Optimal Fixed-Size Controllers for Decentralized POMDPs

Christopher Amato
Daniel S. Bernstein
Shlomo Zilberstein
University of Massachusetts Amherst

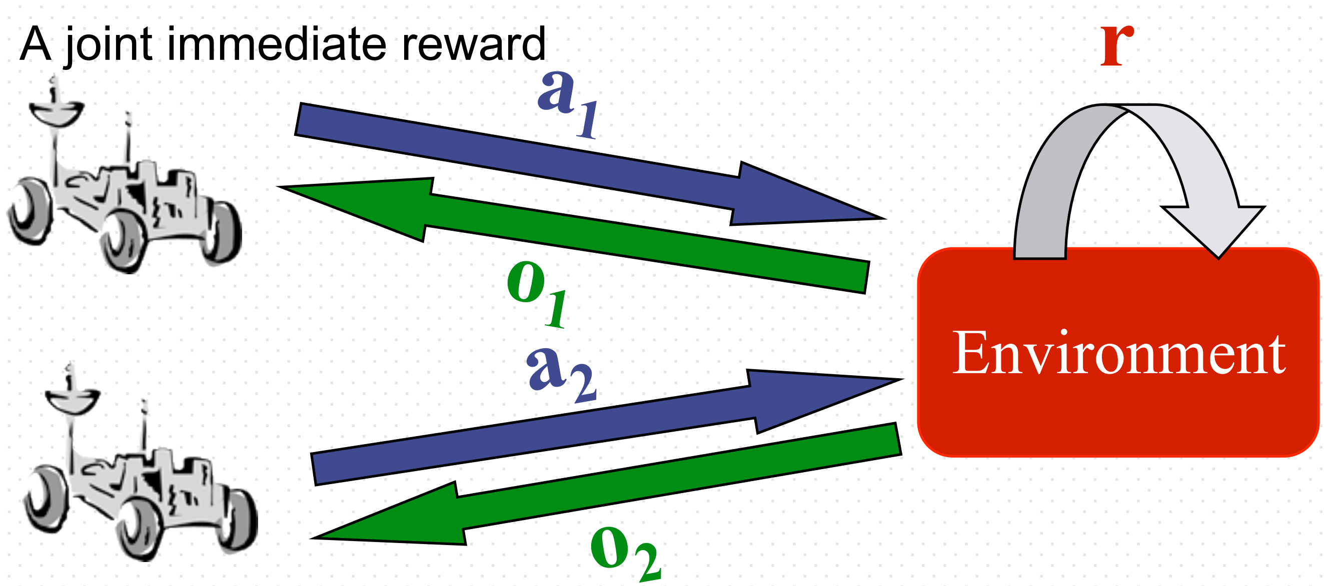
May 9, 2006

Overview

- DEC-POMDPs and their solutions
- Fixing memory with controllers
- Previous approaches
- Representing the optimal controller
- Some experimental results

DEC-POMDPs

- Decentralized partially observable Markov decision process (DEC-POMDP)
- Multiagent sequential decision making under uncertainty
  - At each stage, each agent receives:
    - A local observation rather than the actual state
    - A joint immediate reward



