

MercuAgent: An agent submitted to the ANAC 2020 SCM league

Kazuto Kakutani¹, Takanobu Otsuka²
Nagoya Institute of Technology, Aichi, Japan

¹kakutani.kazuto@otsukalab.nitech.ac.jp, ²otsuka.takanobu@nitech.ac.jp

July 14, 2020

Abstract

We created an automatic negotiation agent that predicts transaction quantity and negotiates with other agents. The prediction method refers to the result of past negotiations. This agent manufactures purchase products immediately. It sells manufactured products the next day. We simulated this agent in SCML environment, it makes a little profit. It is possible to make larger profits by increasing the accuracy of the prediction.

1 Introduction

Supply chain management(SCM) is a management method that links the flow of goods and the flow of money with the flow of information. It shares and cooperates with the entire supply chain to do overall optimization. The sum of partial optimization doesn't always mean overall optimization. Managing and cooperating with each other based on the balance of the entire supply chain is extremely important.

The MercuAgent that we create refers to the information of the partners with whom we negotiate. It changes *unit_price*, *quantity* parameters and negotiation partners. It negotiates with various partners, not only a specific partner. In short, it will negotiate so that all negotiation partners make a profit.

In SCML2020, according to game description, the simulator calculates the profit as follows:

$$Profit = \frac{B_N + \frac{1}{2}I_N - B_0}{B_0} \quad (1)$$

where B_0 and B_N are the agent's balance at the beginning and end of the simulation. I_N is the value of the product in the agent's inventory at the end of the simulation. This value is half the trading price. Agents need not

pay the cost of owning products. But they sell them very cheaply at the end of the simulation. It becomes a loss. Also, input products don't make any profit at the end of the simulation. Therefore, to make profits, it is good to manufacture and sell all the products that they own.

2 Agent Strategy

Our agent's strategy is to keep as little inventory as possible. It produces products using materials that are purchased today. Then, it sells these products tomorrow. If our agent has inventory near the last day, it sells these products even if they are cheap.

2.1 Agent Configuration

The MercuAgent has three classes we created and one required class as follows:

- MyNegotiationManager
- MyTradingStrategy
- MyProductionStrategy
- SCML2020Agent

“MyNegotiationManager” is based on “StepNegotiationManager”, and instead of controllers that manage negotiations step by step, it creates controllers that manage negotiations for sellers or buyers. “MyTradingStrategy” is based on the “TradePredictionStrategy”. These strategies we created are discussed in detail in chapters 2.2 and 2.3, respectively. “MyProductionStrategy” is the same as the “SupplyDrivenProductionStrategy”. We consider this production strategy is optimized. “SCML2020Agent” is required class in SCML2020.

2.2 Negotiation Manager

This negotiation manager is created based on “StepNegotiationManager”. Its functions basically work the same as the original manager's function. Our manager has the seller's and buyer's controllers respectively manage sales and purchase negotiations. The MercuAgent generates these controllers from “MyController” class at the start of the negotiation. These controllers are updated at the start of each step negotiation. This class has `target_quantity`, `is_seller`, `agent_confidence`, `step`, `urange`, `product`, `partners`, `negotiator_type`, `horizon`, `awi`, `parent_name`, `negotiations_concluded_callback`, `negotiator_params`, `max_retries` as argument.

The new one we added is *agent_confidence*. This represents the reliability

of the agent. Our agent sets confidence values for other negotiation agents. It detects a bankrupt agent using this value and does not trade it. If the value is less than the threshold, it doesn't negotiate with that agent. It preferentially negotiates with agents with high confidence values. The confidence value setting method is as follows:

At each step, our agent refers to Financial Report. It uses the breach probability in the financial report to set the confidence value of the negotiation agent. It is $1 - (\text{breachprobability})$.

These controllers try to do concurrent negotiation. This manager is incomplete.

2.3 Trading Strategy

The MercuAgent estimates the necessary quantity for a day in advance, and negotiates with other agents aiming at the estimated quantity. This is useful for it to keep as little inventory as possible. The estimation method is as follows:

- Sales quantity estimation method:
It estimates the sales quantity of the next day from the average purchase quantity of guaranteed contracts.
- Purchase quantity estimation method:
It estimates the purchase quantity of the next day from the average sales quantity of guaranteed contracts.

Then, this agent evaluates the estimated quantity and the quantity actually contracted at that step. It sets the correction value by this evaluation value and adjust the estimated quantity. The estimated quantity has the following effects depending on the correction value:

If the contracted quantity is greater than or equal to the estimated quantity, it is increased. If the contracted quantity is lower than the estimated quantity, it is decreased.

Therefore, the estimated quantity is as follows:

$$\text{EstimatedQuantity} = E_n + C_n \quad (2)$$

where E_n represents the value of the estimation method. C_n represents the correction value.

Our agent makes the delivery date of the contract as short as possible. If the delivery date is unacceptably long, it is generally not signed the contract. Also, it offers a counter offer so that the delivery date is shortened.

2.4 Risk Management

Parallel controlling contracts cause to excessive contracts. This causes more contracts than our agent estimated. Therefore, as described in chapter 2.2, it doesn't over contract by centrally controlling negotiations using two controllers. As described in chapter 2.3, Our agent sets confidence values for other negotiating agents. It avoids negotiations with agents that go bankrupt, and negotiates a stable quantity.

A threshold is set for the delivery date, unit price, and quantity of the contract and contracts that don't meet the threshold are not signed. This prevents disadvantageous contracts.

3 Evaluation

To evaluate the agent's performance, we experiment in the SCML2020 simulator. The parameters are competition='std', n_steps=50, n_configs=4, n_runs_per_world=1, worlds=12 or 24. The MercuAgent(MyAgent), DecentralizingAgent, BuyCheapSellExpensiveAgent are competitors. Table 1 shows the simulation results. Our agent's score is much higher than BuyCheapSellExpensiveAgent's score, but lower than DecentralizingAgent's score. DecentralizingAgent is positive value. MercuAgent and BuyCheapSellExpensiveAgent is negative value. BuyCheapSellExpensiveAgent is high negative value, but MercuAgent is almost zero.

experiments	MercuAgent	DecentralizingAgent	BuyCheapSellAgent
1	-0.00567311	0.0592968	-1.49758
2	-0.0262924	0.0496923	-1.64472
3	-0.00621693	0.136553	-0.95936
4	0.0126465	0.218299	-0.873316
5	-0.0760885	-0.0600613	-1.42085
Average	-0.006628958	0.047369416	-1.1219652

Table 1: Agent's score

4 Conclusions

In this report, we described the MercuAgent we created. This agent mainly estimates the purchase quantity and sales quantity. This agent is a better performing agent than BuyCheapSellExpensiveAgent. We believe that improving the strategy will give better results.

In the future, we improve the accuracy by using machine learning to predict the negotiation quantity and optimize the utility function. To integrate two controllers and do concurrent negotiations enable to make an optimal choice.