AN AGENT-BASED NEGOTIATION APPROACH FOR SUPPLY CHAINS

A Thesis

by

Celal Ozan Berk Yavuz

Submitted to the
Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in the
Department of Computer Science

Özyeğin University
August 2020

AN AGENT-BASED NEGOTIATION APPROACH FOR SUPPLY CHAINS

Approved by:

Assistant Professor Reyhan Aydoğan
Advisor
Department of Computer Science
Özyeğin University

Assistant Professor İsmail Arı
Department of Computer Science
Özyeğin University

Assistant Professor Ömer Korçak
Department of Computer Engineering
Marmara University

Date Approved: 27 August 2020

To Mom and Dad

ABSTRACT

A supply chain is a network enabling the flow of goods from suppliers to end customers. In a supply chain, suppliers, manufacturers, distributors and retailers interact with each other to buy, process, store and sell products. In an efficient supply chain the main aim is to minimize the costs while maximizing the profits. To achieve this, the supply chain entities interact with each other wisely. For example, they need to agree on contracts maximizing their profits but not causing infeasibilities in the delivery or payment. In this thesis we worked on a supply chain simulation, namely the supply chain world, which has been used in Supply Chain Management League (SCML) of Automated Negotiating Agents Competition (ANAC), an international supply chain competition. In the supply chain world simulation environment, we proposed a factory manager agent which can procure products from other agents, processing some of them, and sells it with profit to other agents. The interactions with other agents are done in closed bilateral negotiations protocol. The motivation of the proposed agent is determining a good level of altruism and greediness in negotiations depending on the supply and demands of products. To evaluate the performance of the proposed agent strategy, we run a number of simulations and compare the performance of our agent with the performance of the top performing agents in the SCML 2019. The experiment results show that our agent, MyGreedyAgent, managed to adjust its greediness and altruism effectively based on the supply demand ratio and performed as well as ANAC 2019 top agents.

ÖZETÇE

Tedarik zinciri, ürünlerin tedarikçilerden tüketicilere geçişini belirten bir ağdır. Tedarik zincirlerinde tedarikçiler, imalatçılar, distribütörler, ve perakendeciler birbiriyle etkileşime girerek ürün alır, ürünleri işler, depolar ve satar. İyi işleyen tedarik zincirlerinde maliyetlerin azaltılması ve toplam kârın artırılması amaçlanır. Tedarik zinciri bileşenlerinin bu amaca ulaşabilmesi için birbiriyle stratejik olarak etkileşimde bulunmalıdır. Örneğin başka tedarik zinciri bileşenleriyle varılan anlaşmaların ürün teslimatı ve ödeme bakımımdan olanaklı olmasına dikkat edilmelidir. Olanaklı anlaşmalar yaparken aynı zamanda kârın maksimize edilmesi de amaçlanmalıdır. Bu tezde, ANAC (International Automated Negotiating Agents Competition)'da yıllık yarışması yapılan "tedarik zinciri dünyası" adlı bir tedarik zinciri simülasyonu üzerinde çalıştırılmıştır. Çalışmada bir imalatçı etmen geliştirilmiştir. Geliştirilen etmen, simülasyondaki diğer etmenlerden ürün alıp, bazılarını işleyip başka etmenlere kâr yaparak satabilmektedir. Geliştirilen etmen, diğer etmenlerle ikili pazarlık protokolüne göre etkileşimde bulunmaktadır. Etmenin temel motivasyonu, ürünlerin arz ve talebine göre pazarlıkta ne kadar bencil veya cömert davranması gerektiğine karar verebilmektir. Geliştirilen etmenin performansı SCML 2019'daki en iyi çalışan etmenlerle yarıştırıp karşılaştırıldı. Test sonuçları, etmenin etkili bir şekilde ne kadar bencil veya cömert davranması gerektiğine karar verebildiğini ve yarışmadaki çoğu etmenden daha fazla kâr yapabildiğini gösterdi.

ACKNOWLEDGEMENTS

I would like to express my thanks and appreciation to people who helped me with my thesis.

Firstly, we sincerely appreciate my thesis advisor assistant professor Reyhan Aydoğan for the guidance on my studies, feedbacks for my thesis and publications we worked on.

Also, we would like to express my appreciation for Yasser Mohammed, the creator of the framework we studied in my thesis, he helped me with usage of the framework, and provided feedback on my work in this thesis. We appreciate his responsiveness for the questions we asked about the framework.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS/ABBREVIATIONS

β Amount of output products products procured.

$_p$ Amount of product p procured by negotiations

η Amount of output products produced

η_p Amount of product p produced in factory.

$\gamma\_input(\omega)$ Supply demand cost function of the input product

$\gamma\_output(\omega)$ Supply demand cost function of the output product

γ_{input} Supply demand cost of the input product

γ_{output} Supply demand cost of the output product

κ_p Weighted average price of procurement for product p

μ_i The average production cost of the product type i using the product type i-1.

Ω The weight of the opponent's estimated utility to calculate the overall utility.

ω Amount of products to be scheduled for production

ϕ Unit production cost

$_p$ Total amount of money spent to procure product p.

a Amount of products delivered by the perpetrator.

a' Amount of products supposed to be delivered by the perpetrator according to the contract terms.

a_i Amount of input products in the inventory.

avg_{bp_i} Average buying price of the input product

avg_{bp_o} Average cost of procurement for the output product

avg_{oc_o} Average cost of obtaining the output product.

avg_{sp_i} Average selling price of the input product

avg_{sp_o} Average selling price of the output product

b The range penalty for for undelivered goods

c CFP

c_i Catalog price of the product type i.

c_{price} Catalog price of the product p

d Average demand per time step

d The range of delivery times

d_o Average demand for the output product per time step

d_i Average demand per step of the input product

f Amount of products delivered by the perpetrator.

f Final balance at the end of the simulation run

f' Amount of products supposed to be delivered by the perpetrator according to the contract terms.

i Initial balance at the beginning of the simulation run

i_c Binary variable indicating if the contract c is signed.

k Binary variable indicating if there is a shortage.

l Breach level of a breached contract

m_p Mispurchasing cost

$Overall(o)$ The overall utility.

P Perpetrator of a breached contract

p The range of product types

q The range of quantities

q_o Quantity of the product p in the offer.

r_t Reservation value at time step t.

s Score

s_o Average supply for the output product per time step

s_p Unit price of p in the offer.

s_i Average supply per step of the input product

u The range of unit prices

$U_B(o)$ Utility value of a bid in buying negotiations

$U_O(o)$ Estimated opponent utility function.

$U_S(o)$ The utility function for our agent ,without considering the opponent's utility.

v Victim of a breached contract

V_p Fair price of the product p.

CHAPTER I

INTRODUCTION

A supply chain is a network illustrating the flow of goods from the suppliers to end customers. In the supply chain, the supply chain entities, namely suppliers, manufacturers, wholesalers and distributors interact with each other to make collaborative business decisions. In an efficient supply chain the costs of product procurement, logistics and storage costs are aimed to be minimized while the profits are to be maximized. The process of optimizing a supply chain is called Supply chain management. There are many challenges in supply chain management such as uncertainty of demands and supplies, changing markets, production planning, cooperative decision making with other supply chain entities and so forth [1]. Due to the importance of supply chain management in business, supply chain management is an extensively researched study area.

Recently researchers worked on multi-agent systems based supply chain simulation to automate the negotiations between supply chain entities such as factory managers, and consumers. To address the issues mentioned, and encourage more researches in this field, the International Automated Negotiating Agents Competition (ANAC), introduced a competition in which researchers can design a factory manager agent which aims to maximize its profit and can interact with other agents to procure and sell goods.

In this thesis, we present our proposed factory agent designed and implemented in a supply chain simulation, namely, supply chain world [2]. The supply chain world competition is held in ANAC, which is a competition in which multi agent researchers submit their agents to compete with other participants. In the supply chain world, there are supply chain entities miners, factory managers and customers. The supply chain entities are represented as intelligent agents which can interact with each other by involving bilateral negotiations. In these negotiations, one party act as the buyer agent, which is negotiating to buy goods from the other party, the seller agent. The goal in the supply chain is to design a factory manager agent which procure goods from other agents, process them in their factory and sell the to other agents with profit. Ultimately, the agent with the highest profit wins.

The challenges are met to design the agents are,

CFP posting policy: CFP stands for call for proposals which expresses an agent's interest in a potential negotiation under the specified negotiation issues. A factory manager agent needs to determine the values of the negotiable issues product type, quantity, unit price, and delivery time to post as CFP.

Deciding what CFP's to request a negotiation: A factory manager agent needs to decide on what CFPs on the bulletin board to request negotiation.

Responding to negotiation requests: A factory manager agent needs to decide on what negotiations requests from other agents to accept.

Scheduling production: A factory manager agent needs to decide the production schedule in its factory. That is, to decide what products to produce at what time steps with the right quantity.

Constructing negotiators: To negotiate with other agents, a factory manager agent needs to construct negotiator agents, which handle the negotiation process on behalf of its agent. The factory manager agent forms the negotiation policy of the negotiator by determining it's utility function, bidding strategy, acceptance strategy, and ideally an opponent modelling strategy to prognosticate the preference of other party to avoid unfortunate moves [3]. The factory manager agent can run concurrent negotiators with many agents by creating many negotiators negotiating in separate negotiation threads, and can alter the negotiation policy of a negotiator in a negotiation thread based on bids received from another negotiation threads.

Contract signing: When a negotiation between two agents is successful, the bid that agents agree on becomes a contract between two parties. In order a contract to become binding, both parties are required to sign the contract. The contract is nullified otherwise. Therefore, factory manager agents decide on what contracts to sign or not sign wisely in order to avoid contract violations. Also, due to scarcity of resources i.e. funds and products in the inventory an efficient, allocation of resources is needed to maximize profit.

Our motivation in this study is designing a factory manager agent which effectively determines a good level of greediness and altruism in negotiations depending on the supplies and demands of the products. We have tested our agent against the top performing agents in ANAC 2019 supply chain management league (SCML). The experiment results showed that it effectively determined how greedy or altruistic it is supposed to act based on the supply and demand ratio of the goods and it performed as well as the top performing agents in ANAC 2019 SCML.

CHAPTER II

BACKGROUND

In this thesis we have worked on a supply chain simulation environment, namely supply chain management world [2], in which there are suppliers, factory managers and customers. Suppliers provide raw materials to factory managers who can process raw materials and produce intermediate products, the products which can be processed further by other factory manager agents. Customer agents buy end products which can't be processed further by factory managers.

This thesis aims to design a factory manager agent which procures goods and sell it with maximum profit. The procurement and selling of goods were done based on agreements reached by bilateral negotiations with other agents.

2.1 Simulation Environment Description

In this section first, we will describe the supply chain entities. Then, we mention the simulation mechanism rules. Finally, we will give an example simulation run which illustrates interactions between the agents.

2.1.1 Supply Chain Entities

The supply chain consists of different types of agents and the bulletin board as illustrated in figure 1. The agents can interact with other agents and can reach public information via bulletin board.

Agents

The supply chain simulation consists of supplier agents, factory manager agents, customer agents. The detailed descriptions of each of those are as follows:

Supplier agents: Supplier agents sell raw materials to factory manager agents.

During negotiations with factory manager agents, their utility function incentivizes them to sell maximum amount of products at maximum price in as sooner as possible. Supplier agents are designed in that way so as to maximize the supply chain throughput. They act as sellers.

Factory manager agents: Factory manager agents buy raw materials and/or intermediate products, which are processed by other factory manager agents. They have a factory in which they can process products to produce other products. The output products are called final products, which can be sold to customers and other factory manager agents. The factory manager agents provided by the simulation environment are called greedy factory manager agents whose utility function considers the difference between their current balance and the estimated balance after their agreement, assuming that the other party will be in compliance to the contract terms, i.e the contract will not be violated by the other party. Factory manager agents can both act as sellers and buyers.

Consumer agents: Consumer agents form the last level of the supply chain, i.e they are the end users. They have a dynamic schedule provided by the simulation environment and their utility function encourages the agent to buy products according to the schedule with a minimum price. Therefore, overconsumption and underconsumption are penalized by the utility function. They act as buyers.

The diagram below illustrates how the agents interact with each other in the simulation environment.

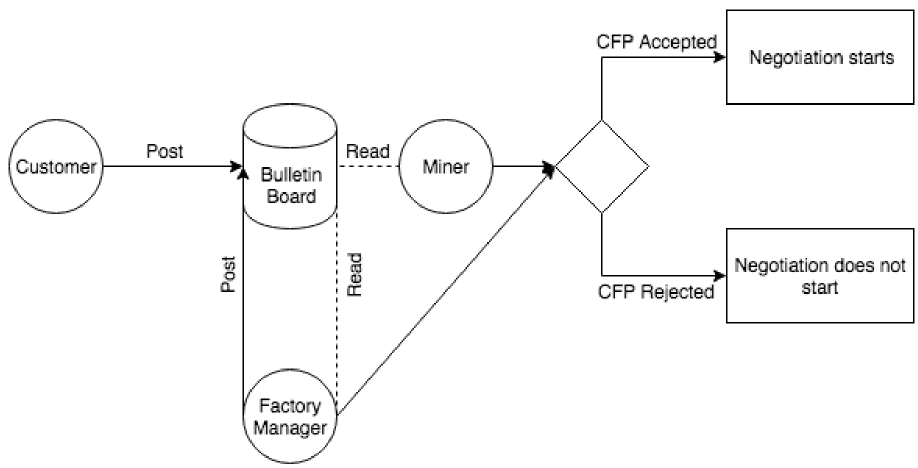

Figure 1: Agent Interactions

2.1.2 Bulletin Board

Bulletin board stores public information accessible to all agents. Public information in the board are listed as follows:

Breach reports: Breach reports indicate a contract violation between two parties. A breach report consists of perpetrator, victim, and breach level indicating the severity of breach. The breaches may arise when the buyer party fails to pay the fee for the products or the seller party fail to deliver the products.

Call for proposals: Call for proposals (CFP) are posted by factory manager agents and customers to propose negotiation terms, which are the range of issues to be negotiated. A CFP can be represented as a tuple shown below:

$$c = (p, d, u, q, b) \tag{1}$$

where,

c : CFP

p : The range of product types

d : The range of delivery times

u : The range of unit prices

q : The range of quantities

b : The range penalty for for undelivered goods (Optional).

