Ashgent: An agent submitted to the ANAC 2020 SCM league

Shuhei Aoyama[1], **Takanobu Otsuka**[2]

[1,2]Nagoya Institute of Technology, Aichi, Japan
[1]Email id: aoyama.shuhei@otsukalab.nitech.ac.jp
[2]Email id: otsuka.takanobu@otsukalab.nitech.ac.jp

July 14, 2020

Abstract

Ashgent manages risk by focusing on the breach levels of the opponents to negotiate with. By reading the breach levels, we adjusts the amount of our needed/secured quantity by looking at the breach level of the opponent to negotiate with. Further in order to make a profit, Ashgent predicts the spot market prices of the products we trades. Thereby, Ashgent stably performs better in competition with built-in agents.

1 Introduction

The parameters needed/secured are used for negotiation. The smaller the difference between needed and secured, the more the stock of the input product required for the step is sufficient, while the larger the difference, the more the stock of the input product is insufficient. These are asked for at each step and are used to determine which negotiations are signed at that step. It is essential that the exact determination of these parameters to ensure that there is no excess or shortage of inventory.

For input and output products, the value depends on the spot market price, which is updated step by step. However, these are not published during the simulation. Then Ashgent predicts the spot market price at each step by using its signed contracts. Furthermore, we predict the value of the products we handle and changes our behavior. When the value of the input product is high, we refrain from negotiating and wait for the value of the input product to fall. While when the value of the output product is high, we sell it at the highest possible price. By making our signatures more flexible, differentiate Ashgent from other agents.

2 The Design of Ashgent

2.1 Negotiation Choices

Ashgent is based on the DecentralizingAgent, a built-in agent. In this report, we describe the following built-in strategies of DecentralizingAgent that have been improved for the Ashgent:

- MyTradePredictior (based on FixedTradePredictionStrategy)
- MyTrader (based on PredictionBasedTradingStrategy)
- MyManager (based on StepNegotiationManager)

Ashgent's strategy is based on predicting. In MyTradePredictior, each time the on_contracts_finalized method is called, the necessary predictions are made comprehensively. MyTrader updates the needed/secured from the results predicted by MyTradePredictior. Similarly, in the acceptable_unit_price method, MyManager adjusts the acceptable prices from the predictions made by MyTradePredictior

2.2 Risk Management

Focusing on risk management is one way to make a profit. Failure to execute the sell contracts that was concluded will result in a loss. By trading in consideration of risks, we will ensure inventory and that production can be carried out as planned. This will prevent breaches of sall contracts and eventually prevent the worst of our factory going bankrupt.

According to the game discription, the breach level increases with each breach of the contract. It can be confirmed through financial reports. Using this, each time a negotiation is finalized, Ashgent predicts how much is actually likely to be secured by that negotiation. By negotiating and securing a large amount of stock, we can prevent a situation in which production cannot be carried out due to a shortage of stock, even if the negotiation breaches.

Ashgent predicts the amount that will actually be obtained for the contracts concluded using the following formula:

$$Quantity = q * (1 - breach\_level), \tag{1}$$

where q is the amount agreed upon in the contract, and $breach_level$ is the breach level of the counterparty to the contract, which is taken from the financial report. $Quantity$ calculated by penalizing $breach\_level$ is subtracted from needed and added to secured.

If the counterparty has never breached, q becomes the $Quantity$ as it is, and the more breaches made by the counterparty, the smaller the quantity. The smaller the Quantity, the smaller the effect on needed/secured. That makes Ashgent behave as if it signed more contracts. As a result, it is possible to limit the disadvantages if the contract is not fulfilled.

2.3 Spot Market Prices Prediction

Predicting market prices and grasping market trends is one of the ways to make a profit. If we can figure out which product price is rising and which one is falling from the market, we can get the contracts in our favor by taking measures against them.

Ashgent predicts the spot market price at that step for the contracts concluded using the algorithm 1.

Algorithm 1 Spot Market Prices Prediction

Require: $step \geq 2$
  for (p, q) in $Contracts$ **do**
    $predict \Leftarrow P_{step}$
    for $i = 0$ to q **do**
      $predict \Leftarrow ((1 - contract\_weight) * predict + contract\_weight * p)$
    end for
    $predict = (breach\_level * P_{step} + (1 - breach\_level) * predict)$
    $P_{step} = \frac{(P_{step-2} + P_{step-1} + predict)}{3}$
  end for

In this algorithm, we derive the spot market price prediction P_{step} of the step at that point *step* from the agreed price p and agreed quantity q in all contracts $Contracts$. The weight $contract\_weight$ of p in the *predict* was set to 0.5 based on empirical results. It also reflects the final predicted prices up to two steps ago, P_{step-2} and P_{step-1}. This allows us to reflect changes in the spot market prices due to negotiations, while reflecting the time-series trend of them.

3 Experiments

To evaluate Ashgent's performance, we run tournaments using the anac2020_std method, and give the following arguments:

verbose True

n_steps 50

n_configs 2

n_runs_per_world 1

As competitors, we adopted the relatively intelligent built-in agents DecentralizingAgent, IndDecentralizingAgent, and MovingRangeAgent. Five rounds of the tournament were held under the set conditions, and the results are shown in Table 1. This table shows that Ashgent outperforms the other agents in all five tournaments.

Among them, it can be seen that even when the entire world is in recession such as Tournament 3, the score loss is most suppressed in Ashgent. This shows that Ashgent is able to manage risks well even in an environment of frequent contract breaches. The Decentralizing Agent, which is the origin of Ashgent, lost to MovingRangeAgent in Tournament 3 and to IndDecentralizingAgent in Tournament 4, but Ashgent showed stable and superior results. Hence, it is considered that the adjustment of trades by predicting spot market prices led to a superior score over other agents.

Table 1: Score of Tournaments

Tournament	**Ashgent**	DecentralizingAgent	IndDecentralizingAgent	MovingRangeAgent
1	**0.12436**	0.10937	-0.54771	-0.12771
2	**0.02715**	-0.09767	-0.25650	-0.10419
3	**-0.05021**	-0.24999	-0.31159	-0.08295
4	**0.06417**	-0.18742	-0.18327	-0.09477
5	**0.08498**	-0.17369	-0.35428	-0.11134
Average	**0.05009**	-0.11988	-0.33067	-0.10419

Conclusions

Ashgent implements the fundamentals of risk management and improves profits by predicting spot market prices. We showed a stable and excellent performance as shown in Experiments. By applying risk management, Ashgent suppresses losses even in a recessed world. And even in a world that is not recessed, stable profit is generated by predicting the spot market price.

However, The accuracy of the spot market price predictions are not guaranteed because it cannot reflect contracts signed between other agents. Therefore, it is possible that we may make predictions that deviates from the spot market prices. In the future, we would like to improve the accuracy by reflecting rejected contracts in the predictions. By carefully reflecting the information obtained in the predictions and adjust the behavior in the strategy using the predictions, it is possible to create even better agents.