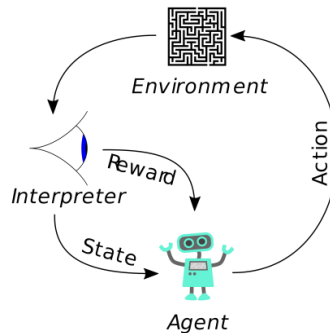


COSC 4368 Project Spring 2022  
*Using Reinforcement Learning*  
*To Discover Paths in a 2-Agent Transportation World*  
Group Project (3-4 Students per Group)  
Fourth Draft



Last updated: April 6, 1p

Deadlines: Submit project report and source code by Tu., April 19 end of the day<sup>1</sup>

In this project we will use reinforcement to learn and adapt “promising paths” in 2-agent setting. Learning objectives of the 2022 COSC 4368 Group Project include:

- Understanding basic reinforcement learning concepts such as utilities, policies, learning rates, discount rates and their interactions.
- Obtain experience in designing agent-based systems that explore and learn in initially unknown environment and which are capable to adapt to changes.
- Learning how to conduct experiments that evaluate the performance of reinforcement learning systems and learning to interpret such results.
- Development of visualization techniques summarizing how the agents move, how the world and the q-table changes, and the system performance.
- Development of path visualization and analysis techniques to interpret and evaluate the behavior of agent-based path-learning systems.
- Develop and learn coordination strategies for collaborating agents
- Learning to develop AI software in a team.

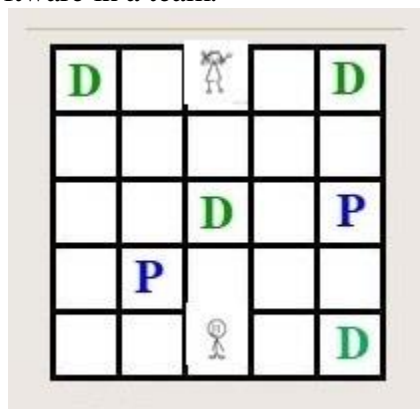


Figure 1: Visualization of the PD-World

<sup>1</sup> There will **no** further extensions of the group project submission deadline!



Figure 2: An Urban Grid World.

In particular in this project you will use *Q-learning/SARSA*<sup>2</sup> for the PD-World assuming a 2 agent setting (<http://www2.cs.uh.edu/~ceick/ai/2022-World.pptx> ), conducting four experiments using different parameters and policies, and summarize and interpret the experimental results. Moreover, you will develop path visualization techniques that are capable to shed light on what paths the learning system actually has learnt from obtained Q-Tables—we call such paths *attractive paths* in the remainder of this document. Moreover, you will analyze if the two agents collaborated well by avoiding blockage that occurs if the two agents work on the same path.

Two agent named ‘M’ (male) and ‘F’ (female) are solving the block transportation problem jointly. Agent alternate applying operators to the PD-World, with the female agent acting first. Moreover, both agents cannot be in the same position at the same time; consequently, there is a blockage problem, limiting agent mobility and ultimately efficiency in case that both agents work on the same path at the same time. There are two approaches to choose from to implement 2-agent reinforcement learning.

- a. Each agent uses his/her own reinforcement learning strategy and Q-Table. However, we assume that the position the other agent occupies is visible to each agent, and can therefore can be part of the chosen reinforcement learning state space.
- b. A single reinforcement learning strategy and Q-Table is used which moves both agents, selecting an operator for each agent and then executing the selected two operators.

Extra credit is given to groups who devise and implement both 2-agent learning approaches and compare their results for experiments 2 and 3 (see below)

In experiments we assume that q values are initialized with 0 at the beginning of the experiment. The following 3 policies will be used in the experiments:

- **PRANDOM**: If pickup and dropoff is applicable, choose this operator; otherwise, choose an applicable operator randomly.
- **PEXPLOIT**: If pickup and dropoff is applicable, choose this operator; otherwise, apply the applicable operator with the highest q-value (break ties by rolling a dice for

---

<sup>2</sup> SARSA is a variation of Q-learning that uses the q-value of the actually chosen action and not the q-value of the best action!

operators with the same q-value) with probability 0.8 and choose a different applicable operator randomly with probability 0.2.

- **PGREEDY**: If pickup and dropoff is applicable, choose this operator; otherwise, apply the applicable operator with the highest q-value (break ties by rolling a dice for operators with the same q-value).

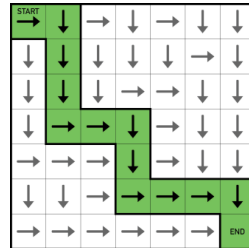


Figure 3: Visualization of an Attractive Path for a Search Problem

Objectives of the Experimental Evaluation: Besides analyzing the performance of various variations with respect to the bank account/how quickly the transportation problem was solved, the experimental evaluation should also additionally analyze:

- a. Agent coordination: Do the two agents get in their ways blocking each other or do they do a good job in dividing the transportation task intelligently between each other. Agent coordination could, for example, be measured by computing the average Manhattan distance between the two agents during the run of a specific experiment.
- b. Paths learned: Does the particular approach do a good job in learning paths between block sources and block destinations; is the learnt path the shortest path or close to the shortest path between the source and the destination.

