Hamdi Bouhani Hybrid Approach for Concurrent Automated Negotiation Master ISG Tunis/ Juin 2017

UNIVERSITÉ DE TUNIS

INSTITUT SUPÉRIEUR DE GESTION

MÉMOIRE DE MASTER RECHERCHE

SPÉCIALITÉ

SCIENCES ET TECHNIQUES DE L'INFORMATIQUE DE DÉCISION

OPTION

INFORMATIQUE ET GESTION DE LA CONNAISSANCE

Hybrid Approach for Concurrent Automated Negotiation

Based on Evolutionary Learning Agent

HAMDI BOUHANI

SOUS LA DIRECTION DE:

NAHLA BEN AMOR PROFESSEUR, ISG TUNIS

RAOUIA AYACHI MAITRE ASSISTANTE, ESSEC DE TUNIS

LABORATOIRE : LARODEC

ANNÉE UNIVERSITAIRE 2016 / 2017

Acknowledgments

This master thesis could not have been completed without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them

Foremost, I am highly indebted to my supervisor Pr. Nahla Ben Amor for her suggestions and instructions that has served as the major contribution towards the completion of this work.

I would like to extend my thanks to my co-advisor Dr Raouia Ayachi for her kindness, support, and guidance throughout this masters degree project.

I would like to express my gratitude towards my family for the encouragement which helped me in completion of this work. Especially my parents Hefaithe and Mongia, my sisters Hela and Chaima, my friends Abdou, Oussama, Abir, Arij and Rania . . .

Thank you all.

Contents

Acknowledgments **i**

Introduction **1**

Part I : Theoretical Aspects **3**

1 Automated Negotiation **4**

1.1 Introduction 4

1.2 Classification of automated negotiation 6

1.2.1 Negotiation process 6

1.2.1.1 Automated level 6

1.2.1.2 Orientation type 7

1.2.1.3 Binding type 7

1.2.2 Negotiation structure 7

1.2.2.1 Protocol category: 7

1.2.2.2 Distribution type: 7

1.2.2.3 Attribution type: 7

1.2.2.4 Number of positions: 8

1.2.2.5 Mediation type: 8
1.2.2.6 Access type: 8
1.2.3 Theoretical foundations 8
1.2.3.1 Game-theoretic approach 8
1.2.3.2 Heuristic approach 8
1.2.3.3 Argumentation-based approach 9
1.2.4 Negotiation restrictions 9
1.2.4.1 Information situation 9
1.2.4.2 Time 10
1.3 Automated negotiation components 10
1.3.1 Negotiation domain 11
1.3.2 The preference profile 11
1.3.3 The negotiation deadline 12
1.3.4 The negotiation strategy 12
1.3.5 Opponent modeling related work 13
1.4 Conclusion 14

2 Multilateral automated negotiations protocols 15
2.1 Introduction 15
2.2 Definitions 16
2.3 Related work in multilateral negotiation 16
2.3.1 One to many negotiation 16
2.3.2 Concurrent negotiation 17
2.3.3 Many to many negotiation 18
2.3.3.1 Mediated negotiation 18

2.3.3.2 Non mediated negotiation 20
2.4 Qualitative comparison 20
2.5 Conclusion 22

Part II : Contributions **23**

3 New Concurrent Evolutionary Automated Negotiation **24**
3.1 Introduction 24
3.2 Evolutionary approach for concurrent negotiation 25
3.3 Concurrent evolutionary learning agent (CELA) 26
3.3.1 Evolutionary Learning Agent 27
3.3.2 Transformation of the Problem 27
3.3.3 DEIWO Optimization Algorithm 28
3.4 Bargaining position 31
3.4.1 The bargaining position estimation BPE 32
3.4.2 Illustrative example 34
3.5 Conclusion 35

4 Experimental study **37**
4.1 Introduction 37
4.2 Experimental scenarios 37
4.3 Integration of CELA IN Genius 44
4.4 Conclusion 46

