Path Planning in Dynamic Environments with the Partially Observable Canadian Traveller's Problem

Mikko Lauri and Risto Ritala
Tampere University of Technology
Tampere, Finland

Autonomous mobile work machines

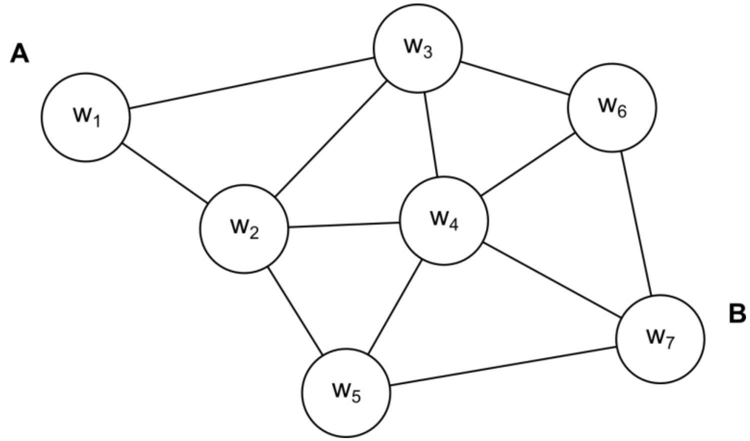
- Increasing level of automation in industrial applications of mobile work machines
- From separation to co-operation
- Dynamics and partial observability

Contents

- Model for navigation tasks in dynamic environments
- Taking advantage of the structure of the problem
- Experimental results
- Implementation of a path planning module for a mobile ground robot
- Conclusions and future work

Canadian Traveller's Problem (CTP)

- An agent wants to find the shortest path between two vertices A and B in a graph.
- An environment state w_i persists at each vertex.
- Edge traversal costs between vertices are determined by the environment states.
- Environment states are initially unknown, but revealed as agent moves.



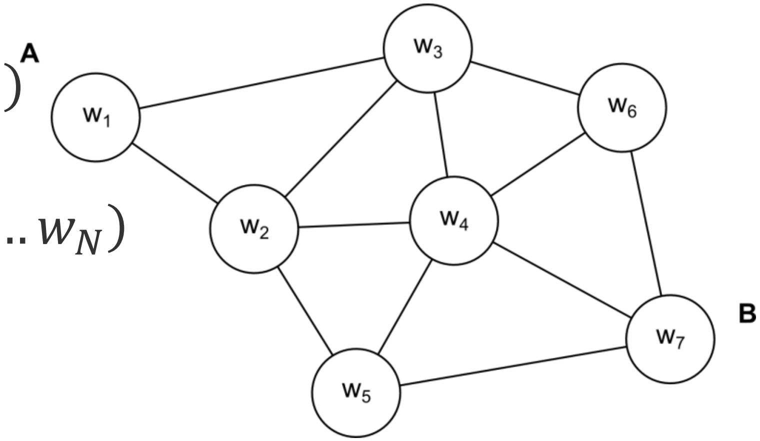
Partially observable CTP

- To model more realistic problems, define a partially observable CTP: environment states evolve by a Markov chain and are partially observable.
- Further assumptions: Environment states independent of the actions a, robot's location x is still fully observable.

- Probabilistic models:

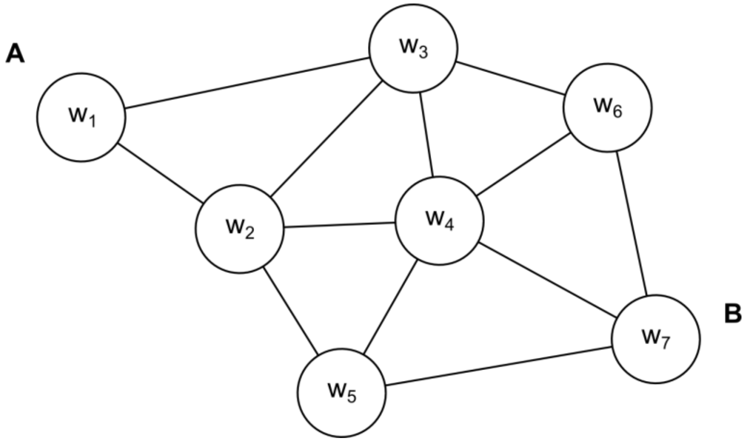
$$p(w_i{}'|w_i)$$

$$p(o|a, x, w_1, \dots w_N)$$



Partially observable CTP

- Minimize the expected cost to reach goal.
- The problem can be formulated as a partially observable Markov decision process (POMDP).
- Belief state of POMDP is mixed observable: $b = (x, w)$ where x is fully observable robot location and w the environment state.



