Approximate approaches to solving grid-world POMDPs for AUVs

Zeyn Saigol

IR Lab, School of Computer Science, University of Birmingham

2nd December 2008

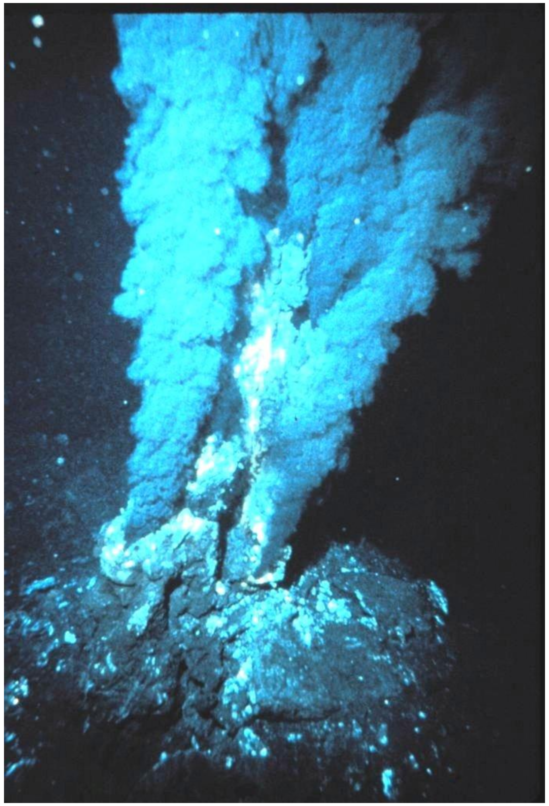
Overview

- Motivation
- Split into mapping and planning
- Simplified model
- Approximate algorithms
  - Information Lookahead (IL)
  - Certainty Equivalent (CE)
- Results
- Future plans

Motivation

Autonomous Underwater Vehicle (AUV)

- Allow more of the oceans to be explored
- Limited on-board intelligence

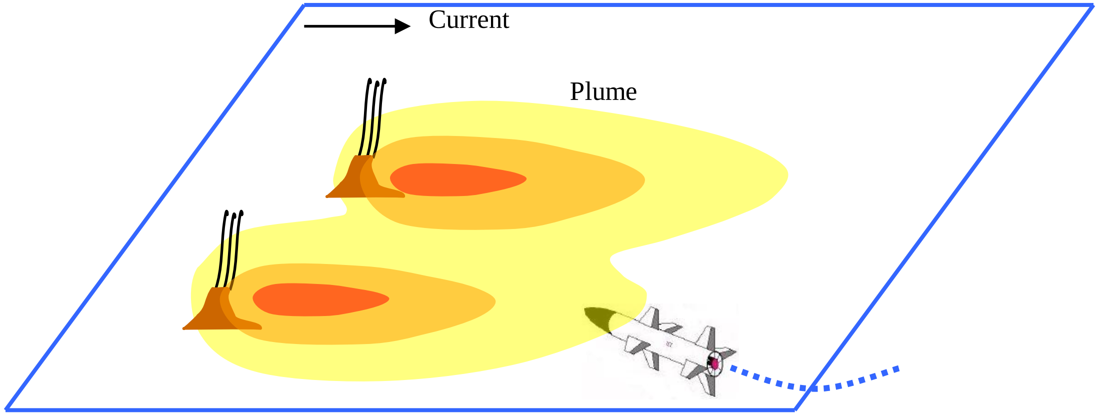


Hydrothermal vents

- Very interesting!
- Emit plume with altered physical properties and dissolved chemicals