Conclusion **47**

Appendix **49**

A Appendix 1 **50**

A.1 Bayesian learning agent (BLR) 50

A.1.1 Introduction 50

A.1.2 Prediction of Negotiation Deadline and Reservation Utility 50

A.1.3 Concession rate adjustment 51

References **53**

List of Figures

1.1 Classification of Automated Negotiations methods 5

1.2 Overview of the defining elements of an automated bilateral negotiation 10

1.3 Negotiation domain 11

2.1 One-sided multilateral negotiation 17

2.2 Architecture of the Necotiator framework 18

2.3 FANA model architecture 19

2.4 Architecture for multilateral automated negotiation 20

3.1 Architecture of CELA. 26

3.2 CELA strategy adjustment 27

3.3 Concurrent evolutionary learning agent. 31

3.4 The concession rate's impact on agent's offering curve 34

4.1 The average utility achieved by CELA and Concurrent BLR . . . 39

4.2 CNO of CELA, concurrent BLR and BPE 40

4.3 Buyer average utility in each thread 41

4.4 Buyer speed in each thread 42

4.5 Joint utility 42

4.6 Seller utility in thread 1, 2, 3 and 4 44

4.7 The most important methods of the genius Agent class. 45

4.8 Source code of the learning mechanism. 45

A.1 Prediction Process Demonstration 51

A.2 Prediction Process Demonstration 52

List of Tables

2.1 Qualitative comparison between the stated negotiation protocols . 21

3.1 Notations 32

3.2 Agent parameters. 35

3.3 Agent's historical offers. 35

4.1 The negotiation parameters 38

List of Algorithms

3.1 The bargaining position estimation Algorithm 34

Introduction

Negotiation is a critical activity in our life that we use to solve conflicts in our daily tasks. For example, friends negotiate to decide where to go out or which movie to watch. Businesses negotiate to close a deal or to sell their products. Attorneys negotiate to settle legal claims. The police negotiate with thug to free hostages. Nations negotiate to open their markets for trades. Negotiation is not a process reserved only for the experienced diplomat, salesperson, etc. It is something that everyone does daily.

Negotiations occur for one of two reasons; to do something that the negotiator can not do on his own, or to resolve a conflict between the parties. The aspects of negotiations can be applied to a large large number of perspectives including economics, psychology, political science, communication, law (Baarslag, Hendrikx, Hindriks, & Jonker, 2015). Because humans negotiate about so many things in different places like the work, the house, the supermarket, etc'. understanding the processes of negotiation is essential for anyone who interacts with other people. However, process of negotiation is a time and money consuming process. In the last couple of decades many researchers in the multi-agent systems community have been interested in the automation of the negotiation process.

In real word applications such as e-business and e-commerce, agents negotiate with each others dynamically. Therefore, when an agent is engaged in an automated negotiation, he could know his opponent (they met before) or not. In both cases, the agent should be able to autonomously negotiate with his opponents. For this reason, researchers in the multi-agent system community design many software agents, that negotiated on behalf of humans and taking into account their preferences. For example, (Hou, 2004) presented a learning agent that is able to learn the opponent's private information using nonlinear regression. (Sim, Guo, & Shi, 2007) proposed a learning agent called BLGAN that predicts the opponent's deadline using Bayesian Learning (BL), etc. But most of those intelligent agents are designed in the context of bilateral negotiation (negotiation between two software agents).

The aim of this work is to develop a new concurrent agent in the context of one-sided multilateral negotiation that negotiated with more than one agent simultaneously in real time using the multi-thread system. The multi-Thread system allows software agents to do simultaneous actions in the same time. the new learning agent so-called, concurrent evolutionary learning agent (CELA) uses a hybrid approach that combines between time dependency strategy and bargaining position. The time dependency tactic estimates the deadline and reservations points of an opponent agent and the bargaining position estimation adjusts the concession rate learned by the time dependency strategy. The learning problem is expressed in term of non-linear equations system in order to benefit from the recent researchers in optimization literature. To solve this learning problem, we will use the Differential Evolution Invasive Weed Optimization (DEIWO) (Zhou, Luo, & Chen, 2013).

This work is outlined as follows. The First chapter introduces the theoretical aspects and definitions of automated negotiation. Then, it exposes the basic elements for classifying automated negotiation. Chapter 2 provides a detailed state of the art related to multilateral negotiation protocols. Chapter 3 introduces the proposed agent CELA. Finally, chapter 4 highlights the results achieved in the experimental study.

Part I
Theoretical Aspects

Chapter 1

Automated Negotiation

1.1 Introduction

Distributed Artificial Intelligence (DAI) is a sub-field of Artificial Intelligence (AI) concerned with systems that consist of multiple independent entities or agents that interact in a domain, DAI has been divided into two sub-disciplines (STONE & VELOSO, 2000):

- *Distributed Problem Solving (DPS):* focuses on the information management aspects of systems with several components working together towards a common goal.
- *Multi-agent Systems (MAS):* deals with behavior management in collections of several independent agents able to execute their tasks autonomously.

Our focus in this work will be on MAS, where agents may have conflicting interests and thus, negotiation becomes a fundamental task. In fact, negotiation is a core activity in human life, and is studied by various disciplines including social psychology, game theory and economics (Baarslag et al., 2015).

An increasing number of software agent systems are being viewed as an encapsulated system that is able of flexible and autonomous action in order to meet its design objectives (Wooldridge, 1997). To achieve those objectives, an agent needs to interact with other agents. Therefore, they might need mechanisms that facilitate and coordinate the information exchange. One of these mechanisms is called "automated negotiation". It is a form of interaction between software agents, with conflicting interests and a desire to cooperate in order to achieve a mutually acceptable agreement (Rahwan et al., 2003).