To illustrate, if a factory manager agent posts a CFP c=(1, (2,4), (3,9), (1,3), 30), the factory manager agent is interested in a negotiation under the negotiation issues p=1, d=2,3,4, u=3,4,5,6,7,8,9, q=1,2,3, b=30. A possible bid for that CFP could be b=(1, 3, 7, 2, 10), meaning that the the agent proposed this bid in the negotiation would like to receive a product of type 1, at the time step 3, with the unit price of 7 (the agent will pay 7 units of of funds per product). Note that the publisher of the CFP always acts as a buyer and the agent who requests a negotiation for the CFP is willing to sell the product, i.e. the publisher receives a product 1 at time step 3 and will pay 7 units of funds to the requester which will deliver the product and get the payment. Note that the b refers to the negotiated penalty, which states the amount of penalty that the seller pays to the buyer in case the seller can't deliver all goods on the agreed time slot. To illustrate, assume that the bid b = (1,3,7,2,10) was agreed on between the buyer and seller agents and they signed a contract for the agreement. The buyer party is liable to deliver 2 units of product of type 1. Otherwise, an insufficient delivery of products would be reported for the seller party and the seller party would have to pay 10 units of funds for each product of type 1 undelivered to the buyer party.

Quarterly reports: At the end of each quarter of the maximum time steps of a simulation run, (i.e if the simulation ends at the time step 100, the quarterly reports will be posted at the time steps 25, 50, 75 and 100) quarterly reports are published. A quarterly report for each agent is published to the bulletin board. The quarterly report includes a snapshot information for the number of items in inventory, balance, and liability.

2.2 Simulation Mechanism

A simulation run consists of three different phases namely, pre-negotiation, negotiation and the post negotiation phase. At the end of the last phase, the simulation state is updated and the simulation moves forward to the next time step. The simulation phases are explained in detail in the following sections.

2.2.1 Pre-Negotiation Phase

This is the beginning phase of a simulation time step during which the agents find their negotiation partners. At this step, each buyer party publishes CFP to the bulletin board to specify what products they are interested in and under which conditions. That determines the bid spaces for a potential negotiation. The other party who is interested in selling the products under the negotiation issues defined in a specific CFP would request a negotiation to the publisher of that CFP. If the request is accepted by the publisher, a negotiation under the corresponding issues takes place. Note that the agents can post and request negotiation by multiple CFPs. Because the supplier agents act only as sellers, they never post CFPs but request negotiations to the CFPs they are willing to negotiate under its issues. Factory manager agents on the other hand can act as buyers and sellers. Thus, they can act as sellers and buyers (i.e., they can both publish or request negotiation to CFPs). The customer agents only buy products and therefore, they do not request negotiations but post CFPs only. Note that each agent can post and request negotiation to multiple CFPs. The phase ends when all agents posted their CFPs and the other agents decided what CFP or CFPs they will request a negotiation to.

2.2.2 Negotiation Phase

It is the next step where closed negotiations take place. The negotiations between the agents are done in a bilateral fashion i.e. where two agents negotiate by exchanging offers until an agreement is reached or negotiation deadline is expired. At the beginning of a negotiation, one of the randomly chosen parties makes an initial offer and the other party can accept the offer or make a counter offer i.e. it comes up with its own offer. If the initial offer is accepted, the negotiation ends with success; otherwise the negotiation moves forward to another round in which the counter offer of other party is evaluated. This process is repeated until one of the parties accepts the other party's offer or the negotiation deadline, 20 rounds, is passed. Any negotiation with an agreement is called as a successful negotiation in which the agreed offer becomes a contract. Note that the agents can make bilateral concurrent negotiations in case there was multiple CFP agreements involving the same agents. For instance assume that the Agent A posted three CFPs in the pre-negotiation phase, and got 4 requests from the agents B,C,D and E. If the Agent A accepts the requests of agents C and E. It would negotiate with the agents C and E independently in different negotiation threads in which the offers and negotiation issues can differ. Furthermore, the negotiations threads are private i.e. the Agent C wouldn't be able to access the negotiation information in the negotiation thread where the Agent A and E would be negotiating. This phase ends after all agents finish their negotiations with other agents.

2.2.3 Post Negotiation Phase

This is the final phase of a negotiation time step in which agents make a decision to sign the contracts from the agreements in the preceding negotiation phase. If at least one of the parties refuse to sign the contract, the contract is nullified and no party is expected to comply with the terms of the contract. If both parties sign a contract, it becomes a binding contract, and the agents are supposed to comply with the terms. If the one of the parties dishonors a binding contract, a contract breach is reported to a bulletin board. A breach report contains the information the breached contract, perpetrator, victim and breach level which indicates at what degree, the contract is violated by the perpetrator. The breach reports are publicly accessible to all agents in the simulation i.e the agents can read the reports and get some idea of the credibility of their partner. For example, an agent which violates many contracts consistently at a high breach level might cause a distrust on other agents, and may affect their policies, such as acceptance of the further negotiation requests, signing contract decisions with the other agents and more, depending on how the other agents are implemented. This phase ends after all of the agents make signing decisions for the contracts they are issued in. Before moving to the next time step, the simulation environment state is updated, i.e, the binding contracts due current time step are executed in which the seller parties get the payment and deliver goods to the buyer parties for the contracts without a breach, if there are breaches, the breach reports are posted to the bulletin board. Finally, the factory manager agents' production states are updated. For instance, if an agent has scheduled a production in the preceding time step, and the production ends in the current time step, the products produced are added to the inventory and the production costs are deducted from the balance.

A simulation run ends when a certain number of time steps, called simulation length, have passed. At the end of the simulation run, the performance of the factory manager agents are measured by the following:

$$s = \frac{i - f}{i} \tag{2}$$

where,

f : Final balance at the end of the simulation run

i : Initial balance at the beginning of the simulation run

Informally, the final score of a factory agent is equal to change in the initial price divided by the initial price. For instance if an agent doubles of its initial balance at the end of the simulation, its score would be 2. Bankrupted agents get negative score, and the winner is the factory agent, which has the highest positive score at the end of the simulation.

2.2.4 An Example Simulation Run

Here is an example simulation run with a supplier Agent A, 2 factory manager agents B and C and a customer Agent D. In this simplified example simulation run, there is only one raw material type, called product 0, the factory manager Agent B can process the product 0 in its factory to produce the intermediate product called product 1. The production process requires one unit of product 0 to produce one unit of product 1, which takes one time step and the production cost associated with this process is 2 per product 0 processed. The factory manager Agent C can produce one product 2 by processing one product 1, the production process takes 1 time step and the production cost is 4 per unit. The factory manager agents B and C starts with 1000 units of funds, with zero inventory. The initial simulation state is illustrated as below.

Table 1: Initial State of the Example Simulation Run.

Agent \ Stats	Product 0	Product 1	Product 2	Funds
Agent B	0	0	0	1000
Agent C	0	0	0	1000

In this example the simulation length is 4, meaning the simulation ends in the time step 4. The detailed explanation of the phases in each steps is show below.

Time step 1:

Pre-negotiation phase

The factory manager agents post the CFPs below:

- Agent B posts the following CFPs: c1=(0, (2, 3), (3, 8), (1, 10), 30), c2=(0, 2, (1, 3), (1, 2), 30), c3=(0, 2, (1, 12), (1, 6), 30), c4=(0, (2, 3), (1, 4), (1, 15), 30), c5=(0, (2, 3), (1, 4), (1, 9), 30), c6=(2, 2, (1, 4), (1, 23), 30).

- Agent C posts the following CFPs: c7=(1, (2, 3), (3, 8), (1, 10), 30), c8=(1, 2, (1, 3), (1, 2), 30), c9=(1, 2, (1, 12), (1, 6), 30), c10=(1, (2, 3), (1, 4), (1, 15), 30), c11=(1, (2, 3), (1, 4), (1, 9), 30).

- Agent D posts the following CFPs: c12=(2, (2, 3), (2, 12), (1, 18), 30), c13=(2, (2, 3), (1, 10), (1, 21), 30).

Note that the CFPs are represented as in Equation 5. To illustrate the publisher of the CFP c1=(0, (2, 3), (3, 8), (1, 10), 30) is interested in buying product 0 where the negotiation issues are shown below:

- Product type: 0

- Delivery time: Ranges between 2 and 3. The negotiation issues values are {2, 3} for the delivery time.

- Unit price: Ranges between 3 and 8, the negotiation issue values are {3, 4, 5, 6, 7, 8}

- Quantity: Ranges between 1 and 10, the negotiation issue values are {1, 2, 3, 4, 6, 7, 8, 9, 10}.

An example offer in the negotiation under the CFP c1=(0, (2, 3), (3, 8), (1, 10), 30) can be represented as a tuple containing all negotiation issues with the values specified between the issue ranges. For example, b1 = (0, 2, 4, 7, 30) would be a valid bid under the CFP c1, while b2 = (0, 2, 9, 2, 30) wouldn't. The agents who are interested to sell products to the publishers of CFPs post the following CFPs:

- Agent A requests a negotiation the CFPs c1, c4, c5, the publisher of these CFPs (Agent B) accepts the requests to the CFPs c1 and c4 and rejects to do negotiation for the CFP c5. Agent A will negotiate with Agent B for the CFPs c1 and c4 only, the negotiation threads for these CFPs c1 and c4 are referred as negotiation thread 1, and negotiation thread 2 respectively.

- Agent B requests a negotiation the CFPs c7, c10, c11, the publisher of these CFPs (Agent C) accepts the requests to the CFPs c7 and c11 and rejects to do negotiation for the CFP c10. Agent B will negotiate with Agent C for the CFPs c7 and c11 only, the negotiation threads for the CFPs c7 and c11 are referred as negotiation thread 3, and negotiation thread 4 respectively.

- Agent C requests a negotiation regarding the CFPs c12, c13, the publisher of these CFPs (Agent D) accepts the requests to the CFPs c12 and c13. Agent C will negotiate with Agent D for the CFPs c12 and c13, the negotiation threads for these CFPs are referred as negotiation thread 5, and negotiation thread 6 respectively.

The pre-negotiation phase ends. In the next phase, namely the negotiation phase, the agents will do negotiations on the CFPs for which there is an agreement.(i.e for the ones when both parties agreed on).

Negotiation phase:

Negotiation thread 1: The negotiation issues under this negotiation thread are determined by the CFP c1=(0, (2, 3), (3, 8), (1, 10), 30), which both Agent A and Agent B agreed on. Agent A is chosen as the agent which will make the initial offer.

- Round 1: Agent A offers the bid b1 = (0, 2, 3, 10, 30), and Agent B counter offers with the bid b2 = (0, 3, 1, 1, 30).

- Round 2: Agent A counter offers with the bid b3 = (0, 2, 3, 9, 30), and Agent B counter offers with the bid b4 = (0, 2, 2, 1, 30).

- Round 3: Agent A counter offers with the bid b5 = (0, 2, 2, 8, 30), and Agent B counter offers with the bid b6 = (0, 2, 1, 3, 30).

- Round 4: Agent A counter offers with the bid b7 = (0, 3, 2, 8, 30), and Agent B counter offers with the bid b8 = (0, 2, 3, 7, 30).

- Round 5: Agent A accepts Agent B's offer. The opponent's bid b8 = (0, 2, 3, 7, 30) will be a contract to be signed. The contract will be referred as contract 1.

Negotiation thread 2: The negotiation issues under this negotiation thread are determined by the CFP c4=(0, (2, 3), (1, 4), (1, 15), 30), which both Agent A and Agent B agreed on. Agent B is chosen as the agent which will make the initial offer.

- Round 1: Agent B offers the bid b1 = (0, 2, 1, 1, 30), and Agent A counter offers with the bid b2 = (0, 2, 4, 15, 30)

- Round 2: Agent B counter offers with the bid b3 = (0, 3, 4, 12, 30), and Agent A counter offers with the bid b4 = (0, 2, 2, 13, 30)

- Round 3: Agent B counter offers with the bid b5 = (0, 2, 4, 10, 30), and Agent A counter offers with the bid b6 = (0, 3, 1, 11, 30)

- Round 4: Agent B counter offers with the bid b7 = (0, 3, 4, 7, 30), and Agent A counter offers with the bid b8 = (0, 2, 4, 8, 30)

- Round 5: Agent B accepts the opponent's offer. The opponent's bid b8 = (0, 2, 4, 8, 30) will be a contract to be signed. The contract will be referred as contract 2.

Negotiation thread 3: The negotiation issues under this negotiation thread are determined by the CFP c7=(1, (2, 3), (3, 8), (1, 10), 30), which both Agent B and Agent C agreed on. Agent B is chosen as the agent which will make the initial offer.

- Round 1: Agent B offers the bid b1 = (1, 2, 10, 15, 30), agent Agent C counter offers with the bid b2 = (1, 3, 4, 1, 30)

- Round 2: Agent B counter offers with the bid b3 = (1, 3, 10, 15, 30), and Agent C counter offers with the bid b4 = (1, 3, 2, 13, 30)

- Round 3: Agent B counter offers with the bid b5 = (1, 3, 6, 11, 30), and Agent C counter offers with the bid b6 = (1, 2, 3, 10, 30)

- Round 4: Agent B counter offers with the bid b7 = (1, 3, 4, 8, 30), and Agent C counter offers with the bid b8 = (1, 3, 4, 8, 30)

- Round 5: Agent B accepts the opponent's offer. The opponent's bid b8 = (1, 3, 4, 8, 30) will be a contract to be signed. The contract will be referred as contract 3.

Negotiation thread 4: The negotiation issues under this negotiation thread are determined by the CFP c11 = (1, (2, 3), (1, 4), (1, 9), 30), which both Agent B and Agent C agreed on.

Agent B is chosen as the agent which will make the initial offer.

- Round 1: Agent B offers the bid b1 = (1, 2, 9, 10, 30), and Agent C counter offers with the bid b2 = (1, 3, 10, 10, 30)

- Round 2: Agent B counter offers with the bid b3 = (1, 3, 3, 7, 30) ,and Agent C accepts the Agent B's bid b3. The opponents bid will be a contract to be signed. The contract will be referred as contract 4.

Negotiation thread 5: The negotiation issues under this negotiation thread are determined by the CFP c12 = (2, (2, 3), (2, 12), (1, 18), 30), which both the Agent D and C agreed on. Agent D is chosen as the agent which will make the initial offer.