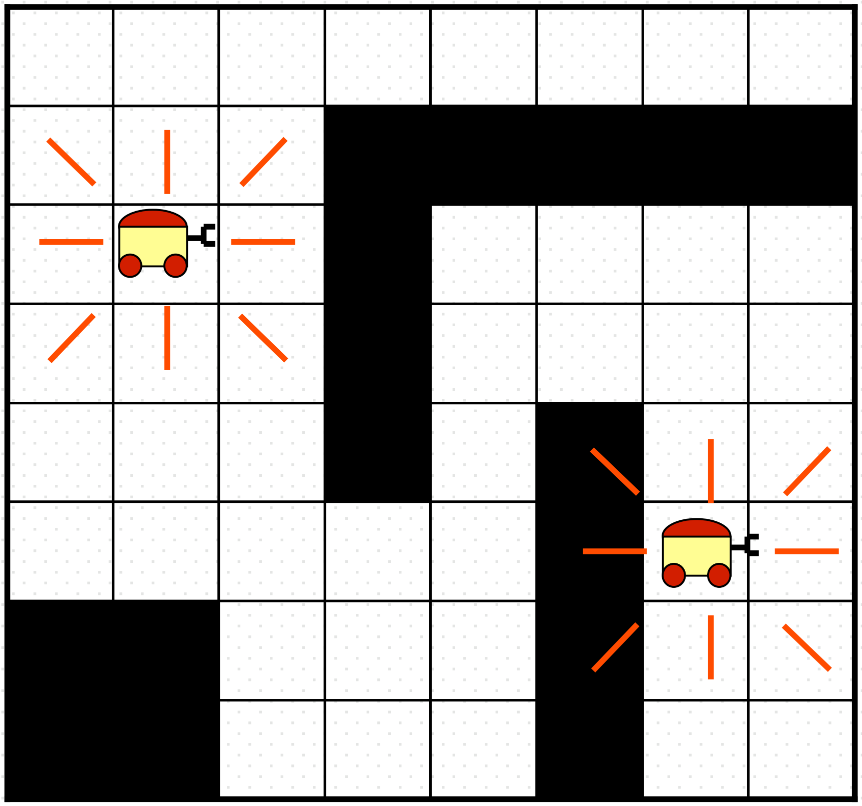
DEC-POMDP definition

- A two agent DEC-POMDP can be defined with the tuple: $M = \langle S, A_1, A_2, P, R, \Omega_1, \Omega_2, O\rangle$
  - S, a finite set of states with designated initial state distribution b_0
  - A_1 and A_2, each agent's finite set of actions
  - P, the state transition model: $P(s' \mid s, a_1, a_2)$
  - R, the reward model: $R(s, a_1, a_2)$
  - Ω_1 and Ω_2, each agent's finite set of observations
  - O, the observation model: $O(o_1, o_2 \mid s', a_1, a_2)$

DEC-POMDP solutions

- A policy for each agent is a mapping from their observation sequences to actions, $\Omega^* \rightarrow A$, allowing distributed execution
- A joint policy is a policy for each agent
- Goal is to maximize expected discounted reward over an infinite horizon
- Use a discount factor, γ, to calculate this

Example: Grid World



States: grid cell pairs

Actions: move ⇧, ⇩, ⇨, ⇦, stay

Transitions: noisy

Observations: red lines

Goal: share same square

Previous work

- Optimal algorithms
  - Very large space and time requirements
  - Can only solve small problems
- Approximation algorithms
  - provide weak optimality guarantees, if any

Policies as controllers

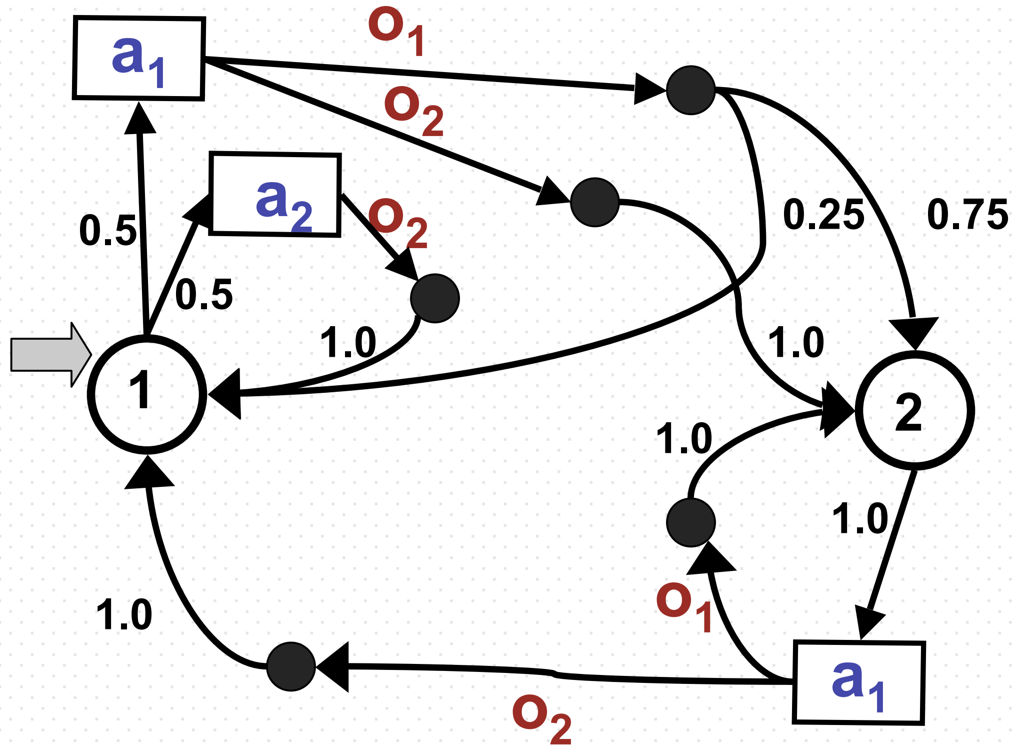
- Finite state controller for each agent *i*
  - Fixed memory
  - Randomness used to offset memory limitations
  - Action selection, $\psi : Q_i \rightarrow \Delta A_i$
  - Transitions, $\eta : Q_i \times A_i \times O_i \rightarrow \Delta Q_i$
- Value for a pair is given by the Bellman equation:

