

Epsilon Greedy Agent for ANAC2024 SCML Agent Strategy

Yuzuru Kitamura

April 17, 2024

Tokyo University of Agriculture and Technology

kitamura@katfuji.tuat.ac.jp

1. Introduction

In SCML 2024, as in the previous year, the impact of transaction prices on scores is much smaller than that of transaction volume. Therefore, a strategy was created to negotiate primarily on transaction volume. Since most of the transactions are based on the first proposal and its acceptance, the agent learns the acceptance rate of the first proposal. The epsilon-greedy method was used for learning, which is the origin of the agent's name. EpsilonGreedyAgent inherits class BetterSyncAgent.

2. Strategy

2.1. Proposal Strategy

Like the BetterSyncAgent, it distributes its current needs to all negotiators. However, unlike BetterSyncAgent, the distribution is not random. It selects one of the following two proposals.

[A] Even Proposal

Equally distribute the quantity demanded q among n counterparties. The quantity allocated to one counterparty will be $\lfloor \frac{q}{n} \rfloor$ or $(\lfloor \frac{q}{n} \rfloor + 1)$.

[B] Uneven Proposal

Keep track of the percentage (acceptance_rate) that the agent has accepted its initial proposal in previous transactions. Distribute $\min(1, q - n + 1)$ to my negotiator with the highest value, and 1 to the others.

The two proposals, [A] and [B] are randomly selected according to the variable $\epsilon(s)$, which depends on the number of steps (s).

$$\epsilon_s = \frac{s}{n_steps}$$

With probability ϵ , A is selected, and with probability $(1 - \epsilon)$, B is selected. In the early stages,

equal proposals are made to explore who is more likely to accept the proposal, and at the end of the process, the learned data is used. This is a strategy based on the epsilon-greedy method.

2.2. Response Strategy

Like BetterSyncAgent, the power set of offers from the counterparties is obtained and the combination that is closest to the quantity demanded is selected from among them. In this agent, the criteria for selecting combinations were changed.

In BetterSyncAgent, out of all the combinations, the one with the smallest difference between the needs and the total amount offered is selected and combined. On the other hand, EpsilonGreedyAgent selects the combination with the maximum amount of offers that does not exceed the needs. By responding "Reject" to the offerer not selected here, the offered amount can be made closer to the needs.

3. Evaluation

We tested EpsilonGreedyAgent in a simulations against *SyncRandomOneShotAgent*, *BetterSyncAgent*, and *RandomOneShotAgent*. The condition of the simulation are shown in Table 1.

Table1: The condition of simulations

Condition	Value
n_configs	10
n_steps	100

The results are shown in Table2. This result shows that EpsilonGreedyAgent is superior to all other agents, especially the original BetterSyngAnget.

Table2: The result of simulation

rank	Agent_type	Score
1	EpsilonGreedyAgent	1.081
2	SyncRandomOneshotAgent	1.052
3	BetterSyncAgent	1.047
4	RandomOneShotAgent	0.733

4. Reference

- [1] Y.Mohammad, “Developing an agent for SCML2024 (OneShot) – A better SyncAgent”.
https://scml.readthedocs.io/en/latest/tutorials/02.develop_agent_scml2024_oneshot.html (accessed April 17)