- Round 1: Agent D offers the bid b1 = (2, 2, 2, 18, 30), and Agent C counter offers with the bid b2 = (2, 3, 12, 10, 30)
- Round 2: Agent D counter offers with the bid b3 = (2, 2, 7, 11, 30), and Agent C counter offers with the bid b4 = (2, 3, 12, 10, 30)
- Round 3: Agent D counter offers with the bid b5 = (2, 2, 10, 11, 30), and Agent C counter offers with the bid b6 = (2, 3, 11, 10, 30)
- Round 4: Agent D counter offers with the bid b7 = (2, 3, 10, 4, 30), and Agent C accepts the Agent D's bid b7. The opponents bid will be a contract to be signed. The contract will be referred as contract 5.

Negotiation thread 6: The negotiation issues under this negotiation thread are determined by the CFP c13 = (2, (2, 3), (1, 10), (1, 21), 30), which both the Agent D and C agreed on. Agent C is chosen as the agent which will make the initial offer.

- Round 1: Agent C offers the bid b1 = (2, 3, 10, 21, 30), and Agent D counter offers with the bid b2 = (2, 2, 1, 21, 30)
- Round 2: Agent C counter offers with the bid b3 = (2, 2, 7, 11, 30), and Agent D counter offers with the bid b4 = (2, 2, 1, 21, 30)
- Round 3: Agent C counter offers with the bid b5 = (2, 2, 10, 11, 30), and Agent D counter offers with the bid b6 = (2, 2, 1, 21, 30)
- Round 4: Agent C counter offers with the bid b7 = (2, 3, 10, 4, 30), and Agent D counter offers with the bid b8 = (2, 2, 1, 21, 30)
- Round 5: Agent C counter offers with the bid b9 = (2, 3, 10, 4, 30), and Agent D counter offers with the bid b10 = (2, 2, 1, 21, 30) Negotiation deadline expires, i.e. the negotiation fails. Therefore, no contract will be signed for this negotiation.

The negotiation phase ends, because all the negotiation outcomes are reached. All the negotiations except the one in the negotiation thread 6 ended with an agreement, meaning that the negotiating parties will decide to sign a contract for the agreements in the next phase.

Post Negotiation Phase:

Simulation State Update

Before making a signing decision, the simulation state is normally updated, i.e. the contracts due this time step is executed and the preceding production processes in the factories finish. However, there are currently no contracts due this time step and no preceding production processes in factories. There won't be a state update for this time step.

Contract Signing

- Contract 1: Both Agent A and Agent B signs the contract and the contract becomes binding. Therefore, Agent A will deliver 7 units of product 0 to Agent B at the time step 2 and Agent A will pay 21 units of funds to Agent B.

- Contract 2: Both Agent A and Agent B signs the contract and the contract becomes binding. Therefore, Agent A will deliver 8 units of product 0 to Agent B at the time step 2 and the Agent A will pay 32 units of funds to Agent B.

- Contract 3: Both Agent B and Agent C signs the contract. Therefore, the Agent B will deliver 8 units of product 1 to Agent C at the time step 2 and the Agent C will pay 32 units of funds to Agent B.

- Contract 4: Both Agent B and Agent C signs the contract. Therefore, the Agent B will deliver 7 units of product 1 to the Agent C at the time step 3 and the Agent C will pay 21 units of funds to Agent B.

- Contract 5: Agent C chooses not to sign the contract. Therefore, the contract is nullified. Neither Agent C nor Agent D is expected to comply with this contract.

After the contract executions and contract signings end, the factory manager agents can schedule production in their factories. However, because they do not have any products in their inventory yet, they cannot schedule for production for this time step.

Time step 2:

Pre-negotiation phase The factory manager agents post the CFPs shown below:

- Agent B posts the following CFPs: c1 = (0, (3, 4), (3, 10), (1, 10), 30), c2 = (0, 3, (1, 3), (1, 2), 30), c3 = (0, 3, (1, 12), (1, 6), 30), c4 = (0, (3, 4), 2, (1, 15), 30), c5 = (0, (3, 4), (3, 4), (1, 9), 70), c6 = (2, 2, (1, 4), (1, 23), 100).
- Agent C posts the following CFPs: c7 = (1, 3, (3, 8), (1, 10), 90), c8 = (1, 3, (1, 15), (1, 10), 30), c9 = (1, 3, (1, 12), (1, 6), 30), c10 = (1, (3, 4), (1, 4), (1, 15), 30)
- Agent D posts the following CFPs: c11 = (2, (3, 4), (2, 12), (1, 18), 180), c12 = (2, 4, (1, 10), (1, 21), 190).

The agents who are interested to sell products to the publishers of CFPs send negotiation request to the CFPs:

- Agent A requests a negotiation the CFPs c1, c2, c3, c4, c5, c6. The publisher of these CFPs (Agent B) accepts the requests to the CFPs c1 and c2 and. and rejects to do negotiation for the CFPs c3, c4, c5, and c6. Agent A will negotiate with Agent B for the CFPs c1 and c2 only, the negotiation threads for these CFPs c1 and c2 are referred as negotiation thread 1, and negotiation thread 2 respectively.

- Agent B requests a negotiation to the CFPs c7, c8, and c9. The negotiation requests for c7 and c8 is accepted by the publisher of these CFPs (Agent C). The negotiation threads in which the negotiations for the CFPs c7, c8 will take place will be referred as the negotiation threads 3 and 4 respectively.

- Agent C requests a negotiation to the CFPs c11, c12, the publisher of these CFPs (Agent D) accepts the requests to the CFPs c11 and c12. The negotiation threads for these CFPs are referred as negotiation thread 5, and negotiation thread 6 respectively.

Negotiation phase:

Negotiation thread 1: The negotiation issues under this negotiation thread are determined by the CFP c1 = (0, (3, 4), (3, 10), (1, 10), 30), which both Agent A and Agent B agreed on. Agent B is chosen as the agent which will make the initial offer.

- Round 1: Agent B offers the bid b1 = (0, 3, 3, 1, 30), and Agent A counter offers with the bid b2 = (0, 3, 10, 10, 30)

- Round 2: Agent B counter offers with the bid b3 = (0, 3, 7, 11, 30), and Agent A counter offers with the bid b4 = (0, 3, 10, 10, 30)

- Round 3: Agent B counter offers with the bid b5 = (0, 3, 10, 11, 30), and Agent A counter offers with the bid b6 = (0, 3, 7, 9, 30)

- Round 4: Agent B counter offers with the bid b7 = (0, 3, 10, 4, 30), and Agent A counter offers with the bid b8 = (0, 3, 6, 10, 30)

- Round 5: Agent B counter offers with the bid b9 = (0, 3, 10, 4, 30), and Agent A counter offers with the bid b10 = (0, 3, 5, 10, 30) Negotiation deadline expires, i.e. the negotiation fails. Therefore, no contract will be signed for this negotiation.

Negotiation thread 2: The negotiation issues under this negotiation thread are determined by the CFP c2 = (0, 3, (1, 3), (1, 2), 30), which both Agent A and Agent B agreed on. Agent A is chosen as the agent which will make the initial offer.

- Round 1: Agent B offers the bid b1 = (0, 3, 1, 1, 30), and Agent A counter offers with the bid b2 = (0, 3, 3, 2, 30)

- Round 2: Agent B counter offers with the bid b3 = (0, 3, 1, 1, 30), and Agent A counter offers with the bid b4 = (0, 3, 3, 2, 30)

- Round 3: Agent B counter offers with the bid b5 = (0, 3, 1, 2, 30), and Agent A counter offers with the bid b6 = (0, 3, 3, 2, 30)

- Round 4: Agent B counter offers with the bid b7 = (0, 3, 2, 2, 30), and Agent A counter offers with the bid b8 = (0, 3, 3, 2, 30)

- Round 5: Agent A counter offers with the bid b9 = (0, 3, 2, 1, 30), and Agent B accepts the opponent's bid b9. The contract for this negotiation will be referred as contract 1.

Negotiation thread 3: The negotiation issues under this negotiation thread are determined by the CFP c7 = (1, 3, (3, 8), (1, 10), 90), which both Agent B and Agent C agreed on. Agent B is chosen as the agent which will make the initial offer.

- Round 1: Agent B offers the bid b1 = (1, 3, 8, 1, 30), and Agent C counter offers with the bid b2 = (1, 3, 3, 2, 30)

- Round 2: Agent B counter offers with the bid b3 = (1, 3, 7, 10, 30), and Agent C counter offers with the bid b4 = (1, 3, 3, 2, 30)

- Round 3: Agent B counter offers with the bid b5 = (1, 3, 7, 10, 30), and Agent C counter offers with the bid b6 = (1, 3, 3, 2, 30)

- Round 4: Agent B counter offers with the bid b7 = (1, 3, 6, 4, 30), and Agent C counter offers with the bid b8 = (1, 3, 3, 2, 30)

- Round 5: Agent B counter offers with the bid b9 = (1, 3, 4, 4, 30), Agent C accepts the opponent's bid b9. The contract for this negotiation will be referred as contract 2.

Negotiation thread 4: The negotiation issues under this negotiation thread are determined by the CFP c8 = (1, 3, (1, 15), (1, 10), 30), which both Agent B and Agent C agreed on. Agent B is chosen as the agent which will make the initial offer.

- Round 1: Agent B offers the bid b1 = (1, 3, 15, 1, 30), and Agent C counter offers with the bid b2 = (1, 3, 3, 2, 30)

- Round 2: Agent B counter offers with the bid b3 = (1, 3, 15, 10, 30), and Agent C counter offers with the bid b4 = (1, 3, 3, 2, 30)

- Round 3: Agent B counter offers with the bid b5 = (1, 3, 10, 10, 30), and Agent C counter offers with the bid b6 = (1, 3, 3, 2, 30), and

- Round 4: Agent B counter offers with the bid b7 = (1, 3, 9, 4, 30), and Agent C counter offers with the bid b8 = (1, 3, 2, 2, 30)

- Round 5: Agent B counter offers with the bid b9 = (1, 3, 8, 4, 30), Agent B accepts the opponent's bid b9. The contract for this negotiation will be referred as contract 3.

Negotiation thread 5: The negotiation issues under this negotiation thread are determined by the CFP c11 = (2, (3, 4), (2, 12), (1, 18), 180, which both Agent C and Agent D agreed on. Agent C is chosen as the agent which will make the initial offer.

- Round 1: Agent C offers the bid b1 = (2, 3, 12, 1, 180), and Agent D counter offers with the bid b2 = (2, 3, 12, 2, 180)
- Round 2: Agent C counter offers with the bid b3 = (2, 3, 10, 11, 180), and Agent D counter offers with the bid b4 = (2, 3, 3, 2, 180)
- Round 3: Agent C counter offers with the bid b5 = (2, 3, 10, 11, 180), and Agent D counter offers with the bid b6 = (2, 3, 3, 2, 180)
- Round 4: Agent C counter offers with the bid b7 = (2, 3, 9, 4, 180), and Agent D counter offers with the bid b8 = (2, 3, 2, 2, 180)
- Round 5: Agent C counter offers with the bid b9 = (2, 3, 8, 4, 180), Agent D accepts the opponent's bid b9. The contract for this negotiation will be referred as contract 4.

Negotiation thread 6: The negotiation issues under this negotiation thread are determined by the CFP c12 = (2, 4, (1, 10), (1, 21), 190), which both Agent C and Agent D agreed on. Agent C is chosen as the agent which will make the initial offer.

- Round 1: Agent C offers the bid b1 = (2, 4, 10, 15, 190), and Agent D counter offers with the bid b2 = (2, 4, 3, 2, 190)
- Round 2: Agent C counter offers with the bid b3 = (2, 4, 10, 15, 190), and Agent D counter offers with the bid b4 = (2, 4, 3, 2, 190)
- Round 3: Agent C counter offers with the bid b5 = (2, 4, 7, 13, 190), and Agent D counter offers with the bid b6 = (2, 4, 3, 2, 190)
- Round 4: Agent C counter offers with the bid b7 = (2, 4, 7, 12, 190), and Agent D counter offers with the bid b8 = (2, 4, 2, 2, 190)

- Round 5: Agent C counter offers with the bid b9 = (2, 4, 6, 10, 190), and Agent D accepts the opponent's bid b9. The contract for this negotiation will be referred as contract 5.

Post Negotiation Phase

Contracts Executed:

The contracts are executed as follows:

- Contract 1 = 0, 2, 3, 7, 30 is executed without any breach, the Agent A delivers 7 units of products 0 to Agent B. Agent B pays 21 units of funds to Agent A.
- Contract 2 = 0, 2, 4, 8, 30 is executed without any breach, the Agent A delivers 8 units of products 0 to Agent B. Agent B pays 32 units of funds to Agent A.

Contract Signing

- Contract 1: Both Agent A and B signs the contract (0, 3, 2, 1, 30).
- Contract 2: Both Agent B and C signs the contract (1, 3, 4, 4, 30).
- Contract 3: Both Agent B and C signs the contract (1, 4, 8, 4, 30).
- Contract 4: Both Agent C and D signs the contract (2, 3, 8, 4, 180).
- Contract 5: Both Agent C and D signs the contract (2, 4, 6, 10, 190).

Productions Scheduled: Agent B schedules production of 15 product 1's from 15 product 0's which received from Agent A. The production cost of product 1 from product 0 is 2 units of funds per product 0s processed, accordingly the production cost in total is 2*15=30.

Time step 3:

Pre-negotiation phase

Table 2: Simulation State at the End of Step 2

Agent \ Stats	Product 0	Product 1	Product 2	Funds
Agent B	0	0	0	917
Agent C	0	0	0	1000

- Agent D posts the following CFPs: c1 = (2, 4, (2, 12), (1, 18), 100).

  The agents who are willing to sell products to the publishers of CFPs send the following negotiation requests.

- Agent C requests a negotiation to the CFPs c1, the negotiation request is accepted by Agent D. The negotiation thread for the negotiation under the CFP c1 is referred as negotiation thread 1.

Negotiation Phase

Negotiation thread 1: The negotiation issues under this negotiation thread are determined by the CFP c1 = (2, 4, (2, 12), (1, 18), 100), which both Agent C and Agent D agreed on. Agent B is chosen as the agent which will make the initial offer.