$$V(q_1,q_2,s) = \sum_{a_1,a_2} P(a_1 \mid q_1) P(a_2 \mid q_2) \Big[R(s,a_1,a_2) + \gamma \sum_{s'} P(s' \mid s,a_1,a_2) \sum_{o_1,o_2} O(o_1,o_2 \mid s',a_1,a_2) \sum_{q_1',q_2'} P(q_1' \mid q_1,a_1,o_1) P(q_2' \mid q_2,a_2,o_2) V(q_1',q_2',s') \Big]$$

Where the subscript denotes the agent and lowercase values are elements of the uppercase sets above

Controller example

- Stochastic controller for a single agent
  - 2 nodes, 2 actions, 2 obs
  - Parameters
    - P(a|q)
    - P(q'|q,a,o)



Optimal controllers

- How do we set the parameters of the controllers?

- Deterministic controllers - traditional methods such as best-first search (Szer and Charpillet 05)
- Stochastic controllers - continuous optimization

Decentralized BPI

- Decentralized Bounded Policy Iteration (DEC-BPI) - (Bernstein, Hansen and Zilberstein 05)

- Alternates between improvement and evaluation until convergence
- Improvement: For each node of each agent's controller, find a probability distribution over one-step lookahead values that is greater than the current node's value for all states and controllers for the other agents
- Evaluation: Finds values of all nodes in all states

DEC-BPI - Linear program

NEED TO FIX THIS SLIDE IF I WANT TO USE IT!

For a given node, q

Variables: ε, $P(a_i, q_i{'}|q_i, o_i)$

Objective: Maximize ε

Improvement Constraints: $\forall s \in S,\ q_{-i} \in Q_{-i}$

$$V(s,\vec{q}) + \varepsilon \le \sum_{\vec{a}} P(\vec{a}\,|\,\vec{q})\left[R(s,\vec{a}) + \gamma \sum_{s',\vec{o},\vec{q}'} P(\vec{q}'\,|\,\vec{q},\vec{a},\vec{o})P(s'\,|\,s,\vec{a})P(\vec{o}\,|\,s',\vec{a})V(s',\vec{q}')\right]$$

$$\sum_{q'} x(q',a,o) = x(a)$$

Probability constraints: $\forall a \in A$

Also, all probabilities must sum to 1 and be greater than 0

Problems with DEC-BPI

- Difficult to improve value for all states and other agents' controllers
- May require more nodes for a given start state
- Linear program (one step lookahead) results in local optimality

- Correlation device can somewhat improve performance

Optimal controllers

- Use nonlinear programming (NLP)
- Consider node value as a variable
- Improvement and evaluation all in one step
- Add constraints to maintain valid values

NLP intuition

- Value variable allows improvement and evaluation at the same time (infinite lookahead)
- While iterative process of DEC-BPI can “get stuck” the NLP does define the globally optimal solution

NLP representation

Variables:

$$x(\vec{q},\vec{a}) = P(\vec{a} \mid \vec{q}),\ y(\vec{q},\vec{a},\vec{o},\vec{q}\,') = P(\vec{q}\,' \mid \vec{q},\vec{a},\vec{o}),\ z(\vec{q},s) = V(\vec{q},s)$$

Objective: Maximize $\sum_s b_0(s) z(\vec{q}_0, s)$

Value Constraints: $\forall s \in S,\ \vec{q} \in Q$

$$z(\vec{q},s) = \sum_{\vec{a}} x(\vec{q}\,',\vec{a}) \left[R(s,\vec{a}) + \gamma \sum_{s'} P(s' \mid s,\vec{a}) \sum_{\vec{o}} O(\vec{o} \mid s',\vec{a}) \sum_{\vec{q}\,'} y(\vec{q},\vec{a},\vec{o},\vec{q}\,') z(\vec{q}\,',s') \right]$$

Linear constraints are needed to ensure controllers are independent

Also, all probabilities must sum to 1 and be greater than 0

Optimality

Theorem: An optimal solution of the NLP results in optimal stochastic controllers for the given size and initial state distribution.

