M4: An agent submitted to the ANAC 2021 SCM league

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

July 30, 2021

1 The Design of M4

This section describes the design of M4. M4 inherits $IndependentNegotiationsManager$, $SupplyDrivenProductionStrategy$ and $SCML2020Agent$.

1.1 Negotiation Choices and Utility Function

M4 makes negotiation choices as per the class $IndependentNegotiationsAgent$. M4 does not coordinate the behavior among multiple simultaneous negotiations. We set utility function U, target quantity q^{target} and acceptable unit price p_i^{accept}, where i is the current step, as follows.

if selling:

$$\begin{aligned}
U &= LinearUtilityFunction((1,1,10)) \\
q^{target} &= n_{lines} \\
p_0^{accept} &= 1.1 \times p_{output}^{catalog} \\
p_i^{accept} &= \begin{cases} 1.1 \times p_{i-1}^{accept} & \text{if } q_{i-1}^{output} > 0 \\ \max\left(0.95 \times p_{i-1}^{accept}, p_{output}^{catalog}\right) & \text{else if } i < 0.5 \times n_{steps} \\ \max\left(0.95 \times p_{i-1}^{accept}, p_{input}^{catalog} + c\right) & \text{else if } i < 0.8 \times n_{steps} \\ 0.9 \times p_{i-1}^{accept} & \text{otherwise} \end{cases}
\end{aligned}$$

if buying:

$$\begin{aligned}
U &= LinearUtilityFunction((1,-1,-10)) \\
q^{target} &= n_{lines} \\
p_0^{accept} &= 0.9 \times p_{input}^{catalog} \\
p_i^{accept} &= \begin{cases} 0.9 \times p_{i-1}^{accept} & \text{if } q_{i-1}^{input} > 0 \\ \min\left(1.05 \times p_{i-1}^{accept}, p_{input}^{catalog}\right) & \text{otherwise} \end{cases}
\end{aligned}$$

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

M4 weights quantity, delivery time and unit price in the ratio 1:1:10. 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

M4 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 sorted in order of increasing unit price p and signed to satisfy the following condition:

$$
\begin{aligned}
&\text{if M4 is located in 0-layer:} \\
&\qquad I_t^{output} \geq \sum_{k=t}^{n_{steps}} q_k^{shipment} + \sum^{sign} q, \\
&\text{otherwise:} \\
&\qquad \hat{I}_t^{output} \geq \sum_{k=t}^{n_{steps}} q_k^{shipment} + \sum^{sign} q,
\end{aligned}
$$

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 M4 is located in 0-layer, M4 is endowed with exogenous buy contracts. These cannot be breached by the opponent. M4 trust all buying contracts and M4 sign the selling contracts if there are expected inventory at the delivery date.

If M4 is not located in 0-layer, M4 sign buying contracts with other agents. These can be breached by the opponent. M4 signs selling contracts for only the quantity M4 can sell with the current inventory. In other words, M4 does not trust all buying contracts that will be executed in the future. As a result, M4 will never breach the selling contracts.

buying contracts

The buying contracts (quantity q, delivery time t, unit price p) are sorted in order of decreasing unit price p and signed to satisfy the following conditions:

$$\begin{aligned}
I_t^{input} + \sum_{k=t+1}^{n_{steps}} q_k^{arrival} + \sum^{sign} q &\leq n_{lines} \times (n_{steps} - t - 1) \\
\frac{b_i - b_0 - \sum^{sign}(p+c) \times q}{b_0} &> -0.2 + 0.5 \times \frac{i}{n_{steps}} \\
t &< 0.8 \times n_{steps}
\end{aligned}$$

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, M4 does not sign if it is impossible to produce. By the second condition, M4 prevents itself from spending too much money. By the third condition, M4 does not sign the buying contracts at the end of the simulations.

1.3 Collusion Strategy

M4s 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 M4s in layer l. The buying price is always less than $p_i^{accept} - (c - c_{min}^l)$ on the collusion track. As a result, there is no conflict between M4s in the same layer. The smaller the production cost, the larger the quantity of trades.

2 Evaluation

This section describes the evaluation of M4. Table 1 shows the results of competing with DecentralizingAgent and BuyCheapSellExpensiveAgent. The configurations for competition are n_steps: 50, n_configs: 10 and running 60 simulations.

Table 1: The results of competing with default agents

Track	M4	DecentralizingAgent	BuyCheapSellExpensiveAgent
Standard	0.1090	-0.3724	-1.2076
Collusion	0.0727	-0.2817	-1.2177