

SmartAgent: An agent submitted to the ANAC 2022 SCM league

Takuma Miwa¹, Takanobu Otsuka²
Nagoya Institute of Technology, Aichi, Japan,
¹miwa.takuma@otsukalab.nitech.ac.jp
²otsuka.takanobu@otsukalab.nitech.ac.jp

June 16, 2022

Abstract

The main strategy of Smart Agent is not to buy products in excess and to control selling price. In SCML2022, if there are extra input products at the end, it is more difficult to make profits. Therefore, our agent determines if the products are really needed before buying them. Moreover, our agent can adjust the selling price. In the real world, the selling price often fluctuates depending on sales. Our strategy incorporates this trend.

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 make more profit than other agents. According to the game description, in SCML2022, the profit is calculated as follows:

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

where, 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. ϵ is the fraction of trading price at which to value the inventory at the end of the game. In SCML2022, the fraction is fixed at 0.5. In other words, inventory is valued at only half the trading price.

Considering these rules, it is desirable to sell out output products even if the price is lowered by a few percent, and not to buy extra input products. Therefore, we made our agent, Smart Agent, has strategies that meet these demands.

2 The Design of Smart Agent

In this section, we explain our agent's strategy. Smart Agent consists of three strategies:

- Production Strategy
- Trading Strategy
- Negotiation Manager

Most of the Production Strategy is the same as Supply Driven Production Strategy. Trading Strategy is based on Prediction Based Trading Strategy, and Negotiation Manager is based on Independent Negotiations Manager.

2.1 Production Strategy

As we mentioned in the chapter 1, if there are input products left at the end, profits are less likely to be generated. So, Most of the Production Strategy is the same as Supply Driven Production Strategy that is converting all inputs to outputs.

However, not all inputs can be always converted to outputs. In SCML2022, the number of production lines is predetermined and each of them can produce one product in one step. In other words, the maximum number of products that can be produced is limited. Therefore, We need to control the number of products we buy in the Trading Strategy.

2.2 Trading Strategy

Our Trading strategy is based on Prediction Based Trading Strategy that uses prediction strategy to manage inputs and outputs needed. As we mentioned in the chapter 2.1, the maximum number of products that can be produced is limited and can be calculated by the number of production lines and the number of steps remaining. Our agent counts the number of products purchased and does not sign contracts that exceed this maximum.

In addition to the inventory control, our agent also controls trading price to make more profits. If we apply it to the real world, most of sellers often lower the price when the product does not sell at all. In contrast, they often raise the price to make more profit when the product sells well. Our strategy incorporates this trend. The selling price is lowered by one dollar when our agent could not sell at all in the previous step. Conversely, the price is raised by one dollar when our agent sold even one product in the previous step. Moreover, the price is lowered by 20% at the half of the number of steps so as not to have products in excess more reliably.

2.3 Negotiation Manager

Our Negotiation Manager is based on Independent Negotiations Manager. It includes the function `_urange`. This function returns range of unit price for negotiations. The range is between 0.5 and 1.5 times the trading price by default. In our Negotiation Manager, The range is changed to between 0.75 and 1.25 to prevent selling cheaply and buying expensively.

According to SCM league tutorial, Negotiation Manager is responsible for proactively request negotiations, responding to negotiation requests, and actually conducting concurrent negotiations. In this strategy, our agent sends negotiation requests to all partners every step.

3 Evaluation

In this section, we run 5 standard track tournaments and discuss the results. We evaluated Smart Agent in the tournaments. We added DecentralizingAgent and BuyCheapSellExpensiveAgent as competitors. The parameters are as follows:

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

The execution result of tournaments show Table1.

Table 1: Scores of 5 standard track tournaments

Tournament	SmartAgent	DecentralizingAgent	BuyCheapSellExpensiveAgent
1	0.0337617	-0.103655	-1.21672
2	0.106215	-0.139417	-1.03004
3	-0.0159312	-0.229476	-0.649182
4	0.0410252	-0.186068	-1.53347
5	0.0518865	-0.122975	-0.98999
Average	0.04339144	-0.1563182	-1.0838804

In these results, the best average score of these tournaments is Smart Agent. In addition, we got the best score in all five tournaments. Our agent got positive score four times even though all other agents got negative scores in the tournaments. These results show our agent often got the best score in these tournaments and our strategy is effective for this tournament.

Conclusions

In this report, we explained Smart Agent. The main strategy is not to store extra products and to adjust selling price under various situations. Our evaluation results show that our agent got more score than any other agents in the tournaments. In addition, they also shows that our strategy is working effectively.

However, the agent can not always get positive score. To improve the agent, it is effective to create new functions, for example, decides whom to negotiate with. In our strategy, the agent always sends negotiation requests to everybody. If this new function is incorporated, we can reduce the number of negotiations with less profitable agents.