CautiousStdAgent: An agent submitted to the ANAC 2024 SCML Standard Track

Ryoga Miyajma
Tokyo University of Agriculture and Technology
miyajima@katfuji.lab.tuat.ac.jp

April 17, 2024

Abstract

This report provides an explanation of CautiousStdAgent for ANAC2024 SCML Standard Track. CautiousStdAgent adopts a cautious strategy focused on inventory control. CautiousStdAgent can obtain a higher score than many other SCML agents.

1 Introduction

In Standard Track, agents need to manage their inventory because they can stock their input products. CautiousStdAgent adopts a cautious strategy focused on inventory control. It aims to buy materials as inexpensively as possible and sell out of inventory as soon as possible every day.

2 The Design of CautiousStdAgent

Basically, to avoid a shortfall penalty, CautiousStdAgent conducts sales negotiations based on the current inventory and the already-determined arrival schedule for that day.

2.1 Buying Strategy

Decision of Buying Price Range

The buying price range is determined depending on inventory changes. The maximum price for buying at day $t(0 \leq t \leq T-1)$ $p_t^{max\_for\_buying}$ is determined in Equation (1) below.

$$p_t^{max\_for\_buying} = \begin{cases} cp^{input\_product} & \text{if } t = 0 \\ \max(0.8p_{t-1}^{max\_for\_buying}, 0.9p_{min}^{input\_product} + 0.1p_{max}^{input\_product}) & \text{if } I_t > I_{t-1} \\ p_{t-1}^{max\_for\_buying} & \text{if } I_t = I_{t-1} \\ \min(1.1p_{t-1}^{max\_for\_buying}, cp^{output\_product} - p^{produce} - 1) & \text{if } I_t < I_{t-1} \end{cases}, \tag{1}$$

where $[p_{min}^x, p_{max}^x]$ is the range of the negotiation issue of the price of product x, cp^x is the catalog price of product x, and I_t is the inventory at day t.

The maximum price for buying at relative negotiation round $r(0 \leq r \leq 1)$ at day t $p_{t,r}^{max\_for\_buying}$ is determined in Equation (2).

$$p_{t,r}^{max\_for\_buying} = p_{min}^{input\_product} + (p_t^{max\_for\_buying} - p_{min}^{input\_product})r^{0.5} \tag{2}$$

Algorithm 1 Making Offers for Consumers

1: **for** each day t in *current_step* to n_{steps} **do**
2: **if** *remained_consumers* is empty or *remained_output_needs* $= 0$ **then**
3: break
4: **end if**
5: **if** $t =$ *current_step* **then**
6: $tmp\_output\_needs \leftarrow \min(remained\_output\_needs, n_{lines} - todays\_accepted\_quantity)$
7: **else**
8: $tmp\_output\_needs \leftarrow \min(remained\_output\_needs, n_{lines})$
9: **end if**
10: Calcurate *distribution* by distributing *tmp_output_needs* among remained_consumers giving priority to frequent consumers.
11: **for** each consumer k, distributed quantity q in *distribution* **do**
12: **if** $q > 0$ **then**
13: Calculate offer price p
14: Offer k to (q, t, p)
15: **end if**
16: **end for**
17: $remained\_output\_needs \leftarrow remained\_output\_needs - tmp\_output\_needs$
18: Remove distributed consumers from *remained_consumers*.
19: **end for**

Negotiation with Suppliers

CautiousStdAgent accepts offers from suppliers in preference to those with lower prices. First, it sorts the offers in descending order of unit price. Then, CautiousStdAgent picks one by one from sorted offers and if it still has room to reach the production limit, the delivery date of the offer is today, and the unit price of the offer is equal to or less than $p_{r,t}^{max\_for\_buying}$, it accepts the offer. If the offer satisfies the quantity condition but does not satisfy the delivery date condition, CautiousStdAgent proposes the counter offer with the delivery date changed to today. If the offer satisfies the quantity condition but does not satisfies the price condition, CautiousStdAgent proposes the counter offer with the unit price changed to $p_{r,t}^{max\_for\_buying}$.

2.2 Selling Strategy

CautiousStdAgent negotiates with its consumers based on the current inventory and exogenous input contracts. When CautiousStdAgent receives offers from consumers, it selects a combination from a power set of the offers that sells the most quantity without exceeding the current inventory and current exogenous input quantity. If such combination exits more than one, CautiousStdAgent selects the one maximizing the total price and accepts it.

If it remains unsold input products, CautiousStdAgent sends offers to its consumers for distribution so that it can be sold at the earliest possible delivery date, while taking into account the daily production limit. This procedure is shown in Algorithm 1. p in line 13 of Algorithm 1 is calculated by Equation (3).

$$\begin{aligned} mn &= \max(cp_{min}^{input\_product} + p^{produce}, p_{min}^{output\_product}) \\ mx &= (mn + p_{max}^{output\_product})/2 \\ p &= mx - (mx - mn)r^{0.5} \end{aligned} \tag{3}$$

Table 1: Result of the Standard Tournament

Agent	Score
CautiousStdAgent	**1.106**
QuantityOrientedAgent	0.943
GreedyStdAgent	0.645
SyncRandomStdAgent	0.400

2.3 Risk Management

Final Level Risk

Basically, CautiousStdAgent takes supply based sales strategy. However, if it is in the final level layer, sales contracts is definitively signed with BUYER as exogenous contracts, so it is important for CautiousStdAgent in the final level layer to meet the demand from BUYER. When it received offers from its suppliers, it selects a combination from a power set of the offers that is enough to the demand from BUYER and minimizing the over quantity error and accepts it. If such combination exists in more than one, CautiousStdAgent selects the one with the lowest total price.

Deficit Transaction Risk

In SCML2024 Standard Track, the unit-price issue will range between $\lfloor(1-\kappa)tp\rfloor$ and $\lceil(1+\kappa)tp\rceil$ where tp is the current trading price of the product being negotiated. Since tp is a dynamic variable, it can be happen that it is impossible to sell output product at a profitable price. Therefore, when the condition of Equation (4) is satisfied, CautiousStdAgent does not buy input products.

$$p_{min}^{input\_product} + p^{produce} > tp^{output\_product}, \tag{4}$$

where $p^{produce}$ is the production cost.

3 Evaluation

To evaluate CautiousStdAgent, I tested it in Standard tournaments with the sample agents, GreedyStdAgent and SyncRandomStdAgent and the winner of SCML2023 OneShotTrack, QuantityOrientedAgent. The configurations are $n\_steps = 100$ and $n\_configs = 5$. The results are shown in Table 1.

As Table 1 shows, the score of CautiousStdAgent is higher than those of any other agents. Only CautiousStdAgent made a profit.

4 Conclusions

In this report, we explained CautiousStdAgent's strategies that emphasize inventory control. These strategies enabled CautiousStdAgent to earn a higher score than any other agents.