SCML 2022 – OneShot Agent Report

Yosef Zumer and Michaela Levi Richter from Bar-Ilan University

Introduction

The OneShot game is intended to further research on agent negotiation. As such, the game design emphasizes negotiation and de-emphasizes operations. We chose to improve existing strategies and solve some of their drawbacks.

Strategy

Our strategy is to build a greedy learning agent that tries to maximize profits. Our agent “MyAgent” is inherited from the given agents in the tutorial – “SimpleAgent”, “BetterAgent” and “AdaptiveAgent”.

In addition, “MyAgent” saves any offer that he has received and the id of the agent that gave the offer. “MyAgent” determines the price range by the data that he collected previously. For each negotiator, “MyAgent” calculates the average unit price of the previous offers that came from the particular negotiator. Also, “MyAgent” calculates the general average unit price of all the previous offers. After that, “MyAgent” can determines the “type” of the negotiator that he currently dealing with, by examine the relation between the average unit price of that negotiator and the general average unit price of all the negotiators.

For example, if “MyAgent” is in selling position and he want to sell to X, he calculates the average unit price of X and the general average unit price. The smaller the average unit price of X in compare to the general average unit price, we say that X is greedier. Otherwise, we say that X is more generous.

For a greedy agent we want to give him an offer that he will accept so we will give him less aggressive offers. In case of generous agent, we want to take advantage of his goodwill and give him more aggressive offers.

“MyAgent” classifies each agent into one of four types: “GreedLevel1”, “GreedLevel2”, “GreedLevel3” and “GreedLevel4”. To determine the agent’s type, “MyAgent” checks the ratio (“R”) between the agent average unit price and the general average unit price:

- In selling position, if 0<=R<0.5, the type will be “GreedLevel4”, 0.5<=R<1 “GreedLevel3”, 1<=R<2 “GreedLevel2”, R>=2 “GreedLevel1”.
- In buying position, if 0<=R<0.5, the type will be “GreedLevel1”, 0.5<=R<1 “GreedLevel2”, 1<=R<2 “GreedLevel3”, R>=2 “GreedLevel4”.

To determine the price range per each type, “MyAgent” takes the price range that the “AdaptiveAgent” returns and improves it as follow:

- In selling position “MyAgent” updates the lower boundary to be: max((1 – w_type) * adaptive_lower_boundary + (w_type) * avg_i, avg_i)
- In buying position “MyAgent” updates the upper boundary to be: min((1 – w_type) * adaptive_upper_boundary + (w_type) * avg_i, avg_i)

Variables:

- “w_type”: the weight that determines by the negotiator type.
  - For “GreedLevel1”, “w_type” equal to -0.15.
  - For “GreedLevel2”, “w_type” equal to -0.075.
  - For “GreedLevel3”, “w_type” equal to 0.075.
  - For “GreedLevel4”, “w_type” equal to 0.15.
- “avg_i”: the average unit price of the current negotiator.
- “adaptive_lower_boundary”: the best selling offer that we have received during the current simulation.
- “adaptive_upper_boundary”: the best buying offer that we have received during the current simulation.

Conclusion

Our agent tries to learn the negotiators types and to tune the price range that the “AdaptiveAgent” gave us, by using the types that he has learned.

Link to git:

https://github.com/yosef199/SCML