RTBSS: An Online POMDP Algorithm for Complex Environments

Sébastien Paquet
Introduction

• Most POMDP algorithms are applicable only on small environments.

• However, most problems of interest have a huge state space,
  – which motivates the search for approximation methods.
RoboCupRescue

• This environment consists of a simulation of an earthquake happening in a city.

• The goal of the agents (representing firefighters, policemen and ambulance teams) is to minimize the damages, such as:
  - buried civilians,
  - buildings on fire,
  - blocked roads.
Motivations

• Our method has to be efficient in large state spaces.

• The agent’s response time has to respect a real-time constraint.

• Our method has to be efficient in previously unknown instances of an environment.
  - This eliminates all offline approaches, because the agent does not have time to learn a complete policy before its execution.

These constraints motivated us to develop an online POMDP algorithm that can ensure a quick response time in a huge state space.
Plan

• The RTBSS algorithm
• Hybrid approaches
• Experiments on standard POMDPs
• Adaptation to a multi-agent environment
  – Reduction of the search space
  – Dynamic reward function
• Discussion and Conclusion
Plan

• The RTBSS algorithm
• Hybrid approaches
• Experiments on standard POMDPs
• Adaptation to a multiagent environment
  − Reduction of the search space
  − Dynamic reward function
• Discussion and Conclusion
RTBSS

- RTBSS: Real-Time Belief State Search

- **Online POMDP algorithm**
  - Used to search online in the belief state tree.
  - It is a branch and bound style search.
The agent’s policy

• The action to perform is the action that maximizes the expected value of the current belief state:

\[
\pi(b, D) = \arg \max_{a \in A} \left[ \hat{R}_B(b, a) + \gamma \sum_{o \in \Omega} \left( \hat{P}_T(o \mid b, a) \times \delta(\tau(b, a, o), D - 1) \right) \right]
\]

• The value of a belief state is estimated by:

\[
\delta(b, d) = \begin{cases} 
U(b) & \text{if } d = 0 \\
\max_{a \in A} \left[ \hat{R}_B(b, a) + \gamma \sum_{o \in \Omega} \left( \hat{P}_T(o \mid b, a) \times \delta(\tau(b, a, o), d - 1) \right) \right] & \text{if } d > 0
\end{cases}
\]
Branch and bound style strategy

• The tree is developed using a depth limited search.
• The most promising actions are tried first.
• An upper bound is calculated for each action.
• If the action cannot improve the current reward, it is not explored.
RTBSS Algorithm

1: Function RTBSS(b, d) returns the estimated value of b.
   
   Inputs: b: The current belief state.
   d: The current depth.
   
   Statics: D: The maximal search depth.
   action: The best action.

2: if d = 0 then
3:   return U(b)
4: end if
5: actionList ← SORT(b, A)
6: max ← −∞
7: for all a ∈ actionList do
8:   curReward ← R_B(b, a)
9:   uBound ← curReward + HEURISTIC(b, a, d)
10: if uBound > max then
11:   for all o ∈ Ω do
12:     curReward ← curReward + γPr(o|a,b)RTBSS(τ(b,a,o), d − 1)
13:   end for
14: if curReward > max then
15:   max ← curReward
16: if (d = D) then
17:   action ← a
18: end if
19: end if
20: end if
21: end for
22: return max
POMDP Example

States: \( \{s_1, s_2\} \)
Actions: \( \{a_1, a_2\} \)
Observations: \( \{o_1, o_2\} \)
Discount factor: \( \gamma = 0.9 \)

