

# CrescentAgent : An agent submitted to the ANAC 2020 SCM league

Yuki Yoshimura<sup>1</sup>, Takanobu Otsuka<sup>2</sup>  
Nagoya Institute of Technology, Aichi, Japan  
yoshimura.yuki@otsukalab.nitech.ac.jp<sup>1</sup> otsuka.takanobu@nitech.ac.jp<sup>2</sup>

July 14, 2020

## Abstract

The purpose of this agent is not to take a loss. The amount of the input product the agent needs at every time step is set as the number of lines not to trade the amount the agent cannot produce. If the agent can link a sales contract to purchase contracts, it signs these contracts. In this way, the agent doesn't sign a sales contract that it cannot secure the quantities of by the delivery time, and the agent doesn't sign purchase contracts that create stock.

## 1 Introduction

According to the game description in SCML2020, an agent's performance will be measured by its score. An agent's score will be the median of the profits accrued by all the factories it is assigned to manage across all simulations. It means that we don't have to consider unusual situations. The game description in SCML2020 also said, the profit accrued by a factory during one simulation is calculated as follows:

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

where  $B_0$  and  $B_N$  are the agent's balances at the beginning and end of the simulation, and  $I_N$  is the value based on the trading price of the products in the agent's inventory at the end of the game. From equations(1), we can understand two things. First, it is obvious, but it is better than the difference of the agent's balance between at the beginning and end of the simulation is bigger. The agent has to buy cheaper input products and

sell more expensive output products and the agent has to do more products. Second, the products in the agent's inventory become profits. Generally, the trading price of an input product is more expensive than trading price of an output product, so it is better to produce output products if it has input products, except that production cost is more expensive than the trading price of an output product.

## 2 The Design of CrescentAgent

The class CrescentAgent inherits from the class MyTradingStrategy, the class MyNegotiationManager, the class MyProductionStrategy, and the class SCML2020Agent and CrescentAgent consists of three strategies that are trading strategy, negotiation control strategy, and production strategy. This design is based on tutorials in SCML2020. I used these strategies because the SCML platform provides several components that can be used to implement.

### 2.1 Trading Strategy

The class MyTradingStrategy inherits from the class SignAllPossible and the class TradingStrategy. According to the tutorials in SCML2020, trading strategy decides the quantity and price to buy and sell at every time-step and consists of two subcomponents that are a pre-negotiation component and a post-negotiation component.

The tutorials in SCML2020 said a pre-negotiation component decides the quantities/prices to negotiate about based on a prediction of future market behavior and partner behavior, but, about this time, a pre-negotiation component is not based on prediction and only decides the quantity to buy and sell at every time-step. It means the `inputs_needed/outputs_needed` that are member variables of TradingStrategy are only set to the number of lines except for the last step/the first step.

The tutorials in SCML2020 said a post-negotiation component decides what agreements to sign as contracts. We focused on a post-negotiation component and write these contents in the `sign_all_contracts` function in the class MyTradingStrategy. The `sign_all_contracts` function links between a sales contract and purchase contracts that has the quantities more than the quantities of the sales contract and each the unit price less than unit price of the sales contract. Of course, this agent can produce products it bought by the delivery time of the sales contract.

## 2.2 Negotiation Control Strategy and Production Strategy

According to the tutorials in SCML2020, the negotiation control strategy is responsible for proactively request negotiations, responding to negotiation requests, and actually conducting concurrent negotiations. The class `MyNegotiationManager` is the same as the class in the tutorials and it will start negotiations to satisfy the needs that it gets from the trading strategy using two `SyncController` objects one for selling and one for buying every simulation step.

According to the tutorials in SCML2020, production strategy decides what to produce at every time-step. The class `MyProductionStrategy` is based on class `SupplyDrivenProductionStrategy`. I just set the argument method of `self.awi.schedule_production` in `on_contracts_finalized` to `""earliest""` because, generally, the trading price of an input product is more expensive than the trading price of an output product, so it is better to produce output products if it has input products, except that production cost is more expensive than the trading price of an output product.

## 3 Evaluation

To evaluate the agent's performance, we experimented with the `run()` function in the template. The scores of each agent for the five experiments shown in Table 1. The parameters are as follows:

Table 1: Scores of each agent

Experiments	CrescentAgent	DecentralizingAgent	BuyCheapSellExpensiveAgent
1	0	0.378322	-0.146419
2	0.00524776	0.230126	-1.12283
3	0	0.542438	-0.977032
4	0	-0.00861694	-0.505319
5	0.0213195	0.246759	-0.412767

- `competition=std,`
- `revealnames=True,`
- `nsteps=20,`
- `nconfigs=2,`

- maxnworldsperconfig=None,
- nrunsperworld=1,

and competitors are DecentralizingAgent and BuyCheapSellExpensiveAgent. As you can see, CrescentAgent won five experiments against BuyCheapSellExpensiveAgent, but, in most cases, CrescentAgent lost to DecentralizingAgent. We can also see CrescentAgent's score is zero in most cases and it means CrescentAgent traded nothing. Besides, CrescentAgent's score is not minus in five experiments and it means CrescentAgent doesn't take a loss.

## Conclusions

In this report, I described CrescentAgent's strategy. Finally, this agent's purpose not to take a loss is reached because CrescentAgent's score is not minus. It is important in this strategy to link between a sales contract and purchase contracts, but an agent signs contracts only if the agent links between these contracts. So, if an agent doesn't link, the agent sign no contracts and the agent's score will be zero. It might have been better to search purchase contracts to be able to link in the next step or later if it couldn't link in the step.