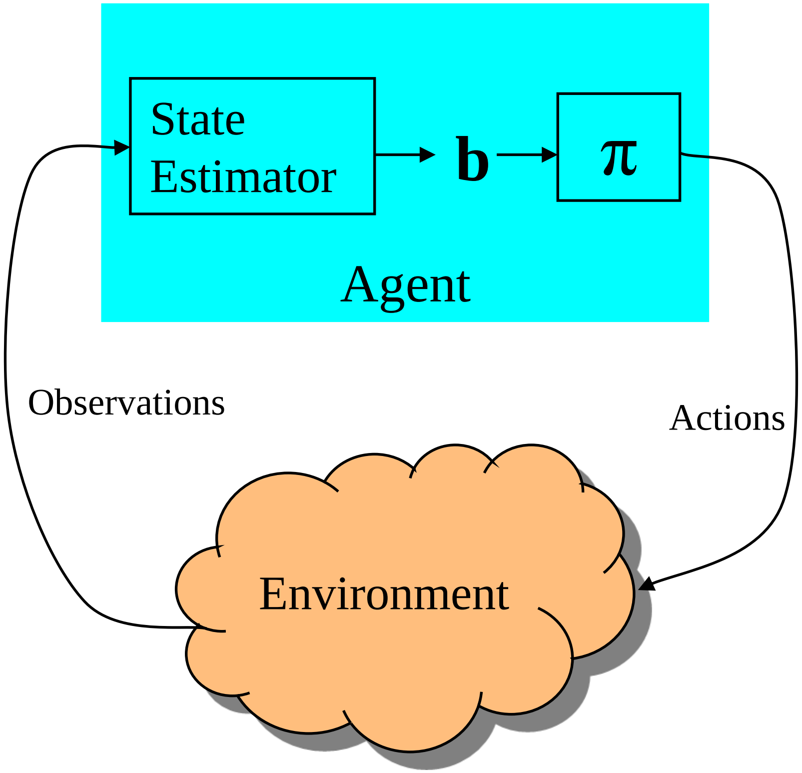
Problem from AI Perspective

- Aim: find as many hydrothermal vents as possible during a mission (limited by available battery power)
- Cannot observe vent locations directly => partial-observability
- State is continuous (AUV location, vent locations, and battery level)
- Standard planning approaches will not cope



POMDPs Recap

- Problem is essentially a POMDP (Partially Observable Markov Decision Process)
- Observation model gives $P(z \mid \mathbf{x})$
- Can't just use observations as state – lose too much information
- Need to keep track of past observations => belief state (**b**)
- Can then reduce to an MDP in belief-state space
- The policy, π, for an MDP can be obtained by solving a set of simultaneous linear equations giving a value to states
- This is done by an algorithm called "value iteration"

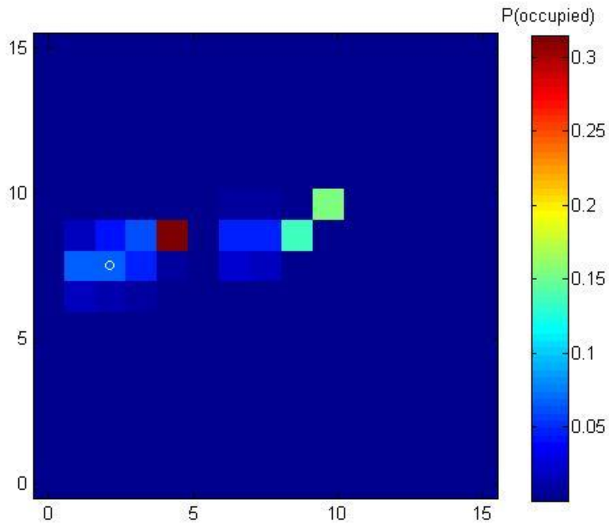


State Estimation – Mapping

- The state estimation component of the POMDP solution can be separated from finding the policy (the planning part)
- State consists of:
  - Location of the AUV
  - Battery level
  - Map of vent locations
- The first two are assumed to be known exactly
- Work by Michael Jakuba solves the problem of creating a map of vent locations from AUV data

Jakuba's OG Algorithm

- Jakuba uses the Occupancy Grid (OG) method
- The area to be mapped is divided into a grid of C cells, each of which can be occupied (i.e. contain a vent) or empty
- The OG consists of a probability of occupancy for each cell
  - So cell occupancies are treated as all independent
  - This reduces the state space from $\Re^{2^C}$ to $\Re^C$



- Jakuba's OG algorithm handles dependency issues that make the standard OG algorithm inadequate for vent mapping
- State estimator is of the form $\mathbf{b}' = \text{jakuba\_og}(\mathbf{b}, a, z)$

Simplified Model

Observation model

- The input to the OG algorithm is a boolean indicating whether or not the plume was detected
- Jakuba uses a simple simulator, where particles are emitted from vents and dispersed by the current plus Gaussian noise