- Round 1: Agent C offers the bid b1 = (2, 4, 12, 15, 190), and Agent D counter offers with the bid b2 = (2, 4, 3, 2, 190)

- Round 2: Agent C counter offers with the bid b3 = (2, 4, 12, 15, 190), and Agent D counter offers with the bid b4 = (2, 4, 3, 2, 190)

- Round 3: Agent C counter offers with the bid b5 = (2, 4, 10, 13, 190), and Agent D counter offers with the bid b6 = (2, 4, 3, 2, 190)

- Round 4: Agent C counter offers with the bid b7 = (2, 4, 9, 12, 190), and Agent D counter offers with the bid b8 = (2, 4, 2, 2, 190)

- Round 5: Agent C counter offers with the bid b9 = (2, 4, 8, 10, 190), and Agent D accepts the opponent's bid b9. The contract for this negotiation will be referred as contract 5.

Post Negotiation Phase

Manufacturing Processes Completed

- The manufacturing process Agent B scheduled at time step 2 is finished, and Agent B now have 15 units of product 1 and all the 15 product 0's in Agent B's inventory is consumed.

Contracts Executed

- The contract c=(1,3,4,8,30) is executed successfully. Thus, Agent B delivers 8 units of product 1 to Agent C, and Agent C pays 4*8 = 32 units of funds to Agent B.

- The contract c=(1,3,3,7,30) is executed successfully. Thus, Agent B delivers 7 units of product 1 to Agent C, and Agent C pays 7*3 = 21 units of funds to Agent B.

- The contract c=(0,3,2,1,30) is executed successfully. Thus, Agent A delivers 1 unit of product 0 to Agent B, and Agent B pays 2*1 = 2 units of funds to Agent A.

- The contract c=(1,3,4,4,30) is failed to be executed because Agent B is supposed to deliver 4 units of product 1 to Agent C, but Agent B doesn't have any product 1's left in it's inventory. Therefore, an insufficient products breach is reported to the bulletin board. The breach report is shown as the tuple below:

$$b = (p, v, l) \tag{3}$$

where,

p : Perpetrator

v : Victim

l : Breach level

The breach reported to the bulletin board is therefore (b, c, 1). The breach report is publicly available to all agents. Note that breach level ranges between $1 \geq l > 0$. The breach level 1 indicates a full breach (i.e when none of the deliverables are delivered to the the victim.). A partial breach may occur when some of the deliverables are delivered to victim. Therefore, the breach level for the insufficient products breach is calculated as shown below:

$$l_{ip} = \frac{a}{a'} \tag{4}$$

where,

a : Amount of products delivered by the perpetrator.

a' : Amount of products supposed to be delivered by the perpetrator according to the contract terms.

Similarly, an insufficient funds for breach would have been reported if the perpetrator (The perpetrator would be the buyer party in this case) failed to pay all the fee for the product procurement. An insufficient funds breach can be represented as follows:

$$l_{if} = \frac{f}{f'} \tag{5}$$

where,

f : Amount of products delivered by the perpetrator.

f' : Amount of products supposed to be delivered by the perpetrator according to the contract terms.

In this breach case, Agent B is the perpetrator, according to the contract terms. Agent B is liable to pay negotiate penalty (30) times $a-a'$ (4-0) units of products to the victim. Therefore, Agent B pays $4 * 30 = 120$ units of funds to Agent C.

- The contract c = (1,3,8,4,30) is failed to be executed because Agent C is supposed to deliver 4 units of product 2 to Agent D, but Agent D doesn't have any product 2's left in it's inventory. Therefore, an insufficient products breach is reported to the bulletin board. The breach report is represented as b=(c,d,1). In this breach case, Agent C is the perpetrator, according to the contract terms. Agent C is liable to pay negotiated penalty (30) times $a-a'$ (4-0) units of products to the victim. Therefore, Agent C pays $4*30 = 120$ units of funds to Agent D.

- The contract c = (2,3,8,4,180) is failed to be executed because Agent C is supposed to deliver 4 units of product 2 to Agent D, but Agent C doesn't have any product 2's left in it's inventory. Therefore, an insufficient products breach is reported to the bulletin board. The breach report is represented as b=(c,d,1). In this breach case, Agent C is the perpetrator, according to the contract terms. Agent C is liable to pay negotiate penalty (180) times $a-a'$ (4-0) units of products to the victim. Therefore, Agent C pays $4 * 180 = 720$ units of funds to Agent D.

Contract Signing

- Contract 1: Both Agent A and B sign the contract (2, 4, 8, 10, 190).

Productions Scheduled Agent C schedules production of 15 product 2's from 15 product 1's which received from Agent B.

Time step 4:

Table 3: Simulation State at the End of Step 3

Agent \ Stats	Product 0	Product 1	Product 2	Funds
Agent B	1	0	0	167
Agent C	0	0	0	848

Pre-negotiation phase

None of the agents posts CFPs in this time step.

Negotiation phase

None of the agents negotiate in this time step.

Post Negotiation phase

Manufacturing Processes completed

The manufacturing process Agent C scheduled at time step 3 is finished. Agent C now have 15 units of product 2 and all the product 1's in Agent C's inventory is consumed.

Contracts Executed

- The contract c=(2,4,6,10,190) is executed successfully. Thus, Agent C delivers 10 units of product 2 to Agent D, and Agent D pays 6*10 = 60 units of funds to Agent C.

- The contract c=(2,4,8,10,30) is failed to be executed because Agent C is supposed to deliver 10 units of product 2 to Agent D, but Agent C have only 5 units of products in its inventory. Therefore, an insufficient products breach is reported to the bulletin board. Note that this breach is a case of partial breach because Agent C can deliver 5 units of product 2s. Therefore, the breach level for this breach case is 5/10 = 0.5. The breach report b=(c,d,0.5) is posted to the bulletin board and it becomes publicly available to all of the agents in the simulation. Agent D pays 8*5=40 units of funds to Agent C and Agent C is supposed to pay 5*190 = 950 units of funds to Agent D. However, Agent C has insufficient funds to pay the total negotiated penalty. Therefore, it goes bankrupt (the balance goes below 0). The bankrupted agents get a negative score at the end of the simulation

The final state of the simulation is illustrated in the table below:

Table 4: Simulation State at the End of The Example simulation

Agent \ Stats	Product 0	Product 1	Product 2	Funds
Agent B	1	0	0	848
Agent C	0	0	0	-743

The simulation ends because the simulation length is 4 time steps as stated. After the simulation ends, the scores each factory manager agent gets are calculated as mentioned in Equation 2. Thus, the final scores for the factory agents are 0.848 and 0 for the agents B and C respectively. Agent B is the winner in this simulation run because it has the highest positive score at the end of the simulation.

CHAPTER III

PROPOSED AGENT STRATEGY

In this section we address the challenges stated in the former section and explain our factory manager agent's policy on decision making. The proposed factory agent strategy evaluates offer with respect to inventory changes and negotiate accordingly. In the following section, we explain our factory agent in a detailed way.

3.1 CFP Posting Strategy

Our factory manager agent is interested in buying all kinds of products regardless of whether the product can be processed in its factory or not. The reason for this strategy is the fact that products can be procured from and sold to other agents with a profit. Our agent posts CFPs to buy the products from its fair trade value or lower. To determine the fair trade value, it uses the catalog price, its is determined by the simulation as well as taking the production costs into account. At the beginning of a simulation run, the catalog value of the products is exposed to all agents in the simulation environment as follows:

$$c_i = 1.15 * (c_{i-1} + \mu_i) \tag{6}$$

c_i : Catalog price of the product type i

c_{i-1} : Catalog price of the product type i-1

μ_i : The average production cost of the product type i using the product type i-1.

Informally, the catalog price of a product type i is determined by the summation of the catalog price of the product i-1, which is the type of product needed to produce product of type i in factory managers' factories, and the average production cost of producing product with type i from the product of type i-1. The summation is then multiplied by the constant 1.15 for the added production value.

The CFP is posted in the following way:

$\alpha \leftarrow 16$
$\theta \leftarrow 10$
$b \leftarrow 10.5$
for p *in products* **do**
    if $average\_supplies[p]/average\_demands[p] < 2$ **then**
        for s *in* $range(\theta)$ **do**
            $q \leftarrow (1, s + \alpha)$
            $d \leftarrow min(c_{step} + s, max\_steps)$
            $u \leftarrow max(0.5, cprices[p])$
            $post(CFP(p, d, u, q, b))$
        end
    end
end

Algorithm 1: CFP posting strategy

Where,

products : The array of all product types available in a simulation run.

c_{step} : The current time step.

max_{steps} : The simulation duration in terms of the number of time steps.

cprices : The array of catalog prices for each product, the prices are determined by the simulation randomly.

This CFP posting strategy aims to post CFPs with a wide range of negotiation issues. Our agent's CFP posting strategy has three advantages. One is the fact that the broad range of negotiation issues would be compelling to the other agents to request negotiation for our agent's CFPs, because the flexibility of the negotiation issues would be favorable to their requesting negotiation to CFPs especially if they have a selective negotiation requesting policy. The another advantage is having a broad range of negotiation issues would be easier for our agent to reach an agreement with other agents due to the abundance of possible bids to be offered to the other party during negotiation. Finally, our agent considers the supply/demand ratio of a specific product to post CFP, it never posts a CFP to buy a product if the supply/demand ratio for that product is more than 2. This helps our agent not to procure too much.

3.2 Requesting Negotiations to Other CFPs Strategy

At each time step, our agent requests negotiations to all CFPs, whose publishers did not go bankrupt, in the bulletin board. The motivation behind this strategy is the fact that there is no cost involved in making negotiations with other agents (Unless an agreement is reached in a negotiation and the contract is signed, the agents are not obliged to comply with any contracts). By following this strategy the agent maximizes the number of possible negotiation opportunities for the CFPs posted by other agents, and can choose a wider set of contracts to sign. Our agent does not request negotiation to a bankrupted agent because our agent, as a requester, would sell products to the publisher. Thus, there is no point in requesting negotiation the the CFPs whose publishers cannot pay the cost of purchase. This reduces the number of breached contracts and filters out unnecessary negotiations, which would optimize the speed of the simulation.

3.3 Responding to Negotiation Requests Strategy

Our agent accepts negotiation requests from all agents. Contrary to requesting negotiations to other CFPs strategy, our agent accepts the negotiation requests from the bankrupted agents as well. In the SCML 2019 version of the supply chain world, the agents who post the CFPs act as buyers and the agents who request for the negotiation act as the sellers. Thus, the sellers will not need to pay money to be in compliance with the contract. The motivation behind responding to negotiation requests strategy is analogous to requesting negotiation to CFPs in the bulletin board, that is trying to maximize the negotiation opportunities so our agent can get a broader range of contracts which can be be signed in a profit maximizing way.

3.4 Production Scheduling Strategy

Our agent decides to schedule production based on the supply-demand cost of the input and output of the product. The supply-demand cost of the input product is shown below:

$$\gamma\_input = \begin{cases} (avg_{sp\_i} - avg_{bp\_i}) * (d\_i - s\_i) & d\_i - s\_i > 0 \\ avg_{bp\_i} * (s\_i - d\_i) & d\_i - s\_i < 0 \\ 0 & d\_i - s\_i = 0 \end{cases} \tag{7}$$

where,

avg_{sp_i} : Average selling price of the input product

avg_{bp_i} : Average buying price of the input product

d_i : Average demand per step of the input product

s_i : Average supply per step of the input product

As for supply-demand cost of the output product, the product can be obtained in two ways: by producing it in the factory or procuring from other factory manager agents. Hence, the weighted average of the procurement and production costs must be considered as the average cost of obtaining the output product. The average cost of obtaining the input product is shown below:

$$avg_{oc\_o} = \begin{cases} \frac{\beta * avg_{bp\_o} + \eta * \phi}{\eta + \beta} & \eta + \beta \neq 0 \\ c_{price} & otherwise \end{cases} \tag{8}$$

avg_{oc_o} : Average cost of obtaining the output product.

β : Amount of output products products procured.

avg_{bp_o} : Average cost of procurement for the output product

η : Amount of output products produced

ϕ : Unit production cost

The average supply-demand cost of the output product is therefore,

$$\gamma\_output = \begin{cases} (avg_{sp\_o} - avg_{oc\_o}) * (d\_o - s\_o) & d\_o - s\_o > 0 \\ avg_{oc\_o} * (d\_o - s\_o) & d\_o - s\_o < 0 \\ 0 & d\_o - s\_o = 0 \end{cases} \tag{9}$$

Where,

avg_{sp_o} : Average selling price of the output product

avg_{bp_o} : Average buying price of the output product

d_o : Average demand for the output product per time step

s_o : Average supply for the output product per time step

Let $\gamma\_input(\omega)$, $\gamma\_output(\omega)$ denote the supply-demand cost functions under ω, where ω represents the amount of output product to be produced for the current time step. Formally:

$$\gamma\_input(\omega) = \begin{cases} (avg_{sp\_i} - avg_{bp\_i}) * ((d\_i + \omega) - s\_i) & (d\_i + \omega) - s\_i > 0 \\ avg_{bp\_i} * (s\_i - (d\_i + \omega)) & (d\_i + \omega) - s\_i < 0 \\ 0 & (d\_i + \omega) - s\_i = 0 \end{cases} \tag{10}$$

and,

$$\gamma\_output(\omega) = \begin{cases} (avg_{sp\_o} - avg_{oc\_o}) * (d\_o - (s\_o + \omega)) & d\_o - (s\_o + \omega) > 0 \\ avg_{oc\_o} * ((s\_o + \omega) - d\_o) & d\_o - (s\_o + \omega) < 0 \\ 0 & d\_o - (s\_o + \omega) = 0 \end{cases} \tag{11}$$

The value of ω is calculated such that $\gamma\_input + \gamma\_output$ is minimized, formally:

$$\omega = \min_{\omega}(\gamma\_input(\omega) + \gamma\_output(\omega)) \tag{12}$$

Note that our factory manager agent has only 10 production lines, and 1 input product is needed to product 1 output product. Therefore, the value ω can take ranges between:

$$0 \leq \omega \leq min(a_i, 10) \tag{13}$$

Where, a_i denotes Amount of input products in the inventory.

