Sorcery Agent: An agent submitted to the ANAC 2021 SCM league

Midori Takemoto[1], Takanobu Otsuka[2]
Nagoya Institute of Technology, Aichi, Japan
[1]takemoto.midori@otsukalab.nitech.ac.jp
[2]otsuka.takanobu@otsukalab.nitech.ac.jp

Abstract

The main strategy of the Sorcery Agent is to consider production cost and to control the buying price and the selling price. In real business, product price is changed by cost, for example, shipping cost, material cost, and production cost. However, in the SCM league, the default strategy doesn't consider these costs. To make it resemble real-world transactions, Sorcery Agent considers mainly cost and production number.

1 Introduction

In reality, the factory determines product price and quality by the costs incurred to handle it when produce, sell and buy. In particular, products prices are determined by money costs and time costs, and the prices are always changed. In addition, demand, supply, and budget determine production quantity.

In the SCM league, there are some differences from reality. The default strategies, Trading Strategy, Production Strategy, and Negotiation Manager don' t consider those costs and budget. Trading prices are written in the catalog, and these prices are used for trading. And, when determining production quantity, agents don' t consider operating funds. But to be realistic, as mentions earlier, the price is based on costs and production quantity is based on supply and demand.

Therefore, our agent, Sorcery Agent, takes them into account when signing contracts and determine production schedule and implement the function to change the production price.

According to the SCM league rules, if there are inputs left at the end, profits are less likely to be generated. Thus, we use Production Strategy as Supply Driven Production Strategy, as soon as agent get the inputs, we produce them as soon as you have the money.

2 The Design of Sorcery Agent

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

- Trading Strategy
- Production Strategy
- Negotiation Strategy

As mentions in Chapter one, Sorcery Agent's purpose keep it to real trading conditions. Our Trading Strategy is based on Prediction Based Trading Strategy. In this strategy, we implemented considering cost when determining the sell and buy prices. Production Strategy is based on Supply Driven Production Strategy. Production quantities determine based on operating funds. And Negotiation Strategy is based on Step Negotiation Manager.

2.1 Trading Strategy

This strategy is based on Prediction Based Trading Strategy. According to SCM league tutorial, Trading Strategy is deciding the quantity (and price) to buy and sell at every time-step. And Prediction Based Trading Strategy is the best trading strategy in the built-in agents. Therefore, we decide making base this strategy.

Prediction Based Trading Strategy uses two components, Trade Prediction Strategy and ERPrediction Strategy. Trade Prediction Strategy predicts the amount of trade on the input and output products of the agent as a function of the simulation step. And ERPrediction predicts the quantity that will be executed from a contract. These strategies predict the base conditions of the trading.

In addition to the previous strategies, we implemented the function of considering the costs. First, calculate production cost for calculating the trading price. Production cost could calculate from input cost. The buying price is obtained by subtracting the production cost from the catalog price, and the selling price is obtained by adding the production cost from the catalog price. These prices use in the transactions.

Furthermore, if our agent makes the last production, our factory produces it 25% more. In the tournament, the number of last products that can be sold is limited. Nevertheless, we have found if the factory has a lot of last product, they tend to buy as much as they can. Therefore, we could get profit to produce a little more of the last product.

2.2 Production Strategy

This strategy is based on Supply Driven Production Strategy. According to SCM league tutorial, production strategy decides what to produce at every step. This strategy converts all inputs to outputs. And this strategy inherits Production Strategy, it includes the function can_be_produced. This function returns True if the selling contract given can be honored in principle given the production capacity of the agent. But it does not check for the availability of enough money to run the production process. Thus, our strategy implements the function that checks for the availability of enough money to produce.

2.3 Negotiation Manager

According to SCM league tutorial, Negotiation Manager is responsible for proactively request negotiations, responding to negotiation requests, and actually conducting concurrent negotiations. In Sorcery Agent, Negotiation Manager is the same as the class in the tutorial. It starts negotiation to satisfy each other agent's contract condition using two SyncController objects one for selling and one for buying every simulation step.

3 Evaluation

In this section, we run the tournaments and discuss the results. We evaluated Sorcery Agent in the tournaments. The competitorsof Whagent were DecentralizingAgent, BuyCheapSellExpensiveAgent, RandomAgent,and MarketAwareDecentralizingAgent. The parameters are as follows:

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

We run this tournament four times under these parameters. The execution result of tournaments show Table1.

Table 1: Score of the execution result in the tournament

Agent Name	1	2	3	4	Average
Sorcery	-0.0434323	0	-0.0442208	-0.0189343	-0.035524
Decentralizing	-0.098456	-0.0525107	-0.0289526	-0.129887	-0.077451
MarketAwareDecentralizing	-0.104253	-0.0634032	-0.0324393	-0.150152	-0.087561
BuyCheapSellExpensive	-0.221671	-0.46323	-0.408673	-0.467855	-0.390357
Random	-0.506272	-1.08861	-0.848941	-1.22464	-0.917115

In this experiment, the best average score of these tournaments is our Sorcery Agent. However, in the third tournament, DecentralizingAgent got the best score. And almost all agents got negative scores, Sorcery Agent got a zero score once. Our agent could not get positive score but checked in the result of profits, sometimes made profits. These results show our agent often got the best score in these tournaments and our strategy is effective for this tournament.

4 Conclusions

In this report, we explained Sorcery Agent's strategy. The purpose of Sorcery Agent is to keep it to real trading conditions, and we updated Trading Strategy mainly. In chapter three, the result was shown our agent could often get the best score in the tournament. Sorcery Agent has strategies that could calculate production costs and consider available money. These strategies are our advantage, and the result shows it is effective for this tournament. However, getting the best score is not always. To improve our agent, it is effective to create new functions, for example, could more effective negotiation, could forecast the optimal trading price. If adding these strategies, we expected to get more profits in the tournament.