<table>
<thead>
<tr>
<th>Reward Function</th>
<th>Transition Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R(s_1, a_1) = 2 )</td>
<td>( T(s_1, a_1, s_1) = 0.3 )</td>
</tr>
<tr>
<td>( R(s_1, a_2) = 1 )</td>
<td>( T(s_1, a_1, s_2) = 0.7 )</td>
</tr>
<tr>
<td>( R(s_2, a_1) = 1 )</td>
<td>( T(s_1, a_2, s_1) = 0.1 )</td>
</tr>
<tr>
<td>( R(s_2, a_2) = 3 )</td>
<td>( T(s_1, a_2, s_2) = 0.9 )</td>
</tr>
<tr>
<td>( T(s_2, a_1, s_1) = 0.6 )</td>
<td>( T(s_2, a_1, s_2) = 0.4 )</td>
</tr>
<tr>
<td>( T(s_2, a_2, s_1) = 0.8 )</td>
<td>( T(s_2, a_2, s_2) = 0.2 )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Observation Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>( O(s_1, a_1, o_1) = 0.9 )</td>
</tr>
<tr>
<td>( O(s_1, a_1, o_2) = 0.1 )</td>
</tr>
<tr>
<td>( O(s_1, a_2, o_1) = 0.9 )</td>
</tr>
<tr>
<td>( O(s_1, a_2, o_2) = 0.1 )</td>
</tr>
<tr>
<td>( O(s_2, a_1, o_1) = 0.5 )</td>
</tr>
<tr>
<td>( O(s_2, a_1, o_2) = 0.5 )</td>
</tr>
<tr>
<td>( O(s_2, a_2, o_1) = 0.5 )</td>
</tr>
<tr>
<td>( O(s_2, a_2, o_2) = 0.5 )</td>
</tr>
</tbody>
</table>
RTBSS Example

- Maximal depth = 2
- $b_0 = [0.1, 0.9]$
- Action’s sort function:
  - return $a_1$ if $s_1$ is the most probable state
  - return $a_2$ if $s_2$ is the most probable state
- $\text{Heuristic}(d) = \sum_{i=1}^{d} \gamma^i R_{max}$
- $U(b) = \max_{a \in A} R_B(b, a)$
RTBSS Example

\[ d = 2 \]

\[ d = 1 \]

\[ d = 0 \]
RTBSS Example

\[ d = 2 \]

\[ d = 1 \]

\[ uBound = R_B([0.83, 0.17], a_2) + \text{Heuristic}(1) \]
\[ = 4.04 \]

\[ d = 0 \]

\[ [0.49, 0.51] \quad [0.10, 0.90] \]

\[ (3.90) a_1 \]

\[ (0.79) o_1 \]

\[ (0.64) o_1 \]

\[ o_2 (0.36) \]
RTBSS Example

\[ d = 2 \]

\[ d = 1 \]

\[ d = 0 \]

\[ [0.49, 0.51] \]

\[ [0.10, 0.90] \]

\[ [0.34, 0.66] \]

\[ [0.05, 0.95] \]
RTBSS Example

\[ d = 2 \]

\[ d = 1 \]

\[ d = 0 \]

\[ u\text{Bound} = R_B([0.35, 0.65], a_1) + \text{Heuristic}(1) = 4.05 \]
RTBSS Example

\[ u_{Bound} = R_B([0.10, 0.90], a_1) + \text{Heuristic}(2) \]

\[ = 6.23 \]

\[ d = 2 \]

\[ (6.34) \quad [0.10, 0.90] \]

\[ (6.34) \quad a_2 \]

\[ o_2 (0.21) \]

\[ d = 1 \]

\[ (3.90) \quad [0.83, 0.17] \]

\[ (3.90) \quad a_1 \]

\[ (3.90) \quad o_1 \quad (0.64) \]

\[ o_2 (0.36) \]

\[ (5.9) \quad a_1 \]

\[ o_2 (0.41) \]

\[ (0.72) \quad o_1 \]

\[ o_2 (0.28) \]

\[ d = 0 \]

\[ [0.49, 0.51] \]

\[ 2.02 \]

\[ [0.10, 0.90] \]

\[ 2.8 \]

\[ [0.34, 0.66] \]

\[ 2.32 \]

\[ [0.05, 0.95] \]

\[ 2.9 \]

\[ [0.69, 0.31] \]

\[ 1.69 \]

\[ [0.20, 0.80] \]

\[ 2.6 \]
Plan

• The RTBSS algorithm
• Hybrid approaches
• Experiments on standard POMDPs
• Adaptation to a multiagent environment
  – Reduction of the search space
  – Dynamic reward function
• Discussion and Conclusion
Hybrid Approaches

• Mix of the RTBSS algorithm with offline algorithms.
  - Improve the online algorithm with precomputed expected rewards.
  - Improve the offline algorithms with an online search.