Therefore, the value of ω can be determined by solving the integer problems p1, p2, p3, p4 with the objective functions o1, o2, o3, o4 under the constraints c1, c2, c3, c4. Each integer problem has the following objective function and constraint sets as follows: p1=(o1, c1), p2=(o2,c2), p3=(o3,c3), p4=(o4,c4). The integer problem models are shown below:

- The objective function for the problem p1:

$$o1 = (avg_{sp\_i} - avg_{bp\_i}) * ((d\_i + \omega) - s\_i) + (avg_{sp\_o} - avg_{oc\_o}) * (d\_o - (s\_o + \omega)) \tag{14}$$

The constraints for the problem p1 (c1):

$$(d\_i + \omega) - s\_i \geq 0$$

$$d\_o - (s\_o + \omega) \geq 0$$

$$\omega - min(10, a_i) \leq 0$$

$$\omega \geq 0$$

$$\omega \in \mathbb{Z}$$

- The objective function for the problem p2:

$$o2 = (avg_{sp\_i} - avg_{bp\_i}) * ((d\_i + \omega) - s\_i) + avg_{oc\_o} * ((s\_o + \omega) - d\_o) \quad (15)$$

The constraints for the problem p2 (c2):

$$(d\_i + \omega) - s\_i \geq 0$$

$$d\_o - (s\_o + \omega) \leq 0$$

$$\omega - min(10, a_i) \leq 0$$

$$\omega \geq 0$$

$$\omega \in \mathbb{Z}$$

- The objective function for the problem p3:

$$o3 = avg_{bp\_i} * (s\_i - (d\_i + \omega)) + (avg_{sp\_o} - avg_{oc\_o}) * (d\_o - (s\_o + \omega)) \quad (16)$$

The constraints for the problem p3 (c3):

$$(d\_i + \omega) - s\_i \leq 0$$

$$d\_o - (s\_o + \omega) \geq 0$$

$$\omega - min(10, a_i) \leq 0$$

$$\omega \geq 0$$

$$\omega \in \mathbb{Z}$$

- The objective function for the problem p4:

$$o4 = avg_{bp\_i} * (s\_i - (d\_i + \omega)) + avg_{oc\_o} * ((s\_o + \omega) - d\_o) \tag{17}$$

The constraints for the problem p4 (c4):

$$(d\_i + \omega) - s\_i \leq 0$$

$$d\_o - (s\_o + \omega) \leq 0$$

$$\omega - min(10, a_i) \leq 0$$

$$\omega \geq 0$$

$$\omega \in \mathbb{Z}$$

The value of ω is the minimum objective value of the problems p1, p2, p3 and p4.

3.5 Negotiation Strategy

The negotiation strategies are to used to determine what bid to offer or accept at a particular negotiation round. Our agent has different negotiation strategies according to its role : selling party or buying party.

3.5.0.1 Negotiation Strategy for Selling Negotiations

Utility Function

When our agent is the selling party, it is negotiating with an agent, which is interested in buying products from our agent. Our agent measures the goodness of a bid as the total profit it would make if there was an agreement with that bid. The quantitative value of a goodness of an offer is called utility value that ranges between 0 and 1. The utility value of 0 means the worst possible offer while the utility value of 1 indicates the best possible offer. The utility value of an offer during a selling negotiation is estimated as:

$$Overall(o) = (1 - \Omega)U_S(o) + (\Omega)U_O(o)$$

Where,

$Overall(o)$: The overall utility.

$U_S(o)$: The utility function for our agent ,without considering the opponent's utility.

$U_O(o)$: Estimated opponent utility function.

Ω : The weight of the opponent's estimated utility to calculate the overall utility

Our agent's utility function without considering the opponent's estimated utility is shown below:

$$U_S(o) = \begin{cases} max(((o[s_p] - V_p) * o[q_o]), 0) & o[q_o] \geq inventory[q_o] \\ 0 & otherwise \end{cases} \tag{18}$$

Where,

$U_S(o)$: Average cost of obtaining the output product.

V_p : Fair price of the product p.

q_o : Quantity of the product p in the offer.

s_p : Unit price of p in the offer.

The fair price of a product p (V_p) is calculated in the following way:

$$V_p = \begin{cases} c_{price} & \eta_p = \beta_p = 0 \\ \kappa_p & (\beta_p \neq 0) \wedge (\eta_p = 0) \\ V_i + \phi_i & (\beta_p = 0) \wedge (\eta_p \neq 0) \\ \frac{((V_i+\phi_i)*\eta_p)+(\kappa_p*\beta_p)}{\eta_p+\beta_p} & (\beta_p \neq 0) \wedge (\eta_p \neq 0) \end{cases} \tag{19}$$

Where,

V_p : Fair price of the product p

c_{price} : Catalog price of the product p

κ_p : Weighted average price of procurement for product p

η_p : Amount of product p produced in factory.

β_p : Amount of product p procured by negotiations

ϕ : Unit production cost

V_i : Fair price of the product input product required to produce product p

Note that the weighted average cost of procurement for product p is equal to:

$$\kappa_p = \frac{s_p}{\beta_p} \tag{20}$$

Where, s_p denotes Total amount of money spent to procure product p. Thus, the utility function for the selling negotiations consider the fair price as the threshold price below which would be unprofitable for the our factory manager agent, because when a product is obtained by negotiations, production, or both, the fair price of the product is equal to the average cost of obtaining the product. In case the product is only procured by the negotiations, the fair price is equal to the total amount of money spent for procurement, divided by the amount of products bought (i.e $\kappa_p = \frac{s_p}{\beta_p}$). If the product is obtained without negotiations but only producing in the factory, the fair price of the product is equal to the average cost of production which is equal to the fair price of the input product used for producing the output product, plus the unit production cost. (i.e $V_i + \phi_i$). If a product is obtained both by procurement and production in the factory, the fair price of product p is equal to the weighted average of the cost of production and procurement cost (i.e $\frac{((V_i+\phi_i)*\eta_p)+(\kappa_p*\beta_p)}{\eta_p+\beta_p}$). If the product is never obtained and so there is no data available on the fair price of the product, the fair price is assumed to be the catalog price of the product. If the quantity of a product is more than the product in the inventory, the utility value of the product is zero because that offer would not be a feasible offer that our agent can accept.

The opponent's utility $O_B(o)$ for an offer is estimated based on the average quantity and weighted average unit price of the opponent's offers during the negotiation. Note that the estimation requires some opponent offer data. Thus, for the first half of the negotiation, our agent does not take the opponent's utility into account (i.e $\Omega = 0$). After the first half ends, our agent estimates the opponent's utility and calculates the overall utility for all offers in the following way.

The opponent's estimated utility value for an offer is based on how much the quantity of product in the offer deviates from the average and how expensive the unit price is than the average weighted unit price from the offers. Both the quantity and the unit price have the equal weights on opponent's estimated utility.

The overall utility of an offer for our agent is the weighted sum of our agent's utility without taking opponent's utility into account and the opponent's estimated utility. The weight Ω is determined as follows:

```
index = 0
for offer_value_pair in possible_offers_and_utility_map do
    my_utility ← offer_value_pair.get("utility")
    offer ← offer_value_pair.get("offer")
    deviation ← | offer.quantity − average_quantity |
    deviation_ratio ← deviation/max_quantity
    quantity_utility ← (1 − deviation_ratio)
    if offer.price ≤ weighted_average_price then
        price_utility ← 0.5
    end
    else
        price_utility ← (average_weighted_unit_price/offer.unit_price) ∗ 0.5
    end
    opponent_utility ← price_utility + quantity_utility
    overall_utility ← opponent_utility_weight ∗ opponent_utility + (1 −
      opponent_utility_weight) ∗ my_utility
    possible_offers_and_utility_map[index] = (offer, overall_utility)
    index = index + 1
end
```

Algorithm 2: Opponent's utility estimation and overall utility calculation

$$\Omega = \begin{cases} 0.5 & supply/demand \geq 2 \\ 0.5 * (supply/demand) - 0.5 & otherwise \end{cases} \tag{21}$$

The equation implies that if the supply/demand ratio for the product to be negotiated is more than or equal to 2, the opponent's utility reaches its maximum value 0.5. At this value, our agent's utility without considering the opponent's utility and the opponent's utility have the same weight on the overall utility. If the demand is less than 2, the weight of the opponent's utility weight decreases. Thus, our agent becomes more indifferent to the opponent's preferences as the the ratio of supply and demand for a product decreases.

Acceptance Strategy

The acceptance strategy is based on the time based concession strategy [4]. The concession function is shown below:

$$cs = 1 + (r - 1) * t^c \tag{22}$$

Where,

cs : Concession score

r : Reservation value

t : Relative time

c : Concession coefficient, $c \geq 1$

The concession score cs is the target utility of the offer below which is not acceptable to our agent. Therefore, if other party makes an offer below the concession score, the offer is rejected. If the utility value of the offer is greater than or equal to the concession score, the offer is accepted. This acceptance strategy is known as the ACNext acceptance strategy [5]. The t denotes the timeline, where $t = 0$ represents the start of the negotiation and $t = 1$ is the end of the negotiation. For instance, if the negotiation deadline is 20 rounds, $t = 0$ before the round 1 ends, and $t = 0.05$ at the end of the first round, $t = 0.1$ at the end of second round, and so forth. The $c \in \mathbf{Q}^+$ denotes the concession coefficient. As the value of c increases, the concession function slowly decreases as the t value is far from 1, but decreases sharply to the reservation value as t tends to be 1. The first derivative of the function with respect to t is shown below:

$$\frac{d}{dt}(cs) = (r-1)c * t^{c-1} \tag{23}$$

Note that the derivative of cs is always negative during the negotiation because $c(r-1) < 0$. Thus, the function is monotonic and decreasing. This indicates that the agent concedes as t increases. The reservation value r is the minimum value of the function which takes the minimum value when $t = 1$.

Offering Strategy

The offering strategy is based on the concession score as well, at each round, among all possible offers, our agent makes the offer whose utility is the closest to the concession score. Formally:

```
min_utility_gap ← 1
concession_score ← get_concession_score()
offer_to_be_proposed ← null
for offer in offers do
    offer_utility ← offer.utility
    utility_gap ← | concession_score − offer_utility |
    if utility_gap < min_utility_gap then
        min_utility_gap ← utility_gap
        offer_to_be_proposed ← offer
    end
end
propose(offer_to_be_proposed)
```

Algorithm 3: Offering strategy

Reservation Value Update Based on The Offers in All Negotiation Threads

When $t = 0.5$ (i.e when the remaining rounds for negotiations are halved), our agent analyzes the bids offered by other parties so far to check if there expected to be enough demand for the products to be sold, and update the reservation values of each negotiators accordingly. The reservation values are updated as follows:

```
expected_demands ← initialize_product_demands_map()
for product in products do
    negotiators ← negotiators_by_product.get(product)
    expected_demand ← 0
    for negotiator in negotiators do
        opponent ← negotiator.opponent
        opponent_offers ← negotiator.opponent_offers
        success_rate ← success_rates.get(opponent)
        compliance_rate ← 1 − breach_rates.get(opponent)
        delivery_rate ← success_rate ∗ compliance_rate
        for offer in opponent_offers do
            quantity ← offer.quantity
            expected_demand ← expected_demand + (quantity ∗ delivery_rate)
        end
    end
    expected_demands[product] ← expected_demand
end
for product in products do
    average_demand ← average_demands.get(product)
    expected_demand ← expected_demands.get(product)
    reservation_value_change ←
      bound(expected_demands/average_demands)
    negotiators ← negotiators_by_product.get(product)
    for negotiator in negotiators do
        negotiator.reservation_value ←
          min(1, negotiator.reservation_value ∗ reservation_value_change)
    end
end
```

Algorithm 4: Reservation value update

The reservation value update algorithm is based on the comparison between the average demands and average quantity of products other parties stated in their bids multiplied by the negotiation success rate with the corresponding agent multiplied by the breach rate of the agent, which is a measure of how compliant the agent is with the contracts. If the expected demand is greater than the average demand for a product, the negotiator's current reservation values are multiplied by the ratio of expected demands and average demands. The intuition behind this approach is the fact that if there's an excess amount of expected demands, the negotiators' reservation values are increased proportionally; otherwise decreased. Note that the reservation value update is bounded at 25%. Therefore, the reservation value cannot decrease or increase more than 25%.

Adjusting the Reservation Value with Respect to Average Supplies and Demands

Our agent uses adaptive reservation value approach to calibrate its greediness with respect to supply and demand of the products. By increasing the reservation value, our agent acts more greedy. When the reservation value is decreased, our agent acts more altruistic. If the supply/demand ratio is high for a product (i.e greater than 1.1), the reservation value is decreased proportional to the excess supply. When the supply/demand ratio is less than 1.1, our agent increases the reservation value by 0.01. The reservation value updates aim to keep a balance between supply and demand. When the supply is too high, the reservation value is decreased to help our agent to reach more successful selling negotiations that will increase the demand. When the supply is not too much, our agent aims to preserve its products in the inventory and only sells it by a high profit margin. The algorithm below shows the reservation value update process:

```
δ ← 0.01
for p in products do
    Δ ← avg_supplies[p]/avg_demands[p]
    if Δ > 1.1 then
        r[p] ← min(r[p] − δ ∗ Δ, 0)
    end
    else
        r[p] ← max(r[p] + δ, 1)
    end
end
```

Algorithm 5: Reservation value based on supply and demands

3.5.0.2 Negotiation Strategy for Buying Negotiations

Utility Function

Buying negotiations refers to the negotiations in which our agent is acting as the buyer party. Our agent uses the same negotiation acceptance and offering strategies as while negotiating as the seller party. However, the utility function, and the reservation value updating process differs from the negotiations in which our agent acts as a seller party.

Our factory manager agent aims to buy products with a low price. Therefore, the utility function in buying negotiations is based on a fair value of a product. Above the fair value, the product is considered to be too expensive, if the unit price of the product is below the fair value, the utility value of the offer is determined by how much the unit price is below the fair value. As our agent buys more of the cheap products, the utility value increases even further. Formally:

$$U_B(o) = \begin{cases} 0.01 * o[q_o] & c_{price} = o[s_p] \\ (c_{price} - o[s_p]) * o[q_o] & otherwise \end{cases} \tag{24}$$

Where,

$U_B(o)$: Utility value of the offer o

q_o : Quantity of the product in the offer o

c_{price} : Catalog price of the product

s_p : Unit price of the product in the offer o

When $c_{price} = o[s_p]$, the product is just sold at the fair value, because there is no storing costs involved. The quantity of the product in the offer is multiplied by a small constant to assign the corresponding bid a non zero utility. If $c_{price} < o[s_p]$, the product is sold below the catalog value, which means the product is sold at a good price. The utility value is equal to the difference between the catalog price and the unit price, and utility value is then multiplied by the quantity to encourage buying more of these products. Note that if $c_{price} > o[s_p]$, the product is considered as overpriced, which yields a negative utility value, the negative utility values are assigned to zero utility value as the utility values of all bids are normalized.