Pros and cons of the NLP

- Pros
  - Retains fixed memory and efficient policy representation
  - Represents optimal policy for given size
  - Takes advantage of known start state
- Cons
  - Difficult to solve optimally

Experiments

- Nonlinear programming algorithms (snopt and filter) - sequential quadratic programming (SQP)
- Guarantees locally optimal solution
- NEOS server
- 10 random initial controllers for a range of sizes
- Compared the NLP with DEC-BPI
  - With and without a small correlation device

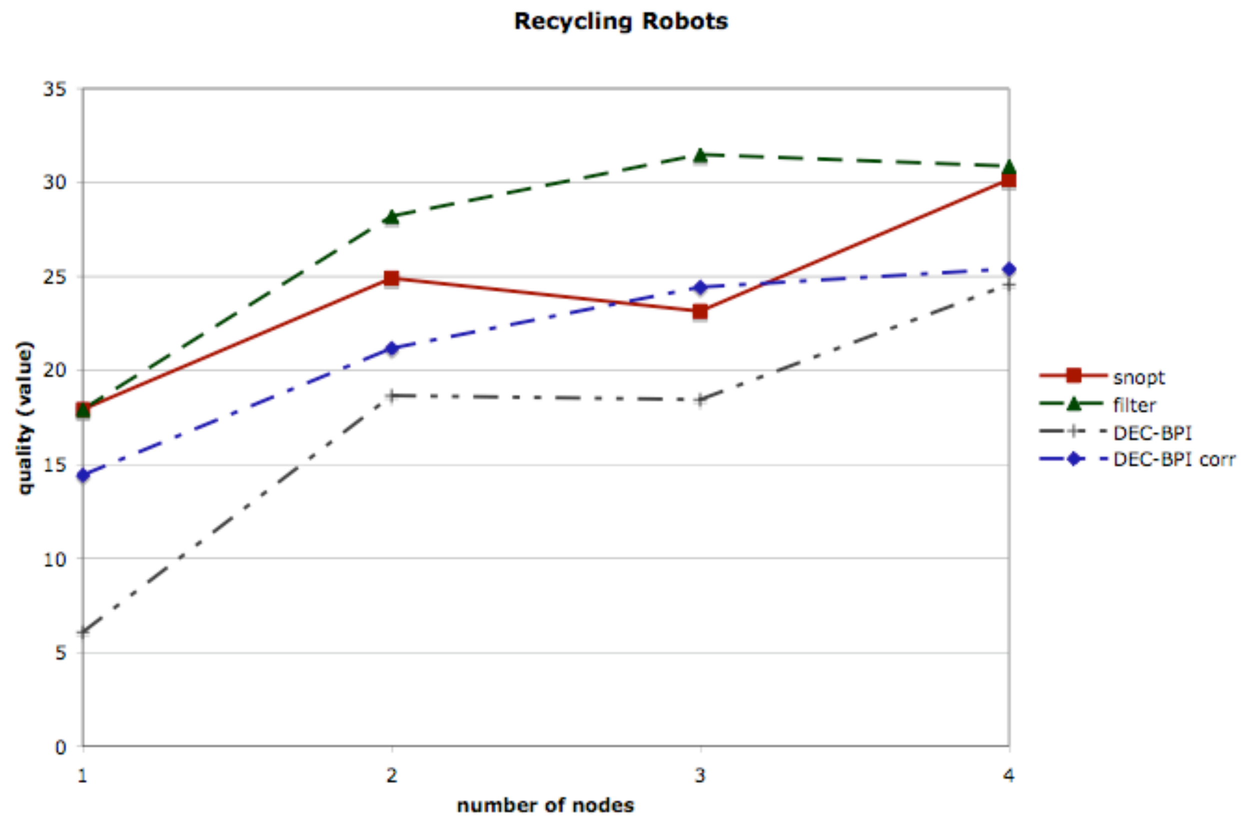
Results: Broadcast Channel

- Two agents share a broadcast channel (4 states, 5 obs , 2 acts)
- Very simple near-optimal policy