• Hybrid algorithms
  - RTBSS-QMDP
    (based on QMDP [Littman et al., 1995])
  - RTBSS-PBVI-QMDP
    (based on PBVI [Pineau et al., 2003] and QMDP [Littman et al., 1995])
  - RTDPBSS
    (based on RTDP-BEL [Geffner and Bonet, 1998])
Plan

• The RTBSS algorithm
• Hybrid approaches
• **Experiments on standard POMDPs**
• Adaptation to a multiagent environment
  – Reduction of the search space
  – Dynamic reward function
• Discussion and Conclusion
Experiments

• Tag problem (870 states)

[Pineau et al., 2003]
## Results Tag

<table>
<thead>
<tr>
<th>Method</th>
<th>Reward</th>
<th>Offline Time (s)</th>
<th>Online Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_{MDP}$</td>
<td>-16.75</td>
<td>0.875</td>
<td>-</td>
</tr>
<tr>
<td>RTDP-BEL</td>
<td>-12.15</td>
<td>3645</td>
<td>0.001</td>
</tr>
<tr>
<td>RTDPBSS</td>
<td>-9.60</td>
<td>24540</td>
<td>0.556</td>
</tr>
<tr>
<td>PBVI Pineau et al. (2003)</td>
<td>-9.18</td>
<td>180880</td>
<td>-</td>
</tr>
<tr>
<td>BBSLS Braziunas and Boutilier (2004)</td>
<td>$\approx$ -8.3</td>
<td>$\approx$ 100000</td>
<td>-</td>
</tr>
<tr>
<td>BPI Poupart (2005)</td>
<td>-6.65</td>
<td>250</td>
<td>-</td>
</tr>
<tr>
<td>HSVI1 Smith and Simmons (2004)</td>
<td>-6.37</td>
<td>10113</td>
<td>-</td>
</tr>
<tr>
<td>HSVI2 Smith and Simmons (2005)</td>
<td>-6.36</td>
<td>24</td>
<td>-</td>
</tr>
<tr>
<td>PERSEUS Spaan and Vlassis (2004)</td>
<td>-6.17</td>
<td>1670</td>
<td>-</td>
</tr>
<tr>
<td>PBVI Pineau (2004)</td>
<td>$\approx$ -6.12</td>
<td>$\approx$ 900000</td>
<td>-</td>
</tr>
<tr>
<td>RTBSS-QMDP(5)</td>
<td>-6.11</td>
<td>0.875</td>
<td>0.311</td>
</tr>
<tr>
<td>RTBSS-PBVI-QMDP(4)</td>
<td>-5.40*</td>
<td>360000</td>
<td>0.220</td>
</tr>
<tr>
<td>RTBSS(12)</td>
<td>-5.03</td>
<td>0</td>
<td>0.750</td>
</tr>
</tbody>
</table>
Experiments

