M5: An agent submitted to the ANAC 2022 SCM league

Kazuki Komori, komori@katfuji.lab.tuat.ac.jp

June 18, 2022

1 The Design of M5

This section describes the design of M5. M5 inherits $IndependentNegotiationsManager$, $SupplyDrivenProductionStrategy$ and $SCML2020Agent$.

1.1 Negotiation Choices and Utility Function

M5 makes negotiation choices as per the class $IndependentNegotiationsAgent$. M5 does not coordinate the behavior among multiple simultaneous negotiations. We set utility function U, target quantity q^{target} and acceptable unit price $(p^{accept}_{i,output}, p^{accept}_{i,input})$, where i is the current step, as follows.

if selling:

$$\begin{aligned}
U &= LinearUtilityFunction((10, 2, 1)) \\
q^{target} &= 2 \times n_{lines} \\
p^{accept}_{i,output} &= \begin{cases} p^{catalog}_{output} - 1 & \text{if } i < 0.2 \times n_{steps} \\ \min\left(1.05 \times p^{accept}_{i-1,output}, p^{catalog}_{output} - 1\right) & \text{else if } q^{output}_{i-1} > 0 \\ \max\left(0.95 \times p^{accept}_{i-1,output}, 1.05 \times \left(p^{catalog}_{input} + c\right)\right) & \text{else if } i < 0.7 \times n_{steps} \\ \max\left(0.95 \times p^{accept}_{i-1,output}, 0.55 \times p^{catalog}_{output}\right) & \text{otherwise} \end{cases}
\end{aligned}$$

if buying:

$$\begin{aligned}
U &= LinearUtilityFunction((1, -2, -4)) \\
q^{target} &= 2 \times n_{lines} \\
p^{accept}_{0,input} &= p^{catalog}_{input} \\
p^{accept}_{i,input} &= \begin{cases} 0.95 \times p^{accept}_{i-1,input} & \text{if } q^{input}_{i-1} > 0 \\ \min\left(1.05 \times p^{accept}_{i-1,input}, p^{catalog}_{input}\right) & \text{otherwise} \end{cases}
\end{aligned}$$

$$
\begin{array}{ll}
n_{lines} & : \text{number of production lines in the factory} \\
n_{steps} & : \text{number of simulation steps} \\
p_{output}^{catalog} & : \text{catalog price of the output} \\
p_{input}^{catalog} & : \text{catalog price of the input} \\
q_{i-1}^{output} & : \text{quantity of outputs contracted at the last step} \\
q_{i-1}^{input} & : \text{quantity of inputs contracted at the last step} \\
c & : \text{production cost}
\end{array}
$$

M5 aims to reduce unsold products. To achieve this, M5 makes it easier to sell and harder to buy. Quantity is weighted higher for selling and unit price is weighted higher for buying.

If there was a selling contract at the last step, the selling price will be rased. If there was no selling contract, it will be redused. In the case of buying, the opposite will be adjusted.

1.2 Risk Management

M5 decides whether or not to sign contracts according to the following rules for risk management.

selling contracts

The selling contracts (quantity q, delivery time t, unit price p) are signed to satisfy the following condition:

$$
\begin{array}{rcl}
\text{if M5 is located in 0-layer:} & & \\
I_t^{output} & \geq & \sum_{k=t}^{n_{steps}} q_k^{shipment} + \sum^{sign} q, \\
\text{otherwise:} & & \\
\hat{I}_t^{output} & \geq & \sum_{k=t}^{n_{steps}} q_k^{shipment} + \sum^{sign} q,
\end{array}
$$

where t is the delivery step of the contract, I_t^{output} is the expected output inventory at the step t, $\hat{I}_t^{output}$ is the expected minimum inventory of outputs at the step t, and $q_k^{shipment}$ is the quantity of shipments at the step k.

If M5 is located in 0-layer, M5 is endowed with exogenous buy contracts. These cannot be breached by the opponent. M5 trust all buying contracts and M5 sign the selling contracts if there are expected inventory at the delivery date.

If M5 is not located in 0-layer, M5 sign buying contracts with other agents. These can be breached by the opponent. M5 signs selling contracts for only the quantity M5 can sell with the current inventory. In other words, M5 does not trust all buying contracts that will be executed in the future. As a result, M5 will never breach the selling contracts.

buying contracts

The buying contracts (quantity q, delivery time t, unit price p) are signed to satisfy the following conditions:

$$
\begin{aligned}
t &< 0.6 \times n_{steps} \\
p_{i,output}^{accept} &> 1.05 \times \left(p_{input}^{catalog} + c\right) \\
I_t^{input} + \sum_{k=t+1}^{n_{steps}} q_k^{arrival} + \sum^{sign} q &\leq n_{lines} \times (n_{steps} - t - 1) \\
\frac{b_i - b_0 - \sum^{sign}(p+c) \times q}{b_0} &> r + s \times \frac{i}{n_{steps}} \\
r &= -0.25 \times \frac{n_{steps}}{200} \\
s &= 0.55 \times \frac{n_{steps}}{200}
\end{aligned}
$$

where i is the current step, I_t^{input} is the expected inventory of inputs at the delivery date t, $q_k^{arrival}$ is the quantity of arrivals at the step k, and b_i is the balance at the step i. By the first condition, M5 does not sign the buying contracts at the end of the simulations. By the second condition, M5 does not sign the buying contracts when the selling price is too low. By the third condition, M5 does not sign if it is impossible to produce. By the fourth condition, M5 prevents itself from spending too much money. The larger the value of n_{steps}, the larger the amount of money to be spent.

1.3 Collusion Strategy

M5s share the information of the located layers and the production cost. The agents in the same layer l reduce their buying price by the difference in production cost $c - c_{min}^l$, where c_{min}^l is the minimum cost of all M5s in layer l. The buying price is always less than $p_{i,input}^{accept} - (c - c_{min}^l)$ on the collusion track. As a result, there is no conflict between M5s in the same layer. The smaller the production cost, the larger the quantity of trades.

This purchase price adjustment allows M5s to spend more money and target more profit. M5s change the values of r and s in the buying contracts on the collusion track:

$$
\begin{aligned}
r &= -0.3 \times \frac{n_{steps}}{200}, \\
s &= 0.7 \times \frac{n_{steps}}{200}.
\end{aligned}
$$

2 Evaluation

This section describes the evaluation of M5. Table 1 shows the results of competing with DecentralizingAgent and BuyCheapSellExpensiveAgent. The configurations for competition are n_steps: 100, n_configs: 8 and running 24 simulations. The score on the collusion track are not high because the score when it is alone in the market is greater than the others.

Table 1: The results of competing with default agents

Track	M5	DecentralizingAgent	BuyCheapSellExpensiveAgent
Standard	0.0078	-0.1735	-1.3946
Collusion	0.0117	-0.0593	0.1272