SalesAgent: An agent submitted to the ANAC 2022 SCM league

Shunya Hibino[1], Takanobu Otsuka[2]
Nagoya Institute of Technology, Aichi, Japan
[1]hibino.shunya@otsukalab.nitech.ac.jp
[2]otsuka.takanobu@otsukalab.nitech.ac.jp

June 16, 2022

Abstract

The main concept of SalesAgent is to sell off products by changing the trading price based on the current number of steps. In SCML 2022, the products in the factory's inventory at the end of the game are valued at only half of the trading price. This means that if products are not sold, a loss is incurred. Therefore we change the trading price or limit the purchase of input products based on the current number of steps.

1 Introduction

The SCM world simulates a supply chain consisting of multiple factories that buy and sell products from one another. The factories are represented by autonomous agents that act as factory managers. Each agent decides which other agents to buy and sell from, and then negotiates with them. Their goal is to turn a profit, and the agent with the highest profit (averaged over multiple simulations) wins. According to the game description, in SCML2022, the profit (score) is calculated as follows:

$$\text{Profit} = \frac{\sum_{a \in F} B_N(f) + \epsilon I_N(f) - B_0}{\sum_{a \in F} B_0(f)} \tag{1}$$

where, ϵ is the fraction of trading price at which to value the inventory at the end of the game. F is the set of all factories controlled by instantiations of the agent, $B_0(f)$ and $B_N(f)$ are the factory's balances at the beginning and end of the simulation, respectively, and $I_N(f)$ is the value of the products in the factory's inventory at the end of the game. This value is based on the trading price, but to incentive trade, inventory is valued at only half the trading price; that way, it is more profitable on average to sell products rather than hoard them. Considering this, it is more profitable to sell product at more than half the trading price than to have the product at the end of the game. We focused on this and created an agent that doesn't finish the game with products in their possession.

2 The Design of SalesAgent

In this section, we explain our agent's strategy. SalesAgent consists of the following strategies:

- Supply Driven Production Strategy
- Trading Strategy
- Negotiation Manager

This agent was created based on the DecentralizingAgent, but the "Trading Strategy" and "Negotiation Manager" have been replaced with ours.

2.1 Supply Driven Production Strategy

In this game, having inputs at the end of the game is not profitable. Therefore, we thought that the Supply Driven Production Strategy, which converts all inputs into outputs, would work well. However, due to limitations imposed by the number of production lines, it is impossible to always convert all inputs into outputs. Considering this, we need to limit the number of products purchased in the TradingStrategy so as not to leave any inputs that are not converted.

2.2 Trading Strategy

Our Trading Strategy is based on Prediction Based Trading Strategy that uses prediction strategies to manage inputs/outputs needed. As mentioned in section 2.1, since we cannot convert all inputs to outputs, we have changed the rules so that we do not make a contract to buy products at more than half the number of steps. Furthermore, as mentioned in Chapter 1 and 2.1, it is more profitable to sell products at more than half the trading price than to have products at the end of the game. Therefore, below half of the number of steps, we do not allow selling contracts for less than the transaction price, but above half of the number of steps, we allow selling contracts for more than half of the transaction price. Also, to prevent bankrupts, we will not make a contract to purchase products that will reduce our balance to less than half of the balance. In SCML 2022, the number of steps is predetermined. However, in the real world, the time period to be negotiated is indeterminate. Therefore, our Trading Strategy, which currently changes the trading price based on the number of steps, is unstable in the real world. In order to improve our Trading Strategy, we believe that changing the trading price based on the sales of the commodity will work in the real world.

2.3 Negotiation Manager

Our Negotiation Manager is based on Step Negotiation Manager that controls a controller and another for selling for every timestep. We have changed urange function and acceptable unit price function. Urange function specifies a price range for negotiation. By default, urange function uses the catalog price, but we used the acceptable unit price, which is described below. Acceptable unit price was set to be above the output price for the seller and below the input cost for the buyer. In this way, when the products are sold, there will always be a profit.

3 Evaluation

To evaluate SalesAgent's performance, we ran five starter track tournaments (n steps=50, n configs=5) using the run() method included in the template. We added DecentralizingAgent and MarketAwareIndDecentralizingAgent as competitors. The parameters are as follows:

- competiton : std
- `reveal_names : True`
- `n_steps : 50`
- `n_configs : 2`

The results of the tournaments are shown in Table 1.

Table 1: Score of the execution result in the tournament

Tournament	SalesAgent	DecentralizingAgent	MarketAwareIndDecentralizingAgent
1	0.0600813	-0.187964	-1.48095
2	-0.104573	-0.364984	-1.82433
3	-0.067404	-0.226425	-0.350431
4	-0.0110029	-0.144158	-2.99742
5	0.0710587	-0.0422016	-1.58619
Average	-0.01036798	-0.19314652	-1.6478642

This table shows that our agent, SalesAgent had the best score all 5 times. Comparing the average scores, our agent outperformed the other two agents. Also, DecentralizingAgent had negative score values all the time, while our agent was positive twice.

Conclusions

In this report we have described SalesAgent. We also showed through experimentation that SalesAgent was superior to the DecentralizingAgent on which our agent was based. However, our agent did not gain enough profit and could not make its score consistently positive. The reason for this is that SalesAgent's strategy is to use the current step position to reduce losses. In other words, it focuses on reducing losses rather than gaining profits. In the future, by taking into account the state of the transaction in addition to the step position, we intend to be flexible in changing the trading price and sell as much of the manufactured product as possible in order to make a profit.