**Collaboration in Artificial Intelligence**

**Course Project**

**Title: Modeling MA - Path Finding as MDP**

**Author: Daniel Portnoy**

**Submission Date: 15.07.2018**

**Project Goals**

Model a multi agent path finding settings in a form of an MDP. Define:

* 1. Joint States.
  2. Joint Actions.
  3. Transition Function.
  4. Start/Terminal state.
  5. Reward function.

Test different MDP solvers and Reinforcement learning techniques on those settings:

* 1. Value Iteration.
  2. RTDP.
  3. Q-Learning

**Tools**

I used **BURLAP** (http://burlap.cs.brown.edu/) as an infrastructure for the project.

BURLAP is a java code library for using and developing single or multi-agent planning and learning algorithms and domains to accompany them.

**Domain and MDP Model**

The Domain i choose is the (n\*n) Grid Domain, so that:

1. All agents have X,Y coordinates.
2. All agents can move to a direction (North,South,East,West), or waiting.
3. The grid has obstacles (walls) that the agent cant pass through.
4. Agents cant collide or swap locations.

The MDP model is:

1. The Joint State is all agents coordinates.
2. The Joint Actions is all agents actions.
3. The Transition Function is independent between agents:
   1. moving in each direction has 80% chances of success.
   2. waiting (staying on current tile) has 100% chance of success.
4. The Reward is defined to be -1 for every state, and 100 for terminal state.

**Maps used**

I have created 2 maps for testing the solving algorithms:

1. 10X10 grid with “4 rooms” and 2 agents:
2. 14X14 grid with “3 walls” and 2 agents:

**Test Results**

**Value Iteration**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Map | Max Iterations | Delta | Steps | Time (sec) |
| 1 | 10 | 10^-9 | 33 | 134 |
| 1 | 20 | 10^-9 | 31 | 286 |
| 2 | 10 | 10^-9 | 1302 | 443 |
| 2 | 20 | 10^-9 | 62 | 816 |

**RTDP**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Map | Max Rollouts | Max Depth | Delta | Steps | Time (sec) |
| 1 | 10000 | 1000 | 10^-9 | 16828 | 24.25 |
| 1 | 100000 | 1000 | 10^-9 | 15266 | 28.82 |
| 1 | 1000000 | 1000 | 10^-9 | 13656 | 23.49 |
| 1 | 10000000 | 1000 | 10^-9 | 9732 | 30.98 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2 | 10000 | 1000 | 10^-9 | 26758 | 24.37 |
| 2 | 100000 | 1000 | 10^-9 | 24894 | 23.93 |
| 2 | 1000000 | 1000 | 10^-9 | 21948 | 22.85 |
| 2 | 10000000 | 1000 | 10^-9 | 19871 | 30.93 |

**Q Learning**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Map | Episodes | Learning Rate | Epsilon | Steps | Time (sec) |
| 1 | 10000 | 0.5 | 0.1 | 63 | 10.28 |
| 1 | 100000 | 0.5 | 0.1 | 54 | 35.56 |
| 1 | 1000000 | 0.5 | 0.1 | 48 | 595 |
| 2 | 20000 | 0.5 | 0.1 | 128 | 76.83 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2 | 100000 | 0.5 | 0.1 | 98 | 122 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2 | 200000 | 0.5 | 0.1 | 85 | 201 |

**Summery**

The Modeling

The entire project is available on Github (https://github.com/portdan/Course-Project)