SavingAgent: An agent submitted to the ANAC 2020 SCM league

Takuma Kawamura[1] , Takanobu Otsuka[2]
Nagoya Institute of Technology, Aichi, Japan
[1]takuma.kawamura@otsukalab.nitech,ac.jp[2]otsuka.takanobu@nitech.ac.jp

July 14, 2020

Abstract

We create an agent that aims to keep production costs as low as possible. This agent refers to the contracts that have been concluded, the inventory quantity of input, the cost of the production line, and then searches for the optimal production schedule. The created agent simulates in the SCML en-vironment. It makes a stable profit. This report describes methods can improve its performance

1 Introduction

According to the game description, in SCML2020, the production cost is calculated as follows:

- there is a set M of manufacturing processes
- each factory f ∈ F is characterized by a manufacturing profile that consists of a set of lines L_f together with a cost function
- $C_f : L_f \times M \to Z^{+}_{\infty}$which indicates the cost of running process m ∈ M on line l ∈ L_f.

The production cost is fixed for each line.Therefore, to make a larger profit, the highcost line needs to be saved rather than used. I focused on this and created an agent that stops production when a certain threshold is exceeded.

2 The Design of MyAgent

SavingAgent consists four classes,StepNegotiationManager, PredictionBasedTradingStrategy,My_ProductionStrategy and SCML2020Agent. This agent was created based on the DecentralizingAgent, but the " Production strategy" and "Negotiation Manager" have been replaced with ours.

2.1 Savingstrategy

In order to make more profit, it is necessary to keep production costs as low as possible. Therefore, in order to reduce the production cost, it is necessary not to use the line which is the highest production cost. Also, if a cheap production line is not used, it will make more money if it can be used. However, there are times when you have to use a line with a high production cost because a penalty will be incurred if the production is not in time for the contract already concluded. With these things in mind, SavingAgent has implemented the following algorithm:

1. Sort production lines in ascending order of production cost
2. The agent confirms the contract that has been concluded and schedules the production plan so that it will be as cheap as possible by the delivery date.
3. The agent check the production plan and own input products. If there are unused production lines then the agent reschedule to produce the output products on the production line as low as possible up to a certain threshold.
4. Do this for each step

By doing this algorithm, it is possible to suppress the production cost as much as possible and make a profit compared with the supplyDrivenStrategy.

3 Experiments

To evaluate the agent' s performance, we experimented with the run() functionpresent in the template. The parameters are follows:

- competition=std
- reveal_names=True

- n_steps=50
- n_runs_per_world=1

and DecentralizingAgent and BuyCheapSellExpensiveAgent were specified ascompetitors. The scores of each agent for the five experiments and their meansare shown in Table 1. Saving Agent and Decentralizing Agent had similar scores.Only BuyCheapSellExpensiveAgent scored a negative score. From this result,Saving Agent has shown stable results.

	DecentralizingAgent	Saving_Agent	BuyCheapSellExpensiveAgent
1	0.598526	0.501869	-1.67193
2	0.409514	0.439049	-0.64169
3	0.166714	0.132176	-1.64955
4	0.462424	0.373113	-0.146942
5	0.179613	0.129989	-1.76965
average	0.3633582	0.314779333	-0.958296

4 Conclusions

In this report, we described our agent, SavingAgent. SavingAgent can make its own profit by pushing as much as possible the cost it produces. With this agent, we set the threshold to three quarters and conducted an experiment。 The amount of increase in profit depends on the threshold value, so we have to find the optimal threshold value. Since this threshold may not be the optimum threshold, it is necessary to set the optimum threshold.