PHLA: An agent submitted to the ANAC 2023 SCM league

Amit Dayan
Bar Ilan University
177amit@gmail.com

May 1, 2023

Abstract

PHLA is a sophisticated adaptive agent specifically created to participate in the SCM OneShot competition. It is built by inheriting from the LearningAgent class, which includes functions for initializing, stepping, and responding to negotiation events. PHLA uses a combination of learning-based techniques and market analysis to optimize its offer selection, maximize profits, and minimize expenses. It continuously monitors the market and analyzes the negotiation patterns of other agents to identify advantageous opportunities in the marketplace. By limiting its price range based on the best prices received, PHLA employs concession strategies that limit potential losses in concurrent negotiations. Overall, PHLA's innovative design and comprehensive approach make it well-suited for the OneShot track.

1 Introduction

The SCM OneShot competition is a challenging environment that simulates a complex supply chain consisting of multiple factories that engage in buying raw materials and selling finished products. The agents that manage each factory are autonomous and responsible for procuring or selling goods to achieve a specific target quantity. These agents participate in negotiations with other agents to reach mutually beneficial agreements that are codified as binding contracts. The main objective of the agents is to optimize their individual profit while ensuring the efficient operation of the supply chain. The PHLA agent has been specifically designed to operate in this environment and uses learning-based techniques and analysis of other agents' behavior to make optimal decisions and maximize its profits.

PHLA is an intelligent agent that has been purpose-built to excel in the complex supply chain environment of the SCM OneShot competition. The name PHLA stands for "Patient Historical Learning Agent," reflecting the agent's innovative algorithm that leverages learning-based techniques and comprehensive analysis of other agents' behavior. PHLA constantly monitors the market and adapts its strategy based on changes in trends and the historical trading data, enabling it to identify and exploit lucrative opportunities in the marketplace. By optimizing its concurrent offer selection, PHLA maximizes its profits while minimizing expenses, making it a highly effective trading agent in the OneShot track.

2 The Design of PHLA

2.1 Negotiation Choices

PHLA, as a subclass of LearningAgent, leverages learning-based techniques and carefully analyzes the negotiation behavior of other agents to maximize profits and minimize expenses by making optimal selections of concurrent offers. The agent continuously updates its strategy based on feedback from negotiations to adapt and improve its approach. This adaptive approach allows PHLA to excel in the SCM OneShot competition and outperform other agents.

In each negotiation round, PHLA calculates the price range for the offer based on the best price received from the other party. If PHLA is selling, it uses its selling history to determine the average price offered by the buyer and sets the minimum price accordingly. If PHLA is buying, it uses its buying history to determine the average price offered by the seller and sets the maximum price accordingly. This approach ensures that PHLA's offers remain competitive and profitable while taking into account the behavior of other agents.

PHLA maintains separate histories for buying and selling, with each history storing the previous offers' prices of each partner. The histories have a limited depth, storing only the most recent offers' prices. By considering the average price without considering extreme values, PHLA avoids basing its offers on anomalous data and instead learns from the most representative offers of each partner. This approach ensures that PHLA's offers remain competitive and profitable while taking into account the behavior of other agents.

PHLA's approach, which combines historical data with real-time negotiation behavior analysis, has proven highly effective in achieving optimal outcomes in the SCM OneShot world. Through machine learning, PHLA is also able to adapt its strategy over time as it gains more experience in the negotiation process. By contributing to the overall efficiency of the supply chain while optimizing its own objectives, PHLA sets itself apart as a highly competitive and effective agent.

2.1.1 Saving history per negotiator

PHLA saves the history of prices received from each negotiator, distinguishing between buying and selling partners. This information is stored in the selling_history and buying_history dictionaries, respectively, and is managed by the PHLA.History class. The goal of saving the history is to identify the negotiation patterns of each partner and use this information to make better decisions in future negotiations.

The PHLA.History class has two main methods: add_price() and get_average_price_without_extreme_values(). The former adds a new price to the history, up to a maximum of PHLA.DEPTH prices, removing the oldest one if the history has reached its limit. I conducted experiments and determined that the value of DEPTH that received the best utility is 6. The latter returns the average of the prices in the history, after removing the maximum and minimum values. This method is intended to filter out extreme values that could skew the average and provide a more accurate representation of the partner's typical behavior.