Actions

- Actions are to move N/E/S/W, with deterministic outcomes

Battery level

- Approximated by using a finite number of timesteps

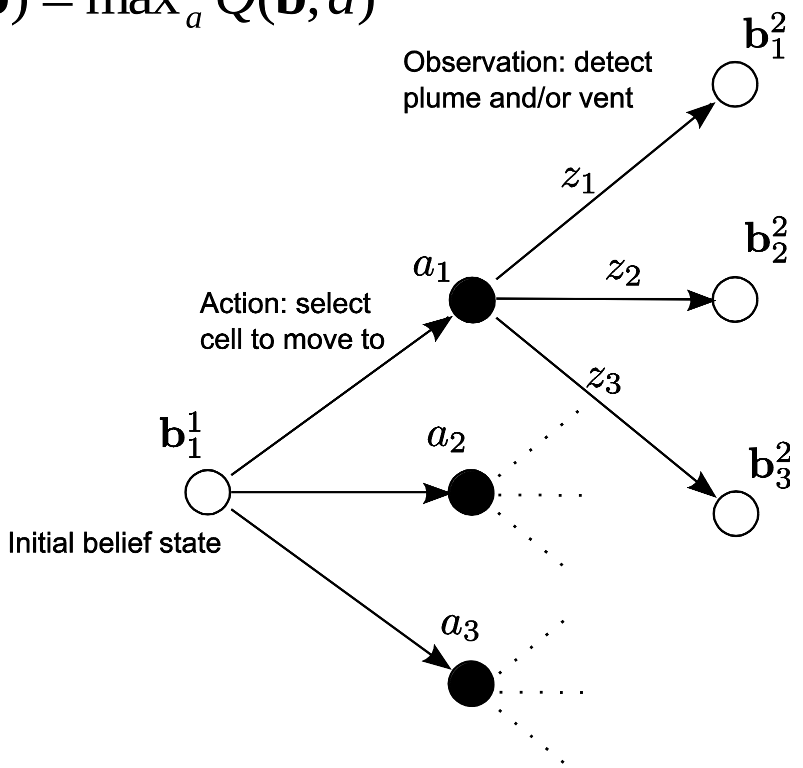
Reward function

- Constant reward R_{vent} for each unique vent found

Information-Lookahead Algorithm

- Basic idea: find the expected reward (=value) of choosing an action and then following the optimal policy
- Belief-state values V and Q-values for an action and belief state are interchangeable: $V(\mathbf{b}) = \max_a Q(\mathbf{b}, a)$
- Information-Lookahead (IL) algorithm:

a) Uses the start belief state (which is assumed to be known exactly) to limit the number of states that have to be evaluated (=> on-line)

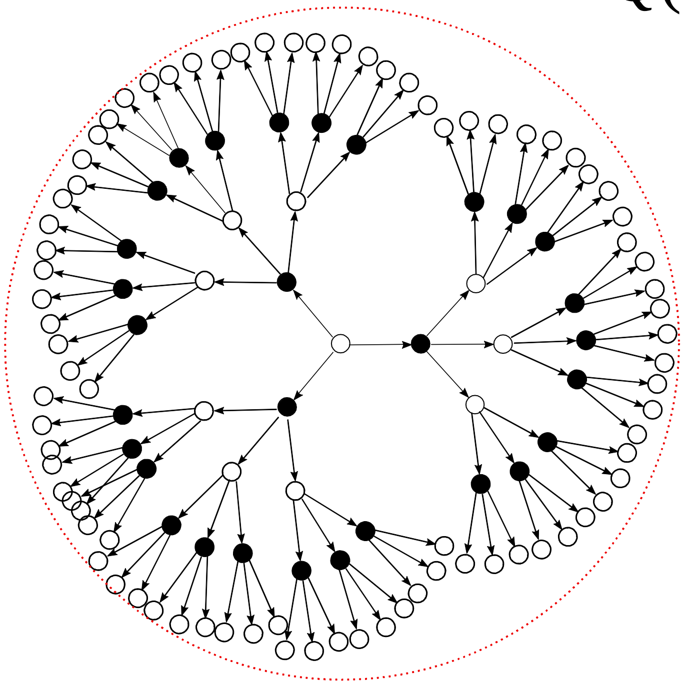


IL Algorithm (2)

b) Assumes a finite horizon, which combined with (a) produces a finite set of belief states to evaluate

c) Calculate node values using

$$Q(\mathbf{b},a) = R(\mathbf{b},a) + \gamma \sum_{\mathbf{b}'} P(\mathbf{b}' \mid \mathbf{b},a) \max_{a'} Q(\mathbf{b}',a')$$