# nodes	BPI ind	BPI cor	NLP algs
1	4.687	6.290	9.1
2	4.068	7.749	9.1
3	8.637	7.781	9.1
4	7.857	8.165	9.1

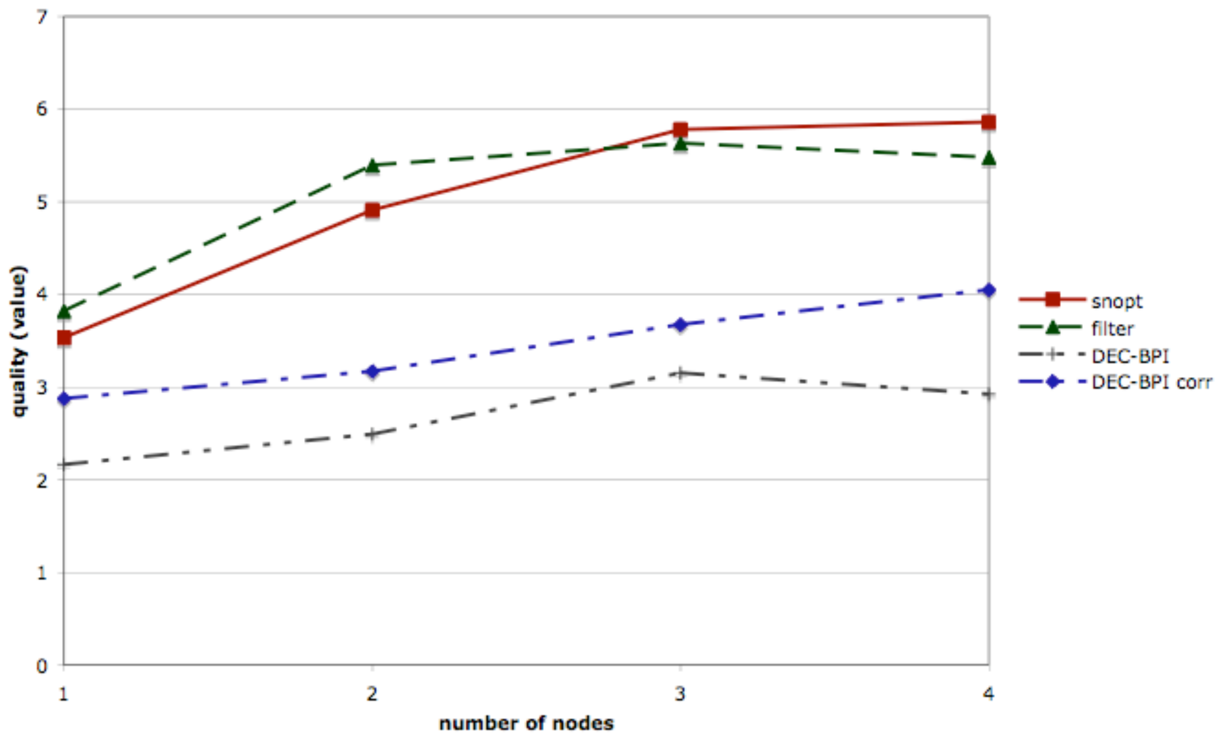
mean quality of the NLP and DEC-BPI implementations

Results: Recycling Robots



mean quality of the NLP and DEC-BPI implementations on the recycling robot domain (4 states, 2 obs, 3 acts)

Results: Grid World



mean quality of the NLP and DEC-BPI implementations on the meeting in a grid (16 states, 2 obs, 5 acts)

Results: Running time

- Running time mostly comparable to DEC-BPI corr
- The increase as controller size grows offset by better performance

Broadcast

# nodes	snopt	filter	DEC-BPI	DEC-BPI corr
1	1s	1s	$<$ 1s	$<$ 1s
2	2s	3s	$<$ 1s	2s
3	14s	764s	2s	7s
4	188s	4061s	5s	24s

Recycle

# nodes	snopt	filter	DEC-BPI	DEC-BPI corr
1	1s	1s	$<$ 1s	$<$ 1s
2	2s	4s	$<$ 1s	1s
3	26s	64s	1s	3s
4	523s	635s	3s	10s

Grid

# nodes	snopt	filter	DEC-BPI	DEC-BPI corr
1	3s	2s	1s	2s
2	4s	5s	8s	31s
3	54s	110s	39s	151s
4	873s	2098s	118s	638s

Conclusion

- Defined the optimal fixed-size stochastic controller using NLP
- Showed consistent improvement over DEC-BPI with locally optimal solvers
- In general, the NLP may allow small optimal controllers to be found
- Also, may provide concise near-optimal approximations of large controllers

Future Work

- Explore more efficient NLP formulations
- Investigate more specialized solution techniques for NLP formulation
- Greater experimentation and comparison with other methods