Modelling navigation problems

- Partially observable CTPs can model realistic navigation problems in dynamic environments under the following two assumptions:

- Independence: Robot’s actions do not affect how environment evolves.

- Full observability of robot location: requires accurate localization data or adequate level of abstraction.

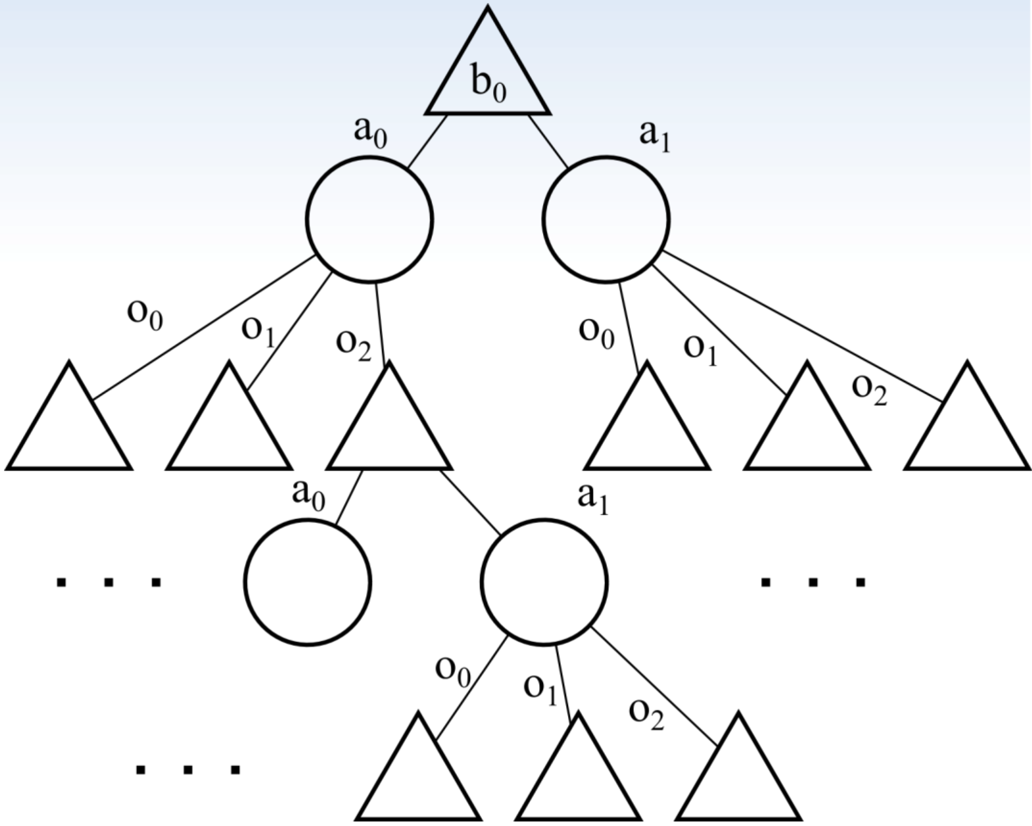
Planning in a partially observable CTP

- With set of vertices V, state space size $|S| = |V| \cdot K^{|V|}$, where K is the number of values the environment variables may assume – challenging for offline POMDP solvers to find good policies!

- On-line planning
  - Finds actions for single belief state at a time: can deal with large state spaces.
  - Allows flexibility in handling changes in underlying models, e.g. new information on environment dynamics.

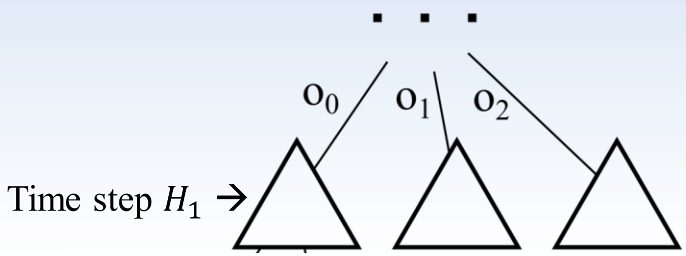
Online planning

- Finite-horizon lookahead from current belief b_0, estimation of the value of nodes by a heuristic function.
- Propagate values to root, select best action.
- Repeat after each decision and observation cycle.



