

DogAgent: An agent submitted to the ANAC 2024 SCM league

Izumi Hamano¹ Takanobu Otsuka²

^{1,2}Nagoya Institute of Technology, Aichi, Japan

¹hamano.izumi@otsukalab.nitech.ac.jp

²otsuka.takanobu@nitech.ac.jp

April 12, 2024

Abstract

We have developed an agent with the aim of minimizing inventory costs and shortfall penalties as much as possible. DogAgent makes optimal contracts based on inventory levels and the number of production lines. Our objective is not to make a large profit at one time, but to make a stable profit.

1 Introduction

The scm world simulates a supply chain consisting of multiple agents. The agents negotiate with each other, signing contracts and making transactions. Their objective is to make a profit, and the agent with the highest average profit across multiple simulations wins. However, achieving a positive profit in scml is difficult. In many cases, agents will have negative profits. This is because agents are penalized for not fulfilling many contracts. Therefore, we decided to aim for a reliable and stable profit instead of aiming for a large profit with many contracts. In addition, in a real supply chain, there are fixed costs. It is necessary to make a minimum number of transactions to make a profit. But on the scml standard track, agent is no need to pay such costs. In other words, in the scml world, you do not sign contracts that are disadvantageous to you, only contracts that are advantageous to you. With this in mind, we have created an agent that only enters into optimal contracts that minimize the cost of inventory and penalties for insufficient inventory. Taking into account penalties and inventory costs, we only tie contracts that minimize them. We also expect that other agents will move to reduce their inventory at the end of the game. So, to accommodate this, we concede the price of the contract as the game progresses. This allows us to maintain productivity throughout the game.

2 The Design of DogAgent

In this section, we explain DogAgent strategy. DogAgent consists of three strategies.

- Counter Strategy
- Offer Strategy
- Future Negotiation

DogAgent is based on the tutorial StdSyncAgent.

2.1 Counter Strategy

This section describes strategies for responding to offers sent by trading partners. In the scml world, there is a cost to inventory. However, this inventory cost is small compared to the shortfall penalty that occurs when a sales contract is concluded but inventory is insufficient. Therefore, the basic strategy is to have enough inventory to reduce the shortfall penalty. For input offers, the needs is the number of production lines minus the number of inventory, and for output offers, the needs is the smaller of the number of production lines and the number of inventory. The contract is then concluded by accepting the offer combination that is closest to the number of needs. By selecting the optimal contract, profits can be obtained consistently.

2.2 Offer Strategy

This section describes the strategy for offers sent from your agent to the counterparty. The method of determining the quantity of needs is the same as in 2.1 section. The price setting is important to determine whether the counterparty will accept the offer or not. Therefore, price concessions are made according to the steps of the game. In the final phase of the game, concessions are made up to 25% compared to the initial state. This ensures that productivity does not decrease throughout the game. This will maintain high productivity and stable profits throughout the entire game.

2.3 Future Negotiation

In the scml world, future contract offers can be sent to other agents. Negotiating future contracts can increase contract opportunities. This allows them to increase their productivity and aim for larger profits. However, it is difficult to perfectly predict future conditions, and the risk of penalties such as inventory shortages is high. Therefore, DogAgent does not negotiate future contracts, but concentrates on current contracts.

3 Evaluation

To evaluate DogAgent’s performance, We ran 5 standard track tournaments using the `anac2024_std()` method include in the template. The parameters are as follows.

- `n_steps = 100`
- `n_configs = 5`
- `competitors = [DogAgent, DoNothingAgent, BuyCheapSellExpensiveAgent]`

The results of the tournaments are shown in Table 1.

Table 1: Table 1: Score of tournaments

tournament	DogAgent	DoNothingAgent	BuyCheapSellExpensiveAgent
1st time	1.19	0.40	-0.36
2nd time	0.92	0.74	-0.10
3rd time	1.10	0.52	-0.05
4th time	1.00	0.70	0.65
5th time	0.94	0.72	-0.36
Average	1.03	0.616	-0.044

From this table we can see that our agents have the best scores. The initial score for each agent is 1.0. The two default agents have decreasing scores in all tournaments. However, our agent’s score is increasing in three tournaments. Although we were not able to obtain a significant increase in scores, we were able to obtain stable scores compared to the default agents.

Conclusions

In this report, we explained DogAgent strategy. This agent’s strategy is to reduce shortfall penalty and inventory cost as much as possible to obtain stable profit. We also offer price concessions as the game progresses. As a result, DogAgent showed a better score than other agents in Table 1. This time we were basing our contract on the amount of inventory, but it could be dynamic, depending on supply and demand. Another strategy is to increase productivity by optimizing the use of future contract offers. There is still room for improvement, but the agent is capable of obtaining a stable score.