Reservation Value Update based on Negotiation Outcomes

The reservation value is determined for each product based on the negotiation success and failure rates. At each negotiation failure, the reservation value of the corresponding product decreases. If the negotiation is successful, the reservation value increases. The motivation behind this strategy is the fact that a negotiation failure may be an indication of our agent acting too greedy. Thus, the reservation value is slightly decreased for the corresponding product to allow our agent to reach more successful negotiations. Otherwise, the reservation value is increased. The reservation value updates are done to find a good level of greediness for the buying negotiations.

The reservation value is updated in the following way:

$$r_{p,t+1} = \begin{cases} r_{p,t} - 0.01 & \text{Failure} \\ r_{p,t} + 0.01 & \text{Otherwise} \end{cases} \tag{25}$$

Where,

$r_{p,t+1}$: Reservation value for the next time step

$r_{p,t}$: Reservation value for the current time step.

Similar to reservation value updates with respect to expected demands for selling negotiations, the reservation values are updated for the expected supplies. When the half of the negotiation rounds are completed in all negotiation threads, the offers so far is analyzed to calculate an expected supply. If there expected to be a surplus of goods, the reservation values are increased by the ratio of expected supplies and average supplies per time step. Note that the reservation value updates are capped by 25%. The formal procedure for the reservation value update based on expected demands is shown below:

```
expected_supplies ← initialize_product_supplies_map()
for product in products do
    negotiators ← negotiators_by_product.get(product)
    expected_supply ← 0
    for negotiator in negotiators do
        opponent ← negotiator.opponent
        opponent_offers ← negotiator.opponent_offers
        success_rate ← success_rates.get(opponent)
        compliance_rate ← 1 − breach_rates.get(opponent)
        delivery_rate ← success_rate ∗ compliance_rate
        for offer in opponent_offers do
            quantity ← offer.quantity
            expected_supply ← expected_supply + (quantity ∗ delivery_rate)
        end
    end
    expected_supplies[product] ← expected_supply
end
for product in products do
    average_supply ← average_supplies.get(product)
    expected_supply ← expected_supplies.get(product)
    reservation_value_change ← bound(expected_supplies/average_supplies)
    negotiators ← negotiators_by_product.get(product)
    for negotiator in negotiators do
        negotiator.reservation_value ←
          min(1, negotiator.reservation_value ∗ reservation_value_change)
    end
end
```

Algorithm 7: Reservation value update

3.6 Contract Signing Strategy for Buying Contracts

Let C be the set of contracts to make a signing decision. A contract $c \in C$ can be represented as the tuple shown below:

$$c = (p, d, u, q, b) \tag{26}$$

where,

c : Contract

p : Product type

d : Delivery time

u : Unit price

q : Quantity

To make a signing decision for buying contracts we have modeled the cost of procurement as an integer programming in which the total cost of a particular product type is minimized. The total cost is calculated as the summation of mispurchasing cost and nominal purchasing cost. The total cost of mispurchasing for items not produced in factory is calculated as shown below:

$$m_p = \begin{cases} (avg_{sp} - avg_{bp}) * (d - \sum_{c \in C_{signed}} q) & d - \sum_{c \in C_{signed}} q > 0 \\ avg_{bp} * ((\sum_{c \in C_{signed}} q) - d)) & d - \sum_{c \in C_{signed}} q < 0 \\ 0 & otherwise \end{cases} \tag{27}$$

where,

avg_{sp} : Average selling price

avg_{bp} : Average buying price

d : Average demand per step

C_{signed} : The set of signed contracts

Note that in the first case where $d - \sum_{c \in C_{signed}} q > 0$ there's a shortage of products because the products acquired are lower than the average demand per time step. For the shortage case, there's an opportunity cost of $avg_{sp} - avg_{bp}$ because the factory manager agent misses the opportunity to sell more products due to shortage. Therefore, the opportunity cost incurred is the multiplication of the unit profit multiplied by the number of items in shortage.

For the second case, where $d - \sum_{c \in C_{signed}} q < 0$ there's a surplus of products because the products acquired are higher than the average demand per time step. For the surplus case, there's a cost incurred for products not sold which is equal to the average cost of procurement. Because the extra items are not expected to be sold. Therefore, the mispurchasing cost for the surplus case is the average buying price of the product multiplied by the amount of surplus.

The mispurchasing cost for the product produced in the factory is the multiplication is similar to the goods traded (i.e those procured and sold with a profit, without being processed in the factory). The difference is the addition of average cost of production added to the cost of average procurement. The mispurchasing cost of product produced in the factory is shown below.

$$m_p = \begin{cases} (avg_{sp} - (avg_{bp} + avg_{oc})) * (d - \sum_{c \in C_{signed}} q) & d - \sum_{c \in C_{signed}} q > 0 \\ (avg_{bp} + avg_{oc}) * ((\sum_{c \in C_{signed}} q) - d)) & d - \sum_{c \in C_{signed}} q < 0 \\ 0 & otherwise \end{cases} \quad (28)$$

where,

avg_{oc} : Average obtaining cost

The nominal cost of purchasing is calculated as follows

$$\sum_{c \in C_{signed}} q * u \tag{29}$$

The mathematical model of the total cost is shown below:

$$Min(k * (avg_{sp} - avg_{bp})(d - \sum_{c \in C} c_q i_c) + (1-k) * avg_{bp}((\sum_{c \in C} c_q i_c) - d) + \sum_{c \in C} u_c c_q i_c) \tag{30}$$

subject to:

$$(\sum_{c \in C} c_q i_c) - d + M(1-k) \geq 0$$

$$(d - (\sum_{c \in C} c_q i_c)) + Mk \geq 0$$

$$i, k \in \{0, 1\}$$

Where

i_c : Binary variable indicating if the contract $c \in C$ is signed.

k : Binary variable indicating if there is a shortage.

The mathematical model is a non-linear integer programming. This problem can be reduced to two linear integer programming problems by considering all possible cases of k=0 and k=1. Optimal value of the non-linear problem is equal to the feasible solutions of the sub-problems with the minimum objective value.

3.7 Contract Signing Strategy for Selling Contracts

The contract selling strategy is used to determine what contracts to sign for the selling negotiations. To decide what contracts to sign for the selling negotiations, our agent checks its future schedule to see if it can satisfy the terms of the contracts (i.e whether it has enough of the products to comply with the contract). If the contract terms are satisfactory with the schedule (i.e there expected to be enough of the products for the contract) the contract is signed. The contract is refused to be signed otherwise.

The future schedule is determined by the product inflows (by buying negotiations and via production) minus outflows (by selling negotiations or production), plus the amount of product in the inventory. Note that breach rate is used as the main metric to calculate the expected consequences of the contracts. Thus, the expected inflow of goods for the buying negotiations, or expected outflow of the goods for selling negotiations is:

$$expected\_flow = contract.quantity * (1 - breach\_rate) \tag{31}$$

If the contract terms are satisfactory with the schedule (i.e there expected to be enough of the products for the contract) the contract is signed. Otherwise, the contract is refused to be signed.

CHAPTER IV

EVALUATION

In order to evaluate the performance of the propose agent, we conduct some experiments in NEGMAS [2] which is a multiagent platform described in the former section 2. In this section, we first present our experimental setup and then report our results with a brief discussion.

4.1 Experiment Setup

We have tested our agent's performance by running simulation runs against the GreedyFactoryManagerAgent, a factory manager agent provided by the organizing comitee, and the top performing agents in ANAC SCML 2019, NVM InsuranceFraudFactoryManagerAgent (IFFM) and FJ2 agents. In the supply chain world simulation, we used the fixed parameters determined by the organized comitee of the ANAC 2019. As for the adjustable parameter, time steps, we used 40 time steps in our work. The reason why we used 40 steps was the fact that we observed the all the test simulation runs during debugging, testing new features and performance reached an approximately steady state for around 15-25 simulation steps. Because the other agents went bankrupt or refused to negotiate further with the other agents. The simulation parameters [6] are shown below:

Table 5: Experiment Parameters

Parameter	Value
Type of Raw Materials	1
Number of Intermediate Products	Uniform(1, 4)
Number of Final Products	1
Number of Miners	5
Number of Consumers	5
Starting Balance	1000
Production Line Count	10
Amount of Manufacturing Process Inputs	1
Amount of Manufacturing Process Outputs	1
Time Required for Manufacturing Process	1 Step
Number of Simulation Steps	40
Simulation Time Limit	7200 Secs
Negotiation Time Limit	120 Secs
Negotiation Rounds Limit	20
Negotiation Time Limit for Each Round	10 Secs
Negotiation Speed Multiplier	21
Immediate Negotiations	No
Default Grace Period for Contract Signing	1
Transportation Delay	0
Negotiable Penalties	Yes
Allow Breach Renegotiations	Yes
Global Breach Penalty	0.02
Base Insurance Premium	0.1

4.2 Experiment Results

We have used the top performing agents in ANAC 2019 SCML to test the performance of our agent. The name of the agents are SAHA Factory Manager Agent Agent, Insurance Factory Manager Agent, and NVM Factory Manager Agent. Additionally, we have created two variants of the proposed agent namely MyGreedyAgentV2 and MyGreedyAgentV3 in order to study the effect of some specific components of the proposed approach. To investigate the effect of updating reservation value, we create a replica of the proposed agent but MyGreedyAgentV2 does not update reservation value during negotiations. To investigate the effect of contract signing mechanism, we created the replica of the original proposed agent but MyGreedyAgentV3 does not optimize the contract signing with integer programming as proposed in the original approach. Instead, it schedules all the contracts to be signed and assumes all will be executed successfully. If there appears to be enough resources to meet with a contract's terms according to the schedule, the contract is signed, it is rejected otherwise.

To test the performance of our agent, we adopted two variants of experiments. In the first one, all the agents together took place in the same simulation run (i.e. the agent types, InsuranceFraudAgent, NVMAgent, MyGreedyAgent2, MyGreedyAgent, SAHAAgent, GreedyAgent, and MyGreedyAgent3 are all included in a single simulation run.). In the other setup, all agents were tested against each other in pairs. That is, the simulation runs for 40 steps in which only two factory agent types are chosen as competitors (e.g., For the InsuranceFraudAgent, some of the possible pairs are InsuranceFraudAgent-NVMAgent, InsuranceFraudAgent-MyGreedyAgent2, InsuranceFraudAgent-MyGreedyAgent etc.), and reported the average scores and compare their performances according to their average scores at the end of multiple runs.

4.2.1 The Experiment Results with All Agents Together in Same Simulation Run

In this experiment setup with all agents together in same simulation run. The score statistics are shown in the table below:

Table 6: Experiment Results

Agent type	Mean	STD	Min	25%	50%	75%	Max
InsuranceFraudAgent	-0.01	0.01	-0.04	-0.02	-0.01	0	0
NVMAgent	-0.17	0.33	-0.99	-0.04	-0.01	0	-0.01
MyGreedyAgent2	-0.10	0.22	-0.96	-0.03	-0.02	-0.01	-0.01
MyGreedyAgent	-0.11	0.22	-0.73	-0.03	-0.02	-0.01	0
SAHAAgent	-0.29	0.59	-0.99	-0.87	-0.09	-0.02	2.6
GreedyAgent	-0.23	0.27	-0.99	-0.31	-0.13	-0.03	0
MyGreedyAgent3	-0.78	0.27	-1	-1	-0.90	-0.04	-0.15

Note that STD stands for standard deviation, min and max represent the minimum and the maximum scores obtained by the specified agent type, and 25%,50%,75% represent the respective n'th percentiles, which is the smallest value greater than or equal to the n% of the scores (e.g., 0% is the minimum score, 50% is the median score, 100% is the maximum score.). The results indicate that our agent overall has achieved the third spot in terms of the mean score. While the variations of our agents MyGreedyAgentV2 and MyGreedyAgentV3 achieved the 2nd and 7th spots respectively. The slight score difference between MyGreedyAgentV2 and MyGreedyAgent implies that the reservation value during negotiations update does not have a significant impact on the average score while the significant score difference between MyGreedyAgentV3 and our agent shows that the contract signing optimization is an essential strategy in achieving a high result. Note that SAHA Factory manager agent can exploit consumer agent's acceptance strategy and sell the final products with high prices, therefore, the maximum score of the SAHA agent is extremely high. However, SAHA agent can only do this exploitation when it is assigned as a factory manager agent who can produce final products in its factory. Otherwise, it does perform below par. Therefore, it has a high standard deviation of score.

Student's t test statistics on the average scores is shown below. Where t indicates the t score indicates how likely the scores obtained are significantly different than the average scores, and p indicates the likelihood of the results obtained occured by chance and does not necessarily represent the reality. Therefore, the higher t value and the lower p value is desired.

Table 7: T Test Results for The Test Statistics

Agent 1	Agent 2	t	p
MyGreedyAgentV3	GreedyAgent	0.73	2.8e-09
MyGreedyAgentV3	SAHAAgent	0.56	3.24e-05
MyGreedyAgentV3	InsuranceFraudAgent	1	1.14e-14
MyGreedyAgentV3	NVMAgent	0.79	8.70e-11
MyGreedyAgentV3	MyGreedyAgentV2	0.87	6.13e-12
MyGreedyAgentV3	MyGreedyAgent	0.85	2.45e-11
GreedyAgent	SAHAAgent	0.23	0.01
GreedyAgent	InsuranceFraudAgent	0.70	7.01e-22
GreedyFactoryAgent	NVMFactoryAgent	0.47	3.53e-09
GreedyFactoryAgent	MyGreedyAgentV2	0.58	3.92e-11
GreedyFactoryAgent	MyGreedyAgent	0.59	1.92e-11
SAHAAgent	InsuranceFraudAgent	0.70	0
SAHAAgent	NVMAgent	0.46	1.99e-08
SAHAAgent	MyGreedyAgentV2	0.57	3.88e-10
SAHAAgent	MyGreedyAgent	0.56	5.07e-10
InsuranceFraudAgent	NVMAgent	0.23	0.014
InsuranceFraudAgent	MyGreedyAgentV2	0.54	1.94e-09
InsuranceFraudAgent	MyGreedyAgent	0.48	1.26e-07
NVMAgent	MyGreedyAgent2	0.43	6.56e-06
NVMAgent	MyGreedyAgent	0.42	1.35e-05
MyGreedyAgentV2	MyGreedyAgent	0.07	1

4.2.2 The Experiment Results with Agents in Pairs

In this experiment setup, the agents participate in simulations in pairs. The table below shows the statistics of the scores.

