Explore Agent Behavior Under Repeated Congestion Games

Cheng Luo, Tyree Chrisp, Mason Hall

1 INTRODUCTION

In this report, we explore the Braess Paradox in congestion games in the context of repeated games where agents are allowed to play a congestion game repeatedly, learn, and adjust their path choice based on either their own past experience or other agents' past actions. In particular, we model the agents' behavior using three different learning algorithms, explore whether agents repeatedly playing the game converge to a equilibrium, and measure how various factors can affect agents' average path cost. In addition, we discuss how agents using different learning algorithms interact with one another.

2 IMPLETMENTATION

2.1 Road Network Setup

The congestion game is configured based on the classical Braess Paradox setting (see **Figure 1**): two roads with a fixed cost of 45, two roads with variable cost of T/1 (T is the number of agents passing through that road), and potentially a superhighway with 0 cost to be built between vertex A and B. A total of 40 agents are expected to travel from Start vertex to End vertex, via a path that contains a subset of the available roads.

In our implementation, we simplify the game into a two/three-choice problem, where the agents are proposed each round to make a choice between paths: Start-A-End, Start-B-End, and Start-A-Superhighway-B-End (depending on if superhighway is built), based on their individual learning algorithms.

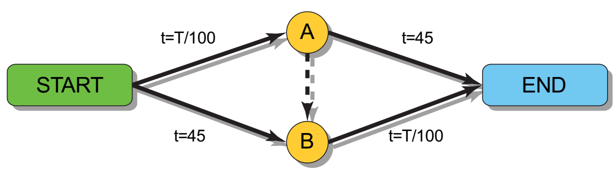

Figure 1

2.2 File Infrastructure

We implemented a program in Java to simulate the road network. A Road class is written with functions to calculate road costs based on number of agents on the road. A Path class consists of a set of Road objects, and can calculate path costs accordingly. An Agent class is written with data structures to record agents' past choices, rewards, and belief distributions. The Agent class also contains functions to make a path choice based on different learning algorithms, and calculate average cost after certain number of simulations. Lastly, a CongestionGame class is used to serve as the main game simulator, in which it configures the number of rounds, the number of each type of agents, path cost, with/without superhighway options, and record relevant data.

2.3 Agent Types

We implement three types of agents using three learning algorithms: fictitious play, ε-greedy, and UCB-1. Note that when in the first round, where there is no historical data on each path, we make all agents choose randomly between the available paths. For the second and third round, the agents are encouraged to choose an unexplored path to get historical data on that particular path.

2.3.1 Fictitious Agent

Fictitious agents presume that the opponents' strategies are stationary, and act according to empirical frequency of opponents' actions. In our implementation, fictitious agents keep track of other agents' path choices, and yield an empirical distribution on other agents' path choices through normalization. When it's their turn to make a path choice, they predict number of agents on each path based on the empirical distribution, calculate the expected cost accordingly, and choose the path with the lowest expected cost.

2.3.2 ε-greedy Agent

Epsilon greedy agents keep track of their own path choice, and corresponding cost for each round. They choose a path randomly with a probability of **ε**. For the remaining 1- **ε** time, they calculate the average cost for each path based on only own past experience, and choose the path with the maximum average cost.

2.3.3 UCB-1 Agent

UCB-1 agents also keep track of their own path choice and corresponding cost for each round. When making a path choice, they calculate the value of $\overline{x_j} + \sqrt{\frac{2\ln(n)}{n_j}}$, where n is the total number of rounds, n_j is the number of rounds that they choose path j, and $\overline{x_j}$ is the average reward for path j based on their own past experience. They choose the path that maximizes the above value.

3 RESULTS AND DISCUSSION

For simplicity, path Start-A-End is referred as path 1, path Start-A-Superhighway-B-End is referred as path 2, and path Start-B-End is referred as path 3.

3.1 Equilibrium Convergence

Of the three types of agents, only the UCB-1 agents can yield a stable equilibrium.

3.1.1 Fictitious Agents Equilibrium

As seen in **Appendix A,** in the first round, fictitious agents first choose randomly between the three paths because there is not historical data to reason from. In the second round, all agents will find path 2 attractive since they predict that only about 1/3 of agents would travel through that path. However, in the third round, they would predict that exactly 1/2 of agents would travel through path 2, which makes the other two paths attractive with lower expected costs. In round 4, they would again predict that less than 1/2 of agents would travel through path 2, making path 2 attractive again. The process continues on based on tiny change of the agents' distribution belief. Thus, the fictitious agents equilibrium is alternating between the three paths.