d) Use a heuristic to calculate values for leaf nodes

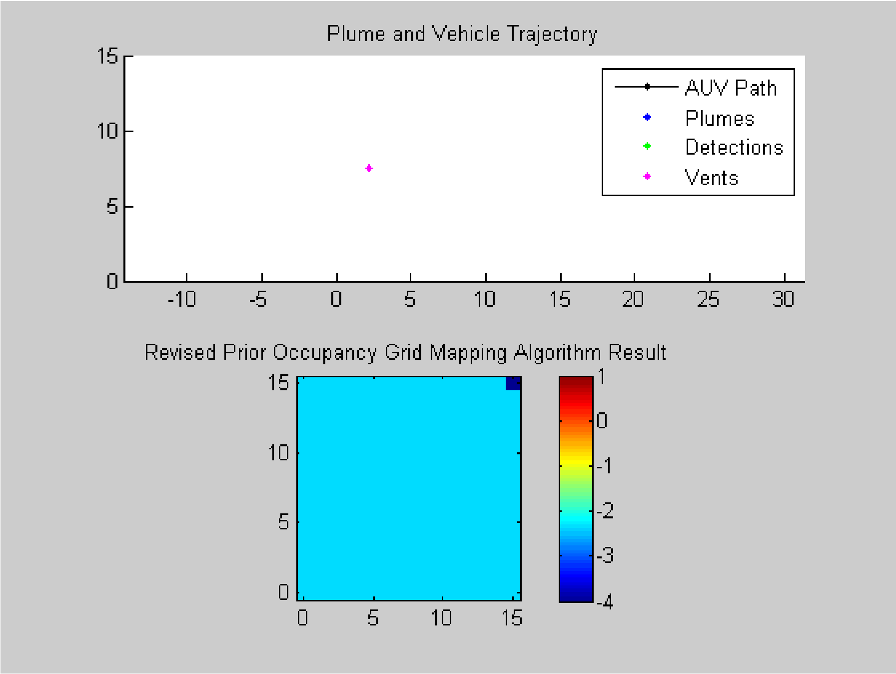
e) Calculate values backwards from leaf nodes, which avoids having to iterate until values converge

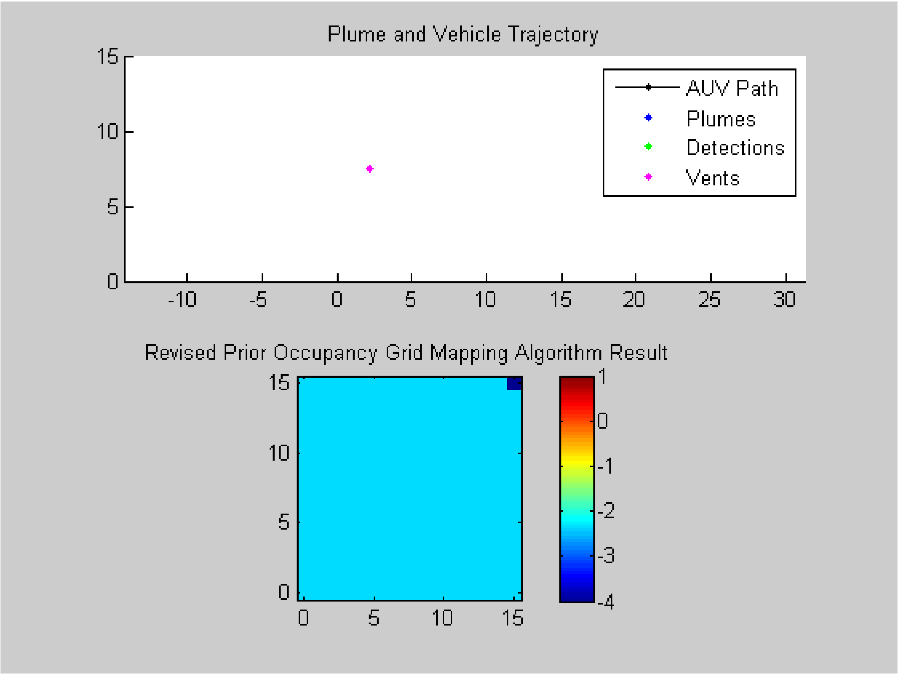
Certainty-Equivalent Heuristic

- Each leaf node is a belief state, which is essentially an OG map
- The Certainty-Equivalent heuristic assigns a value to an OG by assuming that it is correct – so future observations will not lead to better information about where vents are
- This reduces the problem to a fully-observable MDP, which can be solved efficiently
- Effectively this approach ignores potential gains in information we may expect by choosing certain actions over others

Results

- IL and cell-value heuristic
  - Value of leaf nodes is just P(occ) for that cell





- IL-CE planner

Future Plans

- Investigate potential problems with the calculation of CE values
  - Allows reward for visiting same vent multiple times
  - May need a "prize-gathering travelling salesman" type solution
- Improve performance! Can only use a lookahead of 2 steps
- Check how worthwhile a 2-step lookahead is, by comparison with the optimal policy (which can be found using a large lookahead and a tiny grid)
- Try alternatives to the OG mapping approach
  - Converting sensor readings to a binary detection output is throwing away information
  - OG is not able to represent either-or scenarios, where either one set of vents or another set could produce similar sensor readings

References

- Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134.
- Jakuba, M. (2007). *Stochastic Mapping for Chemical Plume Source Localization with Application to Autonomous Hydrothermal Vent Discovery*. PhD thesis, MIT and WHOI Joint Program.