• RockSample Problem

[Smith and Simmons, 2004]
## Results RockSample

<table>
<thead>
<tr>
<th></th>
<th>RTDP-BEL</th>
<th>$Q_{MDP}$</th>
<th>Perseus</th>
<th>RTBSS(6)</th>
<th>HSVI (Smith &amp; Simmons, 2004)</th>
<th>PBVI</th>
<th>RTBSS-PBVI-QMDP(2)</th>
<th>RTBSS-QMDP(2)</th>
<th>RTDPBSS(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RockSample[5,5], (801s,10a,2o)</td>
<td>7.3</td>
<td>1444</td>
<td>0.003</td>
<td>18.5</td>
<td>0</td>
<td>19.0</td>
<td>10208</td>
<td>0.131</td>
<td>19.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>19.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>19.4</td>
</tr>
<tr>
<td>RockSample[5,7], (3201s,12a,2o)</td>
<td>12.5</td>
<td>36000</td>
<td>0.001</td>
<td>17.3</td>
<td>4</td>
<td>18.9</td>
<td>36000</td>
<td>0.015</td>
<td>20.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>22.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RockSample[7,8], (12545s,13a,2o)</td>
<td>4.3</td>
<td>36000</td>
<td>0</td>
<td>8.3</td>
<td>36000</td>
<td>8.7</td>
<td>8332</td>
<td>0.001</td>
<td>13.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>17.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>19.0</td>
</tr>
<tr>
<td>RockSample[10,10], (102401s,19a,2o)</td>
<td>11.2</td>
<td>208</td>
<td>0.029</td>
<td>19.2</td>
<td>208</td>
<td>20.0</td>
<td>0</td>
<td>1.234</td>
<td>20.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>20.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>20.4</td>
</tr>
</tbody>
</table>

© Sébastien Paquet – November 2005

[联络信息] www.damas.ift.ulaval.ca Multi-AgentS
Plan

• The RTBSS algorithm
• Hybrid approaches
• Experiments on standard POMDPs
• Adaptation to a multiagent environment
  – Reduction of the search space
  – Dynamic reward function
• Discussion and Conclusion
Policemen

• Their task is to clear the most important roads.
  – To help the other agents navigate in the city.

• They have to be coordinated.
  – So that they do not interfere with one another.

• This is a complex task in a partially observable, real-time and dynamic environment.
Application to the RoboCupRescue

• Modelling from the police point of view
  – Actions: 4 moving actions and clear
  – States: Approximately 1500 variables
    • 800 roads blocked or not
    • 700 buildings on fire or not
    • 30-40 agents which can be on one of the 800 roads
    • Total number of states: $2^{800} \times 2^{700} \times 800^{40}$

• However, many states are impossible or will never be visited.
Application to RoboCupRescue

• The online search in the belief state space represents a search in the possible paths.

• We have to return a path, thus we return the best branch of the tree instead of only the first action.
  - RTBSS can return action sequences of different lengths.
Reduction of the search space

• Many variables are considered static during the search.
  – In fact, we consider only the agent’s position and the road’s state.

• For the other variables, there is no transition function, thus we rely only on the agent’s perceptions (direct and by messages).
Reduction of the search space

• The fixed variables are considered during the search, but they keep the last value perceived.

• In dynamic multiagent environments, it is hard to predict everything, because there are too many things changing.

• It could better to rely on perceptions for less important variables and concentrate on the most important variables.
Dynamic Reward Function

- The reward function is updated after each new perception.
- Can be used to coordinate the police agents.
Results

• On average RTBSS obtains 11 points more than the heuristic approach.
Results

• With RTBSS, roads are cleared faster.
• At the beginning there are one or two less agents blocked.
Plan

• The RTBSS algorithm
• Hybrid approaches
• Experiments on standard POMDPs
• Adaptation to a multiagent environment
  − Reduction of the search space
  − Dynamic reward function
• Discussion and Conclusion
Advantages

• Need no offline computations
  – Agents are efficient even in new environment configurations.

• Can manage large state spaces.

• Applicable in real-time environments.

• Since it is online, we can:
  – Simplify the belief update function
  – Use a dynamic reward function

  • Useful for dynamic coordination of many agents
Conclusion

• We introduced RTBSS, an online POMDP algorithm useful for large, dynamic and uncertain environments.

• RTBSS obtained good results on standard POMDP problems.

• RTBSS is suited for environments
  − in which the initial configurations can change and
  − when the agents have to be deployed rapidly.

© Sébastien Paquet – November 2005
Conclusion

• Hybrid approaches can be a good compromise between completely offline and online approaches.

• We showed how RTBSS can be used in a multiagent environment. We presented:
  – An approach to manage the dynamic parts of the world.
  – A coordination method based on a dynamic reward function.
Thank You!

• For more information:
  - http://www.damas.ift.ulaval.ca/~spaquet
  - spaquet@damas.ift.ulaval.ca