3.1.2 ε-greedy Agents Equilibrium

ε-greedy agents cannot yield an absolute equilibrium due to their randomness. However, as seen in **Appendix B,** they generally condense on the path with the superhighway, and occasionally spread to path 1 & 3 due to its random factor.

3.1.3 UCB-1 Agents Equilibrium

UCB-1 agents are the only ones that can yield a stable equilibrium. This is because initially when there is no historical data, they will try each path to get a specific cost. After enough rounds, they should consistently choose the one path yielding the lowest expected value of their evaluation function.

3.2 Simulation Time's Impact on Average Cost

With this simulation, we want to explore which type(s) of agents can better leverage on their learning algorithm, and lower average path costs over time (measured in number of rounds). The game is run between 10 to 500 rounds to show this behavior. Each game with a particular round number is run 100 times to exclude possible run-time environment effects, and better predict result on an average basis. The results are as follows in **Figure 2**:

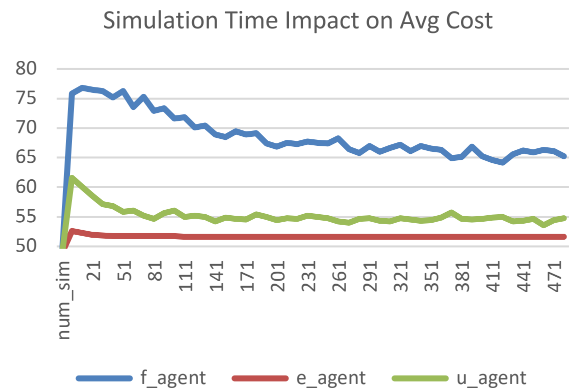

Figure 2

For the first three rounds, all three types get low average costs because agents are spread out either choosing randomly or exploring unexplored options when there are not enough data for each path. Later, we observe that fictitious agents can better leverage their past data, with the largest decrease in average cost over time, possibly due to the fact that they make their decisions based on others' past choices. With more simulations, they know each other's choice pattern better, thus making the whole system more efficient. ε-greedy agents are able to leverage on their randomness to explore paths to maintain their relative low cost from the start. The UCB-1 agents exhibit a near flat line with occasional fluctuation due to the facts that they would continue choose the same path after reaching equilibrium, and that they are more encouraged to choose path choices that they were not choosing as frequently before as n gets larger.

3.3 Learning Algorithm's Impact on Average Cost

With the graph in section 3.2, we observe that overall, fictitious agents have the highest average cost and ε-greedy agents have the lowest average costs. Fictitious agents suffer from their back and forth equilibrium pattern, thus cannot predict next round's agent distribution well enough. This leads to an always overcrowded path, thus increasing the average cost. Both the ε-greedy agents and the UCB-1 agents have reasonably good average cost. UCB-1 agents benefit from their always cost minimizing behavior, while ε-greedy agents benefit from their randomness.

3.4 With or Without Superhighway

With this simulation, we want to compare the average cost for each type of agents both with or without the superhighway. We run each simulation 500 times to get average results. The result is below in **Figure 3**:

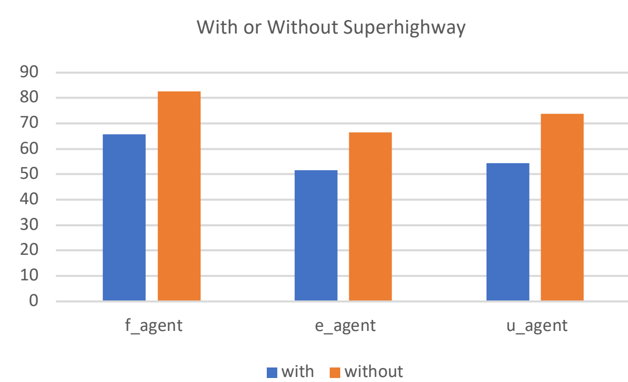

Figure 3

Overall, we observe that all types of agents can lower their average costs with the superhighway. Fictitious agents' back and forth equilibrium pattern is relieved with the superhighway, since before they move between only two available paths, making it crowded nearly all the time. With three paths, the middle path has lower overall cost for 40 agents. When they do choose the middle path, they yield a lower average cost occasionally. Both the ε-greedy agents and UCB-1 agents can benefit from the superhighway, mainly because they make initial choice randomly and later choice based on past average cost. With one more option, it's less likely that one path becomes extremely crowded, yielding lower average cost overall.