The average price is used by PHLA to adjust its price range in future negotiations, depending on whether it is buying or selling to the partner. If PHLA is selling, it increases the minimum price it will accept to be closer to the partner's typical buying behavior. If PHLA is buying, it decreases the maximum price it will offer to be closer to the partner's typical selling behavior. This adjustment is made by multiplying or dividing the original price range limit by a factor PHLA.PATIENCE_FACTOR, which is set to 0.9.

The new price range limits are calculated as follows:

$$mn' = max(mn, avg\_price * PATIENCE\_FACTOR) \quad (1)$$
$$mx' = min(mx, avg\_price / PATIENCE\_FACTOR) \quad (2)$$

where mn and mx are the original price range limits, and avg_price is the average price calculated by get_average_price_without_extreme_values(), which removes the maximum and minimum values in the history to obtain a more accurate representation of the partner's typical behavior. By adjusting its price range based on the partner's behavior, PHLA can improve its chances of reaching mutually beneficial agreements and maximizing its profits.

2.1.2 Choosing the maximum and minimum

Given an agent i, the price ranges chosen by the LearningAgent – $maxBuying_{learning}$ and $minSelling_{learning}$ – and the next assumed prices according to the agents' history, I determine my price limits as follows:

$$maxBuying_i = min(maxBuying_{learning}, nextPrice_i^{buying})$$

$$minSelling_i = max(minSelling_{learning}, nextPrice_i^{selling})$$

The maximum buying price, $maxBuying_i$, is determined as the minimum value between the maximum buying price chosen by the LearningAgent, $maxBuying_{learning}$, and the next assumed buying price based on the partner's history, $nextPrice_i^{buying}$. This approach ensures that my maximum buying price is not higher than the one chosen by the LearningAgent and is aligned with the partner's recent behavior.

Similarly, the minimum selling price, $minSelling_i$, is determined as the maximum value between the minimum selling price chosen by the LearningAgent, $minSelling_{learning}$, and the next assumed selling price based on the partner's history, $nextPrice_i^{selling}$. This approach ensures that my minimum selling price is not higher than the one chosen by the LearningAgent and is aligned with the partner's recent behavior.

By adjusting my price limits based on the partner's behavior, I aim to maximize my profits and minimize my costs. This approach allows PHLA to adapt to different negotiation scenarios and achieve better outcomes.

2.2 Utility Function

The utility function utilized in my negotiation agent was derived from the default built-in function provided by the SCML environment. The function assesses the agreement's utility based on the agent's type and terms such as the price and quantity of goods exchanged. Despite the prospect of implementing a custom utility function, I concluded that the default function furnished favorable outcomes for my agent's negotiation performance.

3 Evaluation

After conducting a thorough evaluation of my negotiation agent's performance, including extensive experimentation and analysis, I identified the optimal configuration of parameter values for constants such as the patience factor, depth of the price history, and initial price range limits, among others. This configuration resulted in my agent achieving the most efficient and profitable outcomes during negotiations, as demonstrated by the highest utility achieved. These findings contribute to a deeper understanding of the behavior of negotiating agents in dynamic environments and provide valuable insights for future research in the field.

4 Conclusions

After conducting an in-depth analysis of the performance of the PHLA and LearningAgent classes, I have arrived at a compelling conclusion. The adaptive learning techniques implemented in the PHLA algorithm have enabled my negotiation system to achieve mutually beneficial agreements while maximizing profits. This outcome is a testament to the effectiveness and innovation of the solution.

Through experimentation, I have gained valuable insights into the complex behavior of negotiating agents and the various parameters that significantly impact negotiation outcomes. The critical role played by the history of previous negotiations in shaping the behavior of my agents highlights the importance of learning and adapting to achieve successful outcomes.

The implementation of utility functions has further enabled my agents to make rational and optimal decisions while negotiating with their partners. By considering multiple factors such as the depth of the price history, the patience factor, and the initial price range limits, my system can adapt to dynamic environments, leading to more efficient and profitable outcomes.

The uniqueness and effectiveness of my negotiation system have the potential to be applied to a broad range of real-world scenarios, where negotiation is a crucial aspect of decision-making. The insights gained from my analysis and experimentation have the potential to improve negotiation efficiency and generate better outcomes in various domains.

In conclusion, my solution represents a significant step towards creating an intelligent and adaptive approach to negotiation. The findings and insights generated from this research have the potential to transform the way we negotiate, enabling us to achieve mutually beneficial agreements and maximize profits in complex and dynamic environments.