Please conduct the following four experiments:

1. In Experiment 1 you use  $\alpha=0.3$  and  $\gamma=0.5$ , and run the traditional Q-learning algorithm for 8000 steps; initially you run the policy PRANDOM for 500 steps, then
  - a. Continue running PRANDOM for 7500 more steps<sup>3</sup>
  - b. Run PGREEDY for the remaining 7500 steps
  - c. Run PEXPLOIT for the remaining 7500 steps

Summarize and interpret the different results you obtain by running these three strategies. Also report one of the final Q-Table of experiment 1.c. Also assess the quality of the coordination between the two agents for experiments 1b and 1c.

2. Experiment 2 is the same as experiment 1.c except you run the SARSA q-learning variation for 8000 steps. When analyzing Experiment 2 center on comparing the performance of Q-learning and SARSA. Also report one of the final Q-tables of this experiment. Also assess the quality of agent coordination,
3. In Experiment 3 you rerun either<sup>4</sup> Experiment 1.c or 2 but with learning rates  $\alpha=0.15$  and  $\alpha=0.45$ . When interpreting the results focus on analyzing the effects of using the 3 different learning rates on the system performance.
4. Experiment 4 is the somewhat similar to Experiment 1c or 2; you use  $\alpha=0.3$  and  $\gamma=0.5$

<sup>3</sup> A step is the application of an operator by one of the two agents.

<sup>4</sup> You have a choice here!

in conjunction with either Q-learning or SARSA<sup>5</sup> as follows: you run PRANDOM for the first 500 steps; next, you run PEXPLOIT; however, after a terminal state is reached the third time, change the two pickup locations to: (1,2) and (4,5); the drop off locations and the Q-table remain unchanged; finally, you continue running PEXPLOIT with the “new” pickup locations until the agent reaches a terminal state the sixth time. When interpreting the results of this experiment center on analyzing on how well the learning strategy was able to adapt to the change of the pickup locations and to which extend it was able to learn “new” paths and unlearn “old” paths which became obsolete.

For all experiments, if a terminal state is reached, restart the experiment by resetting the PD world to the initial state, but do not reset the Q-table. Run each experiment twice, and report<sup>6</sup> and interpret the results; e.g., utilities computed, rewards obtained in various stages of each experiment.

Assess which experiment obtained the best results<sup>7</sup>. Next, analyze the various q-tables you created and try identify attractive paths<sup>8</sup> in the obtained q-tables, if there are any. Moreover, briefly assess if your system gets better after it solved a few PD-world problems—reached the terminal state at least once. Briefly analyze to which extend the results of the two different runs agree and disagree in the 4 experiments. Analyze agent coordination for experiments 1.c and 4. Finally, analyze how well the approach adapted to change in the fourth experiment.

Moreover,

- Make sure that you use different random generator seeds in different runs of the same experiment to obtain different results—having identical results for the 2 runs of the same experiment is unacceptable. It is okay just to report and interpret the Q-tables for the better of the two runs for each experiment, but you should report the performance variables for all eight runs.
- You should use the traditional Q-learning and SARSA algorithm in the project and not any other Q-learning variations or reinforcement learning algorithms!
- Never update the q-values of operators are not applicable in a Q-Tables!
- Groups who develop good methods for visualizing q-tables and good visualizations for the analysis of attractive paths obtain extra credit.
- Groups who develop sophisticated methods to analyze agent coordination receive a small amount of extra credit.

---

<sup>5</sup> Again you have a choice here.

<sup>6</sup> Additionally, report the following Q-tables for Experiments 2 (or Experiment 3 if you prefer that, in this case you will only need to report the final Q-Table of Experiment 2) in your report a) when the first drop-off location is filled (the fifth block has been delivered to it) and b) when a terminal state is reached and c) the final Q-table of each experiment.

The Q-table in the screenshot should be presented as a matrix, with  $s$  rows (states) and  $t$  columns (operators). Thus, the Q-table for State Space 0, in the World 2022 pptx slides, has 25 x 2 rows and 6 columns; however, the q-values for the drop-off and pickup operators do not need to be reported.

<sup>7</sup> Provide graphs that show, how the algorithm’s performance variables changed over the duration of the experiment in the three experiments.

<sup>8</sup> A path going from  $(i,j)$  to  $(i',j')$  is attractive if, the q-values of the motion operators are high in comparison to other directions. Be aware of the fact that different paths are attractive for agents who hold a block and for agents how do not hold a block.

- Allow in your software design that the positions of dropoff and pickup positions might be changing before and during a run; otherwise, you will need to write a lot of additional software for experiment4.
- Evidence of the running of your system has to be provided using screen shots that will be delivered in a separate document.
- Groups that develop a very well designed and visually appealing visualization component will receive extra credit for this part of the 4368 Group Project!

Write a 8-12 pages report that summarizes the findings of the project. Be aware of the fact that at least 15-20% of the points available for this project are allocated to the interpretation of the experimental results. Finally, submit the source code of the software you wrote in addition to your project report and be ready to demo the system you developed.

## Project Links

<http://courses.cs.washington.edu/courses/cse473/15sp/assignments/project3/project3.html>  
[http://ai.berkeley.edu/project\\_overview.html](http://ai.berkeley.edu/project_overview.html)  
<https://github.com/kristofvanmoffaert/Gridworld>  
[http://cs.stanford.edu/people/karpathy/reinforcejs/gridworld\\_td.html](http://cs.stanford.edu/people/karpathy/reinforcejs/gridworld_td.html)  
<https://mediatum.ub.tum.de/doc/1238753/1238753.pdf>  
<http://www2.econ.iastate.edu/tesfatsi/RLUsersGuide.ICAC2005.pdf>  
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.113.7978&rep=rep1&type=pdf>  
[1911.10635.pdf \(arxiv.org\)](http://arxiv.org/abs/1911.10635)