3.5 Interaction Between Different Types of Agents

We wish to see how different types of agent interact with each other. In the following settings, we use 42 agents with 1/3 of each type for 300 simulations and compare their average cost with the situations that there is only one type of agents. The result is below in **Figure 4**.

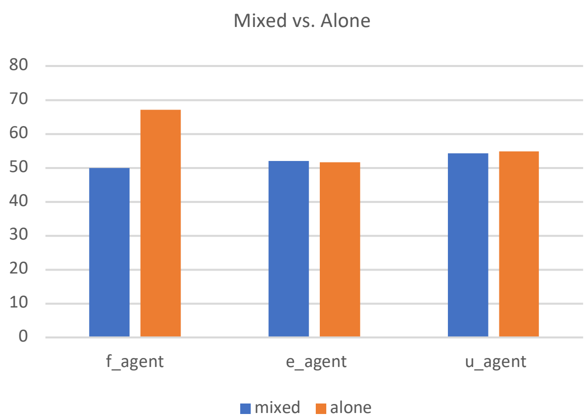

Figure 5

Figure 4

With mixed types of agents, only the fictitious agents are able to leverage on other agents' behavior, which makes sense since they are the only type that make path choice based on others' past actions. As a result, they can best respond to other agents' strategy. The ε-greedy agents and UCB-1 agents have similar costs to the situation where they are the only type of agents, due to the fact that they make decision only based on their own past experience.

3.6 How Greedy is Good

In this setting, we want to explore how the value of ε can affect the average cost of ε-greedy agents. We run 300 simulations with ε values ranging from 0 to 1. The result is as above in **Figure 5**. With increasing randomness, the ε-greedy agents are more likely to spread out to all three paths, rather than condensing on path 2. This makes their average cost generally decrease with increasing value of ε.

4 CONCLUSIONS

In this report, we explored three types of agent learning algorithm in the context of repeated congestion games. Overall, fictitious agents can best respond to and benefit from other agents' actions. However, ε-greedy and UCB-1 agents benefit from their randomness and cost minimizing behavior, which yields to lower average costs.

REFERENCES

[1] Johari, R. (2018, May 5). *Lecture 6: Fictitious Play*. Reading presented at MS&E336, Stanford, CA. http://web.stanford.edu/~rjohari/teaching/notes/336_lecture6_2007.pdf
[2] Lattimore, T. (2018, February 10). The Upper Confidence Bound Algorithm. Retrieved from http://banditalgs.com/2016/09/18/the-upper-confidence-bound-algorithm/
[3] McCaffrey, J. D. (2017, December 03). The Epsilon-Greedy Algorithm. Retrieved May 5, 2018, from https://jamesmccaffrey.wordpress.com/2017/11/30/the-epsilon-greedy-algorithm/
[4] Welsh, N. (2010, November 9). *Bandit Algorithms Continued: UCB1*. Reading presented in University of Birmingham, B15 2TT, United Kingdom. https://www.cs.bham.ac.uk/internal/courses/robotics/lectures/ucb1.pdf

APPENDIX A & B

A sample of first 20 rounds of 40 fictitious agents playing the congestion game is shown below (left).
A sample of first 20 rounds of 40 ε-greedy agents playing the congestion game is shown below (right).

Round	Path 1	Path 2	Path 3
1	16	18	6
2	0	40	0
3	0	0	40
4	0	40	0
5	7	33	0
6	40	0	0
7	0	40	0
8	0	40	0
9	0	0	40
10	0	40	0
11	0	40	0
12	40	0	0
13	0	40	0
14	0	40	0
15	0	0	40
16	0	40	0
17	0	40	0
18	40	0	0
19	0	40	0
20	0	40	0

Round	Path 1	Path 2	Path 3
1	13	13	14
2	19	10	11
3	11	16	13
4	7	27	6
5	5	17	18
6	16	13	11
7	6	30	4
8	15	13	12
9	11	19	10
10	5	31	4
11	12	10	18
12	11	17	12
13	13	24	3
14	4	25	11
15	9	29	2
16	12	13	15
17	13	18	9
18	3	31	6
19	20	12	8
20	1	20	19