Online planning: An approximation

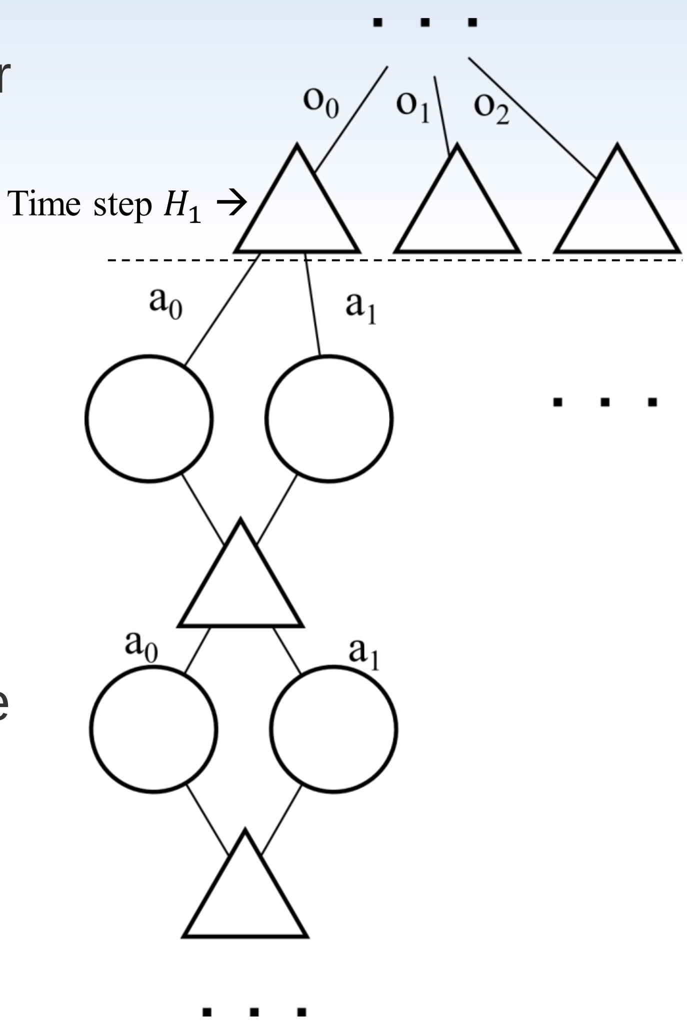
- In a PO-CTP, estimating the value of a decision tree node *b* is equivalent to finding the expected cost to reach the goal from *b.*



- Make a simplifying assumption: after H_1 time steps, robot no longer has access to observations of environment states.

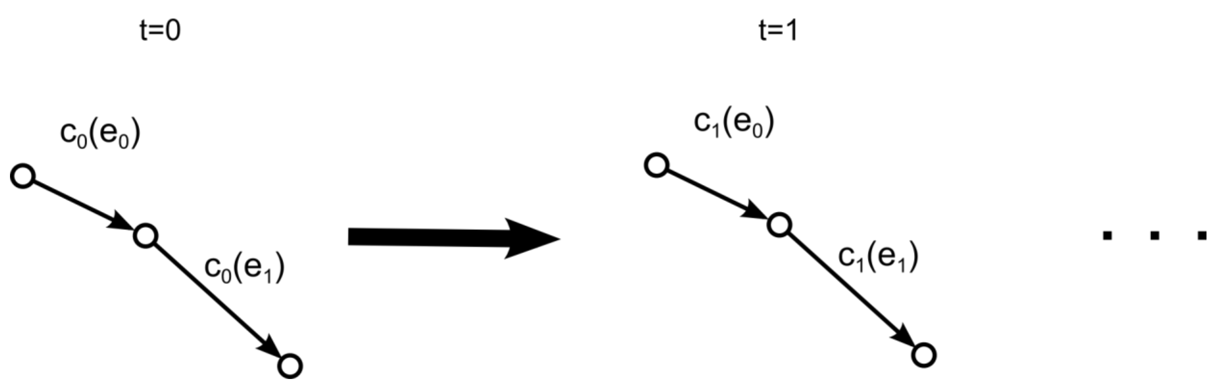
Online planning: An approximation

- If no observations are available after H_1, belief over the environment state forms a single branch.
- Any action by the robot results in the same belief over the environment state at next time step.
- Based on dynamics model, can predict belief over environment state for any time step after H_1.



