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
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: $P(s'| s, a_1, a_2)$
- $R$, the reward model: $R(s, a_1, a_2)$
- $\Omega_1$ and $\Omega_2$, each agent’s finite set of observations
- $O$, the observation model: $O(o_1, o_2| s', a_1, a_2)$
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, $\gamma$, 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:

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

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
    - $P(a|q)$
    - $P(q'|q,a,o)$

![Diagram](image-url)
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
NEED TO FIX THIS SLIDE IF I WANT TO USE IT!

For a given node, \( q \)

Variables: \( \varepsilon, P(a_i, q_i', | q_i, o_i) \)

Objective: Maximize \( \varepsilon \)

Improvement Constraints: \( \forall S \in S, q_{-i} \in Q_{-i} \)

\[
V(s, \bar{q}) + \varepsilon \leq \sum_{\bar{a}} P(\bar{a} | \bar{q}) R(s, \bar{a}) + \gamma \sum_{\bar{a}, \bar{q}} P(\bar{q} | \bar{a}, \bar{q}) P(s' | s, \bar{a}) P(\bar{o} | s', \bar{a}) V(s', \bar{q}')
\]

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

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
Variables:
\[ x(\bar{q}, \bar{a}) = P(\bar{a} \mid \bar{q}) \, , \, y(\bar{q}, \bar{a}, \bar{o}, \bar{q}') = P(\bar{q}' \mid \bar{q}, \bar{a}, \bar{o}) \, , \, z(\bar{q}, s) = V(\bar{q}, s) \]

Objective: Maximize \[ \sum_{s} b_0(s)z(\bar{q}_0, s) \]

Value Constraints: \( \forall s \in S, \, \bar{q} \in Q \)
\[ z(\bar{q}, s) = \sum_{\bar{a}} x(\bar{q}', \bar{a}) \left[ R(s, \bar{a}) + \gamma \sum_{s'} P(s' \mid s, \bar{a}) \sum_{\bar{o}} O(\bar{o} \mid s', \bar{a}) \sum_{\bar{q}'} y(\bar{q}, \bar{a}, \bar{o}, \bar{q}') z(\bar{q}', s') \right] \]

Linear constraints are needed to ensure controllers are independent
Also, all probabilities must sum to 1 and be greater than 0
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
Two agents share a broadcast channel (4 states, 5 obs, 2 acts)

Very simple near-optimal policy

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

<table>
<thead>
<tr>
<th># nodes</th>
<th>snopt</th>
<th>filter</th>
<th>DEC-BPI</th>
<th>DEC-BPI corr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Broadcast</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1s</td>
<td>1s</td>
<td>&lt; 1s</td>
<td>&lt; 1s</td>
</tr>
<tr>
<td>2</td>
<td>2s</td>
<td>3s</td>
<td>&lt; 1s</td>
<td>2s</td>
</tr>
<tr>
<td>3</td>
<td>14s</td>
<td>764s</td>
<td>2s</td>
<td>7s</td>
</tr>
<tr>
<td>4</td>
<td>188s</td>
<td>4061s</td>
<td>5s</td>
<td>24s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># nodes</th>
<th>snopt</th>
<th>filter</th>
<th>DEC-BPI</th>
<th>DEC-BPI corr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recycle</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1s</td>
<td>1s</td>
<td>&lt; 1s</td>
<td>&lt; 1s</td>
</tr>
<tr>
<td>2</td>
<td>2s</td>
<td>4s</td>
<td>&lt; 1s</td>
<td>1s</td>
</tr>
<tr>
<td>3</td>
<td>26s</td>
<td>64s</td>
<td>1s</td>
<td>3s</td>
</tr>
<tr>
<td>4</td>
<td>523s</td>
<td>635s</td>
<td>3s</td>
<td>10s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># nodes</th>
<th>snopt</th>
<th>filter</th>
<th>DEC-BPI</th>
<th>DEC-BPI corr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3s</td>
<td>2s</td>
<td>1s</td>
<td>2s</td>
</tr>
<tr>
<td>2</td>
<td>4s</td>
<td>5s</td>
<td>8s</td>
<td>31s</td>
</tr>
<tr>
<td>3</td>
<td>54s</td>
<td>110s</td>
<td>39s</td>
<td>151s</td>
</tr>
<tr>
<td>4</td>
<td>873s</td>
<td>2098s</td>
<td>118s</td>
<td>638s</td>
</tr>
</tbody>
</table>
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