Table 8: Statistics of the Pair-Wise Experiment Results

Agent	Mean	Standard Deviation	Min	25%	50%	75%	Max
InsuranceFraudAgent	0	0.01	-0.05	-0.01	0	-3e-05	0
NVMAgent	-0.15	0.32	-1	-0.03	-0.01	0	0.08
MyGreedyAgent	-0.13	0.21	-0.74	-0.03	-0.02	-0.02	-0.01
GreedyAgent	-0.14	0.19	-0.85	-0.23	-0.04	0	0
SAHAAgent	-0.19	0.45	-1	-0.47	-0.06	-0.02	0.92

Note that these statistics are obtained by considering all pairs possible. For instance, in the first column, the score statistics are based on the score data for the pairs InsuranceFraudAgent-NVMAgent, InsuranceFraudAgent-MyGreedyAgent, InsuranceFraudAgent-GreedyAgent, InsuranceFraudAgent-SAHAAgent. The results show that, InsuranceFraudAgent is the best performing agent in terms of the mean scores and our agent has the second highest mean score. As in the case of all agents in the same simulation, SAHA agent has the highest maximum score because of the fact that it exploits the consumers under special circumstances. Otherwise it does not particularly get high score. This is why, it has a high standard deviation.

Student's T test results for the statistics are shown below:

Table 9: Student's T Test Statistics

Agent1	Agent2	t	p
GreedyAgent	InsuranceFraudAgent	0.52	0
GreedyAgent	MyGreedyAgent	0.38	0
GreedyAgent	SAHAAgent	0.23	0
GreedyAgent	NVMAgent	0.36	0
InsuranceFraudAgent	MyGreedyAgent	0.71	0
InsuranceFraudAgent	SAHAAgent	0.68	0
InsuranceFraudAgent	NVMAgent	0.19	0.01
MyGreedyAgent	SAHAAgent	0.51	0
MyGreedyAgent	NVMAgent	0.54	0
SAHAAgent	NVMAgent	0.5	0

4.3 Further Analysis and Conclusion

Our agent successfully made profits thanks to its adaptive reservation value system, which finds a smart balance between the level of greediness and altruism. Unlike most of the other agents in the simulation, it traded all types of products available to maximize the number of negotiation opportunities, instead of buying input products only and selling it after they are processed in the factory. When the supply for a product type is excessively high, compared to the demand, it acted greedy when buying products (i.e tried to bought as by the minimum price). Furthermore, when there's excess product in the inventory, and not much time left for the completion of the simulation, it tried to act altruistic to sell products with a relatively smaller profit margin.

This table shows that as the supply/demand ratio of a product increases, the buying negotiations success rate tends to decrease, because the agent acts greedy in buying negotiations when there is a surplus of the products. This behavior of the

Table 10: The Relation Between Negotiation Success Rates and Supply/Demand Ratios

Supply/Demand	Buying Negotiations Success Rate	Selling Negotiations Success Rate
1.23	0.08	0.95
1.26	0.16	0.88
1.76	0.18	0.82
1.90	0.23	0.85
2.28	0.43	0.86
2.56	0.29	0.73
42.36	0.09	0.85
123.4	0.09	0.88
inf	0.08	0.84
inf	0.12	0.81

agent is achieved by reservation value updates based on the supply/demand ratio. The inf refers to infinite (i.e, the demand is zero).

CHAPTER V

RELATED WORK

In this section, we provide an overview of related work in the fields of automated negotiation particularly about supply chain.

Putten *et al.* carried out a case study in automated negotiation applications for Vos Logistics [7]. In the study, there are 3PL and 4PL logistics service providers, where the former has it's own fleet and logistics plan and the latter doesn't have them. 4PL Companies receive transport orders from shippers and allocate orders in two different ways. First, closed group negotiations in which 4PL companies outsource the transport orders to 3PL companies are done by multi-issue bilateral negotiations in sequential fashion. Second, in open market negotiation where multiple 4PL companies negotiate with the open market, which are in association with outside carriers. The negotiations with the open market are done by means of combinatorial auction protocol.

For these negotiations, a negotiation agent elicits its user's preferences and negotiates on behalf of its users accordingly. Then, the human parties then evaluate the results and approve or reject the contracts reached by negotiation. Therefore, the study aims to introduce a negotiation support tool for human parties who lead the logistics companies. Moreover, due to the high overhead cost of the preference elicitation process, preferences on only two of the negotiation issues are elicited and the rest of them are negotiated without automation (i.e. via human-human negotiation).

Chen *et al.* implemented a dynamic supply chain simulation in which there are products introduced to the simulation and a supply chain of the particular product emerges accordingly [8]. Then, seller and buyer agents can enter and leave the supply chain in accordance with the rules determined by their owners. Seller and buyer agents negotiate with each other to sell and buy goods.

Two negotiation protocols are used in that simulation, pair-wise and auction protocol. In the pair-wise negotiation protocol, one of the agents in a supply chain sends an offer to another agent and negotiation starts if the other agent replies to the offer. The reply can be a counter-propose or accept. The negotiation between the parties is terminated when an agreement is reached. In the auction protocol, one of the seller agents inform an auctioneer about the good to be sold and highest desired price. Then, the auctioneer broadcasts the message to potential bidders and receives its bid. Finally, an auction is organized according to the seller's bid. This process is repeated until the seller reaches an agreement with one of the bidders, and auctioneer announces the results to the agents involving in the negotiation. The acceptance strategy is based on a constraint satisfaction problem in which agents have a set of constraints such as production capacity, budget and so forth, imposed by the owner of the agents. The agents evaluate the bids and accept if a bid satisfies the constraints, rejects otherwise.

Nguyen *et al.* stated that in supply chains, buyers have an advantage of negotiating with many suppliers concurrently and choosing the best agreements to satisfy their consumption schedule, which is an unfairness for seller agents [9]. To alleviate this issue, they introduced a flexible commitment model in which multiple agents negotiate in a bilateral fashion concurrently. There is a global deadline which imposes a time constraint on all ongoing negotiations in a simulation run. Before the global deadline, the negotiator agents can reach non-binding intermediate agreements, all of those are finalized after the global deadline is expired. Each party can decommit intermediate agreements by paying a fee to the other party involving in the agreement. The amount of fee is determined by several factors such as the amount of time passed after an intermediate agreement, the utility value of the agreement for the agent who decommits, and so forth. The flexible commitment model proposes a remedy for unfair burden imposed to the parties who is loyal to the contract because they receive a payment for compensation. Furthermore, it introduces a restriction on the decommiter party which might make decommitment a non profitable action.

Wang *et al.* developed an ontology based supply chain model to alleviate human interactions in supply chain [10]. The negotiations are based on ontologies which determine the organization of private and public information. Each agent has a private ontology, which stores the information private to the agent (i.e. not accessible to other agents). The private ontology is composed of modules which determine the negotiation strategies such as concession, bidding, acceptance, and opponent modelling. The public ontology on the other hand includes the public information such as the negotiation deadline, bid details such as item name, amount, proposer and so forth. The private ontology enables the agents to make their negotiation strategy confidential while the public ontology makes the exchange of information transparent to all agents in negotiation.

In the supply chain simulation, there are seller and buyer agents which negotiate which each other. A negotiation between the agents start when a buyer agent indicate its demand by sending a CFP (Call for proposal) to multiple sellers. The buyer and seller agents concurrently exchange their proposals in a bilateral fashion until an agreement is reached or negotiation deadline is passed.

[11] stated that the collaboration of supply chain agents is essential to achieve a dynamic, flexible and agile supply chain. They introduced supply chain coordination models for communication between supply chain agents. One of the proposed models is based on an ontology. In this model, there are modules namely task, knowledge management and policy models. The task and policy modules are the ones providing a communication mechanism to the agents through sending messages, each message contains the actions the receiver agent is supposed to perform. Another proposed

coordination mechanism proposed is based on negotiation and bid, in which there are knowledge management module contractor and contract manager agents. In this collaboration mechanism, the contract manager agents try to find contractors which act as trading partners. The knowledge management module is responsible for communication between the contract manager and contractor agents. Finally, software based collaboration mechanism is intended to deal with the changes in customer demands. The order management is handled with the policy module, and the communication between the policy module and business agents are performed with the knowledge management platform.

Rady proposed a multi agent system application in supply chain management [12]. The agents in the system form a network to work collaboratively in order to manage procurement, selling, production scheduling and financing. The network consists of seven agents interacting with each other. Each agent is assigned to separate tasks and their collaboration forms the supply chain management strategy. The brief description of tasks each agent handles is given below:

- **Manager agent:** This agent manages other agents and, collaboration between agents are done by interaction with this agent.
- **Sales agent:** This software agent is responsible for order fulfillment. It informs the manager agent to have the orders recorded, so inventory agent, factory agent and supplier agent can work in collaboration.
- **Inventory agent:** The inventory agent manages the inventory level.
- **Factory agent:** The factory agent controls the production schedule.
- **Supplier agent:** This agent is responsible for procurement.
- **Finance agent:** This agent is responsible for controlling the financial tasks such as recording expenditure, and revenue.

- **Database agent:** It keeps track of the repository data such as available stock, components and materials. The data are updated after the state of information is changed.

It's found out that a multi agent system approach for supply chain management (SCM) enabled the collaboration of agents and improved efficiency compared to traditional SCM methods.

Lin *et al.* proposed a solution for order fulfillment problem (OFP) which emerges upon receiving orders from customers and resolves when the goods are delivered to the customers [13]. The OFPs are composed of a set of constraints such as cost, quality, and total cycle time. In their study, the set of all OFPs supply chain agents encounter are modeled as a distributed constraint satisfaction problem (DCSP). The solution to the DCSP is the set of delivery schedules that satisfy all OFPs forming the DCSP, which is solved in two phases. The first phase is called local constraint resolution in which a seller agent tries to find a feasible solution to its local constraints by negotiating with a customer agent. During the negotiation, the supply of goods which are needed to satisfy the customer orders are estimated by using negotiation history with supplier agents. In the local constraint resolution phase, the orders are scheduled without collaborating with the supplier agents to ask them to alter their production schedule in order for the constraints to be satisfied. In the global constraint resolution phase, the seller agents negotiate with its suppliers to satisfy their OFP constraints. For instance, a seller agent might get an offer from its customer, which is infeasible with its supply schedule and may negotiate with their suppliers in order to satisfy the infeasible constraints. The evaluation of the proposed approach is compared with an existing method namely, the distributed coordination model (CCSP), to solve OFPs. It is shown that the proposed approach can achieve promising results with a small performance gap with CCSP.

Plinere *et al.* studied interactions in the manufacturing node of supply chain environments, adopting a multi agent system [14]. In the proposed method, there exists the following agents crystal seller, procurement, crystal buyer, production, and microchip seller agent. The procurement agent determines the amount of products needed for a specific time and records the needs to a database namely the raw material buyer DB which is used by the crystal buyer agent while negotiating with the crystal seller agent. The crystal buyer agent's negotiation strategy is structured based on the records in the raw material buyer DB to procure at right quantities at right time. The production agent is responsible for the production of microchips using crystals procured and recording the amount of finished goods to a database namely the finished goods DB. The microchip seller agent negotiates with the microchip buyer agent under the issues price, quantity and the delivery time. Th negotiations are performed in a bilateral fashion exchanging proposals and counter-proposals until an agreement between the negotiating parties are reached. In this study it's shown that adopting a multi agent system approach for manufacturing node of supply chains is a feasible and an effective approach.

Forget *et al.* introduced multi-behavior agents in a lumber supply chain environment in which the agents need to satisfy the orders from its customers [15]. To fulfill the orders, the agents adopt multiple planning behaviors. Contrary to mono behavior agents, which uses a fixed planning behavior, the multi behavior agents learns what planning behaviors to adapt in different situations. The planning behaviors the agents can learn are *direction reaction*, *reaction with anticipation*, and *negotiation*. The *direct reaction* planning behaviors are planning tasks based on local information, without any feedback. The *reaction with anticipation* planning behavior uses a model to predict the partner's actions and make the planning decision accordingly. The *Negotiation* planning behaviors are based on doing negotiations with the partners in supply chain in order to satisfy the needs. The agents adopting negotiation behavior exchange proposals and counter-proposals until an agreement is reached. To test the effectiveness of multi behavior agents, mono behavior and single behavior agents' performances, measured in latency of deliveries, several simulation runs with different levels of demands and supplies of products are run. It is shown that the planning behaviors are influenced by the simulation settings and each planning behavior can perform well in one setting and might not work well in another. Therefore, multi behavior agents can alleviate this problem by learning a good performing planning behavior based on the environmental setting and superior to mono behavior agents.

Xue *et al.* proposed an agent based negotiation platform for construction supply chain environments (CSC), in order to improve the efficiency of decision making while collaboration with other agents in the supply chain environment [16]. In their proposed multi agent system there are a set of agents called specialty agents which negotiate by adopting multi-attribute negotiation theory (MAN). In the system, a general contractor agent negotiates with multiple agents such as sub contractor agents, supplier agent, designer agent and owner agent which forms the CSC. The negotiations are done in a bilateral fashion and the issues for each negotiation thread are different (i.e. outcomes in one negotiation thread is not influential in the utility gained in other negotiation threads). During the negotiation, agents exchange offers until an agreement is reached, the bids are proposed according to feasible solutions for each agent and if no solution is found for one agent, it may send a *no response* message to another agent so the other agent may alter its offer. In case the other agent also does not have solutions left, the negotiation may end with a failure. Each agent evaluates the utility of the offers by taking its and the other party's utility into account to reach a Pareto optimal solution. In this work, its shown that the implementation of multi agent system is effective in CSC because agents have different preferences on the negotiation issues, and agent's utility functions also consider the other party's utility besides its own to reach a pareto optimal agreement.

Mandic *et al.* proposed multi agent system for supply chains to improve communication between supply chain entities, customer satisfaction, and inventory management [17]. In the proposed MAS, there are three agents namely, distributor agent, retailer agent, and customer agent. In the system, the distributor agent waits for a *distribution of goods* request from the retailer agent. When a request is received, it checks if the requested amount of products are obtainable from suppliers. If so, the goods are distributed to the retailer and the excess is backlogged. The retailer agent is responsible for selling goods to customers by doing negotiations and requesting goods from the distributor agent when the inventory is needed to be replenished. The customer agents produce stochastic orders in stochastic times. In this study, it is shown that introduction of multi agents systems to supply chain can decrease the overhead, and ease the modelling process, and doing negotiations with customers in a circumspect manner.

