin_recording	want_facemail
<recording_facemail>	0	record_facemail	recording_facemail
flo send it	1	ask_repeat	want_facemail
flo sent it	0	ask_which_facemail	want_facemail
flo i want to send it to mike	3	ask_repeat	want_facemail_mike_m
flo i want to send it to mike m	0	deliver_facemail_mike_m	want_facemail_mike_m
thanks flo goodbye	0	do_nothing	request_done

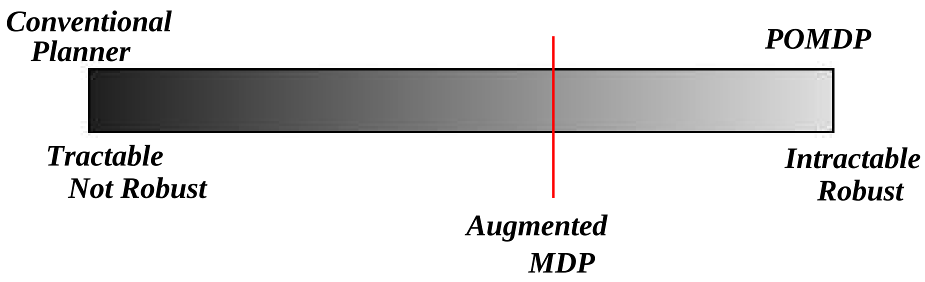
Experimental Results





Differences from Other Approximation Methods

- Can be viewed as solution to POMDP problem, if complete belief state can be reconstructed from single state and entropy.
- Much worse approximation than Witness, etc. for many (most?) problems.
  - ➤ *Rests on the assumption that the entropy and maximum likelihood state are sufficient statistics. Not a general solution method.*
- Better than greedy heuristics
  - ➤ *Represents uncertainty much more appropriately. Does not rely on any particular relationship between the policies from neighbouring states.*



- By explicitly maintaining high state certainty, we stay to the edges of the state space, allowing us to take advantage of the convexity of the value function.

Open Questions

- Over-coming the (increasingly bogus) uni-modal Gaussian assumption
- Non-static observation model
  - ➤ *How much better or worse do we do by taking the observation probability from Sphinx at run-time?*
- Relationship between dialog management and cost-observable MDPs