Online planning: An approximation

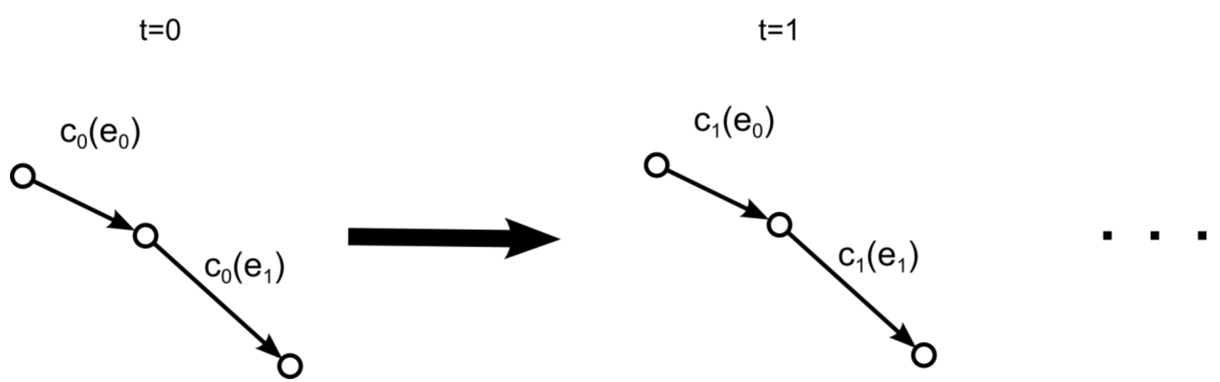
- Belief over the environment state can be predicted for an arbitrary time step => **Expected** edge traversal costs can be predicted as well.
- After H_1, can estimate cost to reach the goal by shortest path search in a graph whose edge costs are equal to the expected costs.



c_t known for each t

Online planning: An approximation

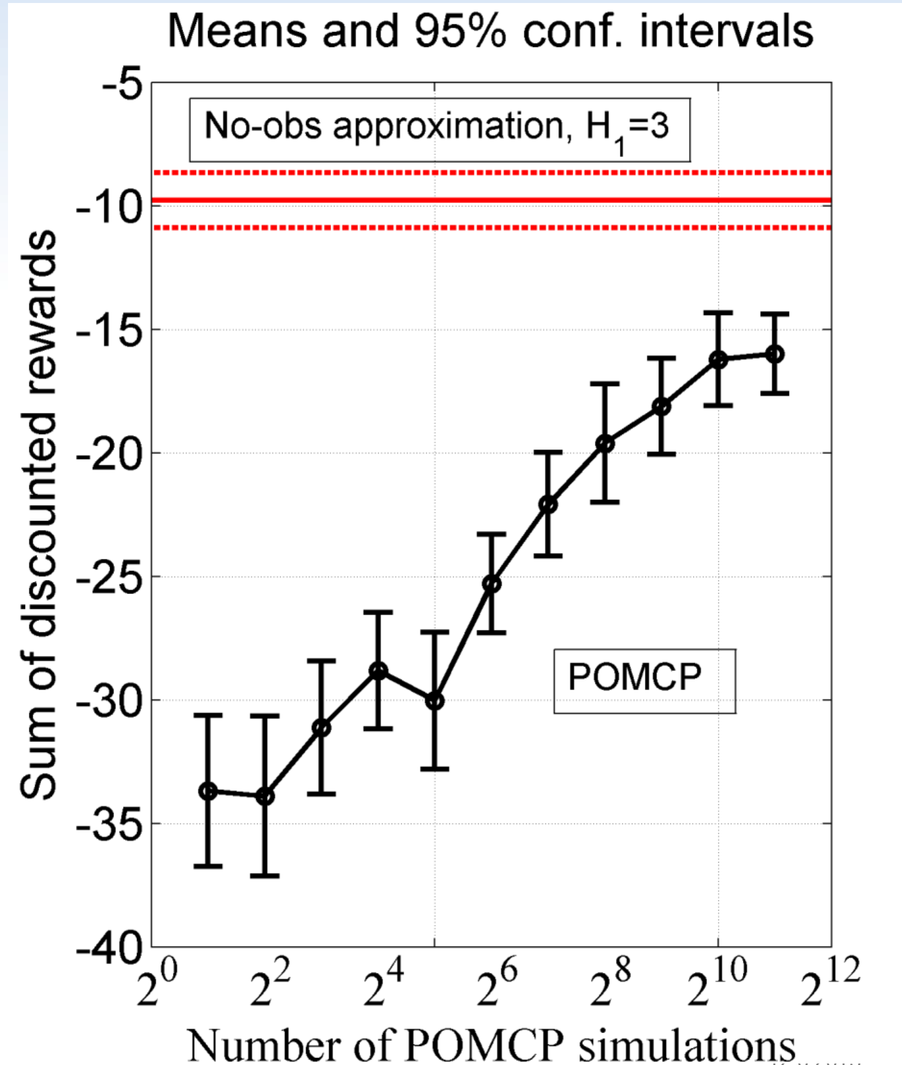
- We assumed that after a finite number of time steps H_1, agent loses ability to observe environment state.
- As a consequence, expected edge traversal costs may be predicted for any time step after H_1.