Hindriks *et al.* proposed an opponent modelling algorithm to predict opponent preferences to reach mutual agreements in a multi agent negotiation [18]. Assuming the opponent acts rational, the opponent uses a time based concession strategy, and its utility function can be approximated as a utility function, their algorithm modelled a probability of a bid offered at a certain time of a negotiation under a set of hypotheses. The hypotheses are a range of opponent profiles can be learned. During a negotiation when a bid is offered by the opponent, all defined hypotheses are updated based on Bayesian update rule. Thus, the estimator for preferences of the opponent is updated when a bid is received and among the bids in compliance with the offer strategy of an agent, the bid maximizing the other party's utility is offered. To evaluate the proposed strategy, negotiation agents with the Bayesian learning model is tested with the negotiation agents existing in the literature. Based on the experiment results, it's demonstrated that the Bayesian learning agent can improve the efficiency in the bidding process.

Wang *et al.* proposed an agent based mediation for to coordinate supply chains [19]. Their supply chain model is composed of, a service requester, a customer order service agent, service brokers and service providers. In the supply chain model, the service requester acts as the customer requesting orders. The orders are composed of a set of constraints such as quantity, delivery time and so forth. The customer order service agent distributes the customer request to multiple service providers to find a solution for the customer order constrains, as called *global solution*. A global solution for the customer is found in three phases. In the first phase, various service brokers negotiate each other to find a partial solution for the constraints they are assigned by the customer order service agent. In the second phase, the service brokers negotiate with the service providers to find mutually compatible solutions. Finally, the mutually compatible solutions are generated by the service broker agents.

Paolo *et al.* proposed an multi agent system based approach for electronic procurement based on production planning activities [20]. In their approach there is an E-marketplace environment in which there three types of actors namely, the system manager agent, the customer agent and the supplier agent. The system manager agent provides negotiation constraints for the customer and the buyer agent. The customer agent acts as the e-marketplace client who introduces its own negotiation strategies and place an order. The supplier agent acts as the supplier for the market and introduces its own negotiation strategies, planning information and catalogues. During the simulation run, there is one customer and many suppliers negotiating in a multilateral fashion. In each round suppliers submit their proposal and the customer can accept, reject, or ask for a new counter-proposal. The negotiation ends successfully when the customer accepts a counter proposal and an electronic contract is signed. The negotiation fails when the round based deadline is reached or the customer rejects a counter offer. The negotiation starts when the customer agent submits the order constraints. Then, the supplier agents find feasible solutions for the constraints using mathematical programming to construct the feasible offers. Throughout the negotiation, each suppliers send its offers adopting a time based concession strategy.

Jiao *et al.* proposed using a multi agent system for a collaborative manufacturing supply chain [21]. In the proposed supply chain environment, there is a customer agent, a configuration agent, a manufacturer agent consisting of multiple contract managers, and multiple suppliers. At each simulation run, the customer agent places an order to the configuration agent who decomposes the order constraints to the contract managers. In order to satisfy the constraints, each contract manager negotiates with many suppliers in a concurrent multilateral fashion. During a negotiation, each party exchanges offers until an agreement is reached. The offer exchange is done by proposing counter offers which can be accepted by the other party. The negotiation agents can be adjusted by humans to automate negotiations or humans can involve in the negotiations by making decisions in the negotiation. The system was tested on a case study involving Moxober, a mobile phone company, in which the utility of the agents were measured in terms of the feasibility of offers for the production schedule of manufacturing agents.

Um *et al.* proposed a mathematical model to maximize the total profits of the supply chain agents namely, supplier, producer and distributor [22]. In their study, distributor agents negotiate with the customers. During the negotiation, the distributor agent tells the demand information to the supplier and producer agent and pricing is determined according to the mathematical model which aims to maximize the total profit made by supplier, producer and distributor agents. The performance of their negotiation strategy is benchmarked against the Kasbah system [23] in which seller and buyer parties introduce the information delivery date, desired price and lowest/highest acceptable price to intelligent agents and the negotiation agents negotiate on behalf of the users. Their evaluation shows that the proposed negotiation strategy is slightly better than the Kasbah.

Hang *et al.* improved single machine earliness-tardiness model (SET) based supply chain scheduling by decreasing the necessary amount of information flows. In SET, there are orders, manufacturer mediator agents, which schedules and assigns the orders to manufacturers, suppliers, and supplier mediator agents, which places the supply orders for manufacturers. In the study, they proposed a modified version of SET scheduling approach namely collaborative single machine earliness-tardiness model (CSET) [24]. In this approach, there is a collaborative agent exchanging simultaneous information with all mediators in the system and make the scheduling accordingly. Between each mediator and collaborative agent pair, the information exchange is performed in a bilateral fashion where the collaborative agent send a scheduling request to the collaborative agent, if the collaborative agent validates the request, the request is sent down to further stream otherwise, the pareto agent makes a new request. Compared to SET approach, CSET decreases the information flow cost considerably and the level of improvement increases further as the number of participants in the supply chain increases.

Neubert *et al.* proposed a software agent for negotiation of parties in a supply chain which consists of buyers and suppliers. In the supply chain, a buyer agent finds a suitable supplier by using extended value chain management system. An EVCM consists of competency cells which receive buying requests from buyers and selling requests from suppliers in the supply chain and manages the matching of both parties. Then, the parties negotiate to reach an agreement for the transaction. To speed up the negotiation process and deal with the challenges of human-human negotiations, Neubert *et al.* introduce an automated negotiation approach in which both parties elicit their preferences on the negotiation issues by a comparison matrix of each issues and upper and lower bounds acceptable values for all negotiation issues. During the negotiation, each agents exchange offers in a bilateral fashion. The bidding strategy is based on the concession of the opponent, based on the concession, a target utility is calculated and the bid with the closest utility value to target value is offered.

The table below depicts the negotiation settings of the studies mentioned.

Table 11: Related work negotiation protocols

Work	**Protocol**	**Issue**	**Concurrency**
Putten	Bilateral	Multi-issue	Sequential
Papangelis	Bilateral or Auction based	Multi-issue	Sequential
Nguyen	Multilateral	Multi-issue	Concurrent
Wang	Multilateral	Multi-issue	Concurrent
Rady	Bilateral	Single-issue or multi-issue	Sequential
Lin	Multilateral	Multi-issue	Concurrent
Plinere	Bilateral	Multi-issue	Sequential
Forget	Bilateral	Multi-issue	Sequential
Hindriks	Bilateral	Multi-issue	Sequential
Paolo	Multilateral	Multi-issue	Sequential
Wang2	Bilateral	Multi-issue	Sequential
Jiao	Multilateral	Multi-issue	Sequential

CHAPTER VI

CONCLUSION

In this thesis, we present a novel factory manager agent for a supply chain simulation namely, the supply chain world, in which there are raw material suppliers, factory manager agents, and consumer agents. In this environment, the supplier agents provide raw materials to the simulation environment. Factory manager agents can obtain the raw materials by negotiating with the supplier agents, process them in their factory and selling them. Furthermore, the factory manager agent can also buy intermediate products which are the products processed by another factory manager agents but not purchased by the consumer agents. The consumer agents are the end customers, which buy final products only.

The proposed factory manager agent adaptively determines a good level of greediness and altruism based on the supply/demand ratio of each product. As the supply/demand ratio increases, our agent decreases its reservation value for the selling negotiations which enabled our agent to achieve more successful negotiations. The motivation behind decreasing reservation value is the fact that as the reservation value decreases, our agent can concede more which helps to reach an agreement with the other supply chain agents. When the supply/demand ratio is not high, our agent increases its reservation value for the selling negotiations which helps our agent to increase profits by acting greedy. As for the buying negotiations, our agent determines the reservation value of the negotiations involving a specific product by the consequences of negotiations, at each successful negotiation, the reservation value is increased to act more greedy in the further negotiations. The reservation value is decreased when there's a failed negotiation. Our agent adopts a time based concession strategy. For the selling negotiations, the opponents preferences are also taken into consideration by considering the opponent's bids during the negotiation. More value is put on the opponent's preferences as the supply demand ratio is increased. Finally, all of the offers in each negotiation thread is analyzed to estimate if there will be sufficient products to procure or sell. The estimation is based on the average quantities in the offers, the agents breach level, and successful negotiation rate by the corresponding agent. The reservation values are updated based on the estimations to utilize the concurrency of the negotiations. Finally, the contract signing strategy for buying negotiations and production scheduling is based on the optimization of the excess or shortage product cost. By optimizing the contract signing decisions by each product for and each time step, and reducing non-linear problems to multiple linear problems, the problem size is decreased to a practical problem size.

In order to evaluate the performance of our agent in supply chain world, we ran simulations involving top performing agents in ANAC 2019 Supply Chain Management League. The results show that our agent can achieve good performance as its competitors. In our tournament, it took the second place. Furthermore, we also study the effect of some of its components such as marginal update of reservation values within concurrent negotiations and the contract signing mechanism. According to our results, it can be seen that the proposed contract signing mechanisms play an important a role in our agent's performance, while the marginal reservation value update does not have a significant effect in performance. As future work, it would be interesting to explore the factory manager agents with various negotiatior strategies proposed in other leagues of ANAC. A set of negotiation strategies can be employed by our agent and it can test these strategies to other agents in the simulation. It may pick separate optimal strategies for each opponent agent in the simulation. This could make our agent's strategy even more flexible and yield even better results.

Bibliography

[1] S. Beth, D. N. Burt, W. Copacino, C. Gopal, H. L. Lee, R. P. Lynch, and S. Morris, "Supply chain challenges. building relationships," *Harvard business review*, vol. 81, pp. 64—73, 117, July 2003.

[2] Y. Mohammad, E. Viqueira, N. Ayerza, A. Greenwald, S. Nakadai, and S. Morinaga, *Supply Chain Management World*, pp. 153–169. 10 2019.

[3] K. Hindriks, C. Jonker, and D. Tykhonov, "The benefits of opponent models in negotiation," pp. 439–444, 09 2009.

[4] P. Faratin, C. Sierra, and N. Jennings, "Negotiation decision functions for autonomous agents," *Robotics Auton. Syst.*, vol. 24, pp. 159–182, 1998.

[5] T. Baarslag, K. Hindriks, and C. Jonker, *Acceptance Conditions in Automated Negotiation*, vol. 435, pp. 95–111. 01 2013.

[6] C. Yavuz, Süslü, and R. Aydogan, "Taking inventory changes into account while negotiating in supply chain management," pp. 94–103, 01 2020.

[7] S. Putten, V. Robu, H. Poutré, A. Jorritsma, and M. Gal, "Automating supply chain negotiations using autonomous agents: a case study in transportation logistics.," in *5th International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 1506–1513, 01 2006.

[8] Y. Chen, Y. Peng, T. Finin, Y. Labrou, R. Cost, B. Chu, J. Yao, and R. Sun, "A negotiation-based multi-agent system for supply chain management," *Proc. Agents'99 Workshop Agent- Based Decision-Support for Managing Internet-Enabled Supply Chain*, pp. 15–23, 06 1999.

[9] D. Nguyen and N. Jennings, "Managing commitments in multiple concurrent negotiations," *Electronic Commerce Research and Applications*, vol. 4, pp. 362–376, 12 2005.

[10] G. Wang, T. Wong, and X. Wang, "An ontology based approach to organize multi-agent assisted supply chain negotiations," *Computers Industrial Engineering*, vol. 65, p. 2–15, 05 2013.

[11] Hu Cui-hong, "Automated negotiation model of supply chain management based on multi-agent," in *MSIE 2011*, pp. 178–180, Jan 2011.

[12] H. A. Rady and El-Shorouk, "Multi-agent system for negotiation in a collaborative supply chain management," 2013.

[13] F. ren Lin and Y. yu Lin, "Integrating multi-agent negotiation to resolve constraints in fulfilling supply chain orders," *Electronic Commerce Research and Applications*, vol. 5, no. 4, pp. 313 – 322, 2006. Web-enabled Business and Customer Value.

[14] D. Plinere and A. Borisov, "A negotiation-based multi-agent system for supply chain management," *Scientific Journal of Riga Technical University. Computer Sciences*, vol. 45, 01 2011.

[15] P. Forget, T. Monteiro, S. D'Amours, and J.-M. Frayret, "Collaborative agent-based negotiation in supply chain planning using multi-behaviour agents," 01 2009.

[16] X. Xue, J. Lu, Y. Wang, and G. Shen, "Towards an agent-based negotiation platform for cooperative decision-making in construction supply chain," vol. 4496, pp. 416–425, 05 2007.

[17] K. Mandic and B. Delibašić, "Application of multi-agent systems in supply chain management," *Management - Journal for theory and practice of management*, vol. 17, pp. 75–84, 06 2012.

[18] K. Hindriks and D. Tykhonov, "Opponent modelling in automated multi-issue negotiation using bayesian learning," vol. 1, pp. 331–338, 05 2008.

[19] M. Wang, H. Wang, D. Vogel, K. Kumar, and D. K. Chiu, "Agent-based negotiation and decision making for dynamic supply chain formation," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 7, pp. 1046 – 1055, 2009. Distributed Control of Production Systems.

[20] P. Renna and P. Argoneto, "Production planning and automated negotiation for smes: An agent based e-procurement application," *International Journal of Production Economics*, vol. 127, no. 1, pp. 73 – 84, 2010.

[21] J. R. Jiao, X. You, and A. Kumar, "An agent-based framework for collaborative negotiation in the global manufacturing supply chain network," *Robotics and Computer-Integrated Manufacturing*, vol. 22, no. 3, pp. 239 – 255, 2006.

[22] W. Um, H. Lu, and T. Hall, "A study of multi-agent based supply chain modeling and management," vol. 2, pp. 333–341, 01 2010.

[23] A. Chavez and P. Maes, "Kasbah: An agent marketplace for buying and selling goods," in *Proceedings of the first international conference on the practical application of intelligent agents and multi-agent technology*, pp. 75–90, 1996.

[24] Y. Hang, S. Fong, and Z. Yan, "Cset automated negotiation model for optimal supply chain formation," *Int. J. Information Technology and Management Int. J. Information Technology and Management*, vol. 7, pp. 0–0, 03 2010.