c_t known for each t

Experimental results

- Simulation in a grid world, robot tries to avoid occupied areas.
- Compared against POMCP, based on Monte Carlo tree search.
- Good quality solutions even with short lookahead horizon.
- No-obs approximation effectively uses information on graph structure.

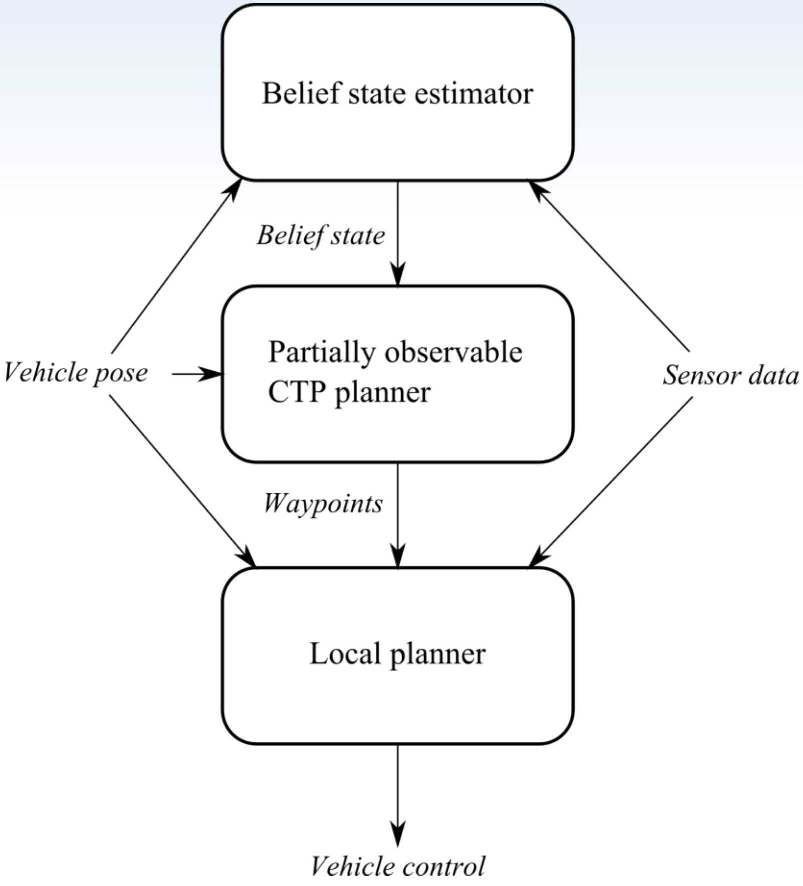


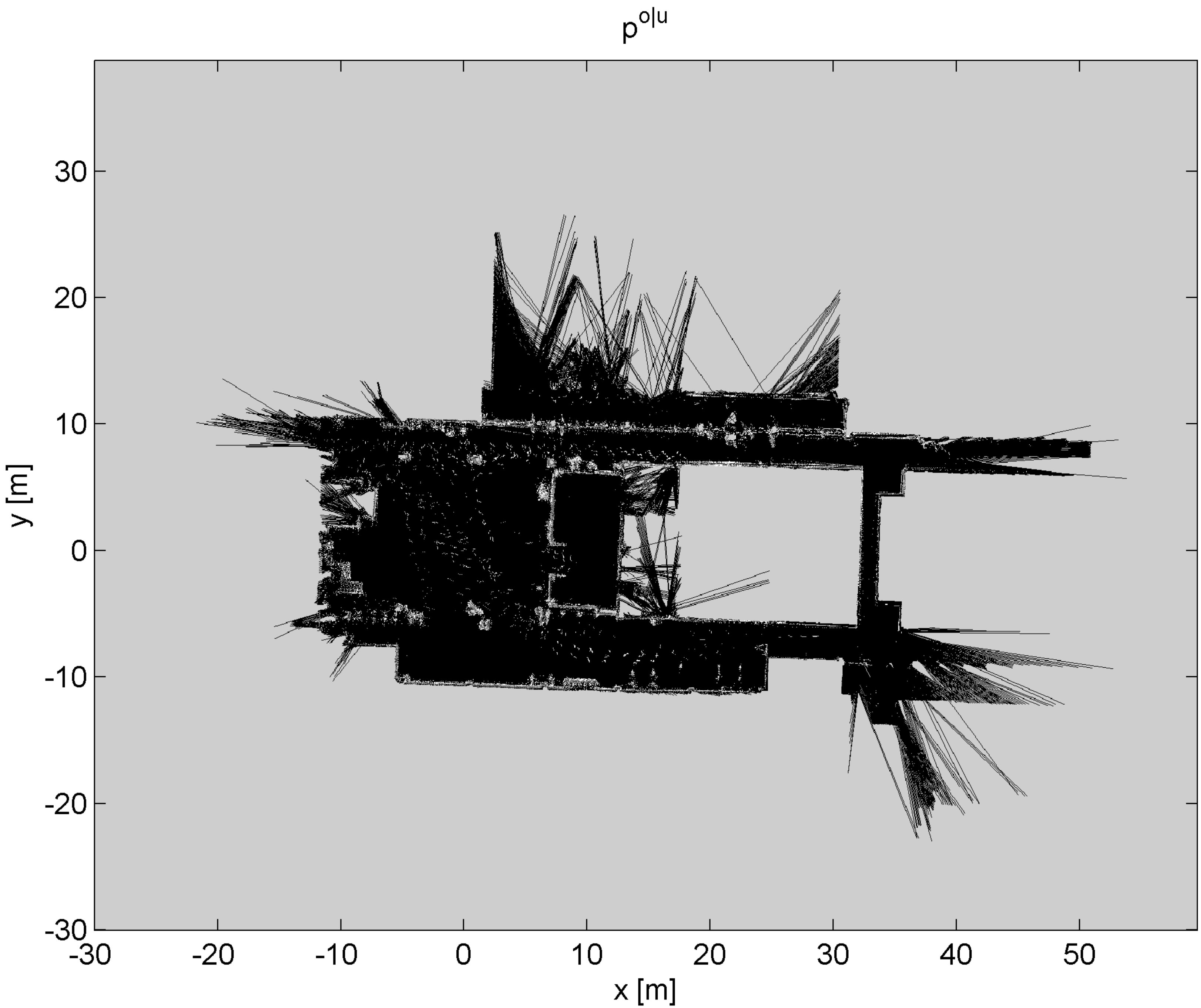
Experimental results

- There are also other methods appropriate for comparison
  - Taking advantage of mixed observability of belief state (Ong et. al., 2010)
  - Goal-POMDPs and Real-time dynamic programming (Bonet & Geffner, 2009)

Towards a practical implementation

- Suited for planning on topological maps.
- Output waypoints to a local planner generating feasible control signals.
- Learning of environment dynamics.





p^{o|u}
y [m]
x [m]
30
20
10
0
-10
-20
-30
-30
-20
-10
0
10
20
30
40
50

Conclusion

- Partially observable CTPs can model realistic navigation problems in dynamic, partially observable environments.
- Special structure of the problem can be exploited to find approximate solutions to the planning problem.
- Future work
  - Comparison with other methods
  - Extensions: e.g. information-gathering
  - Implementation

Thank you for your attention!

References:

Ong, S. C.W.; Png, S.W.; Hsu, D.; and Lee,W. 2010. Planning under Uncertainty for Robotic Tasks with Mixed Observability. The International Journal of Robotics Research 29(8):1053–1068.

Bonet, B., and Geffner, H. 2009. Solving POMDPs: RTDP-Bel vs. point-based algorithms. In Proceedings of the Twenty-Second International Joint Conference on Artificla Intelligence (IJCAI), 1641–1646.