Negotiation is a necessary daily activity, but it is time-consuming and expensive. As a result, there has been growing interest in the automation of the negotiation process in the last two decades.

The reminder of this chapter is organized as follows: Section 2, is dedicated to the basic elements for classifying automated negotiation methods on the basis of a literature investigation. Then, in Section 3, the automated negotiation components will be presented.

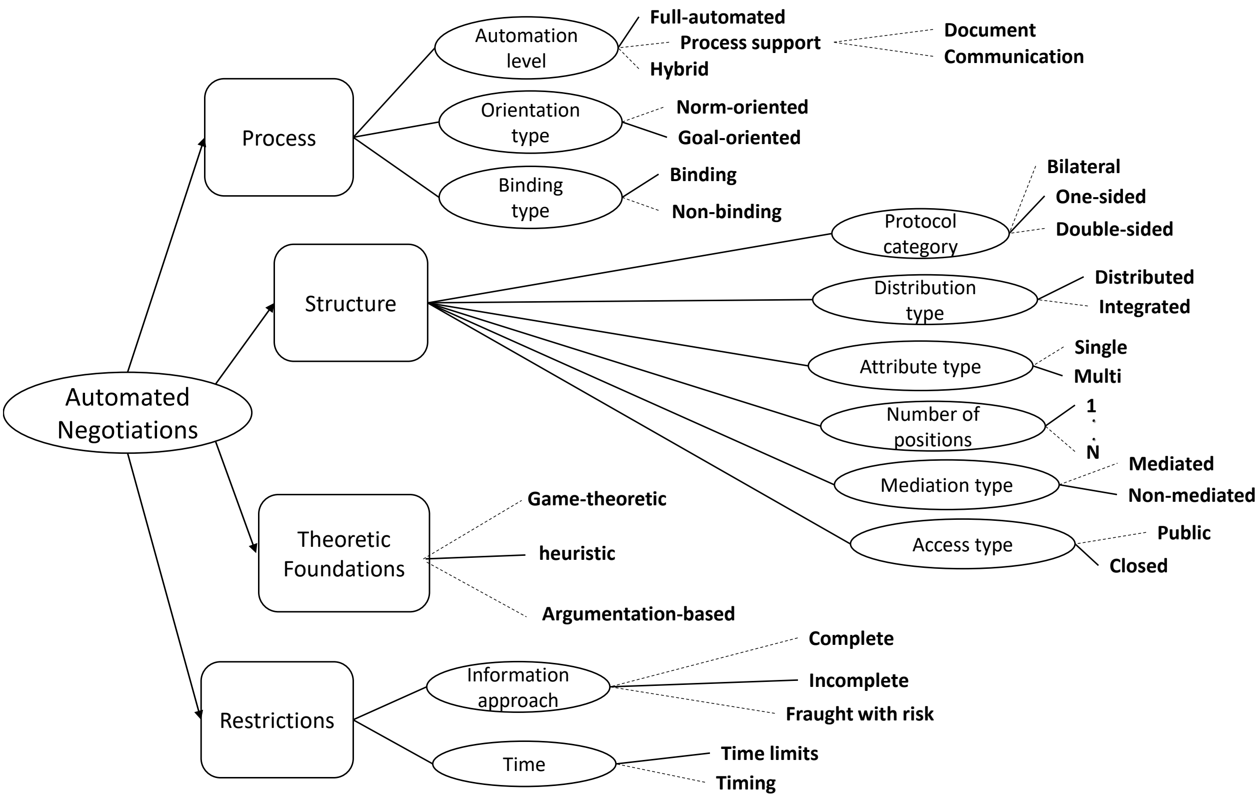


Figure 1.1: Classification of Automated Negotiations methods (Buttner, 2006)

1.2 Classification of automated negotiation

As outlined in the work of (Buttner, 2006), automated negotiations can be classified according to four criteria namely, *negotiation process*, *structure*, *theoretical foundations* and *restrictions* see Figure 1.1. In what follows we will detail each criteria.

1.2.1 Negotiation process

A process is a series of actions or steps taken in order to achieve a particular goal (OxfordUniversityPress, 2017). The automated negotiation process classified according to three criteria:

1.2.1.1 Automated level

Despite of the fact that the automation of negotiation was first proposed by (Davis & Smith, 1983), the automation level still raises questions in literature. There are three different automation levels in the automated negotiation process :

- *Full automated models:* a full-automated models should be well structured to make sure that software agents are able to negotiate autonomously. This type of negotiation process is result-oriented.
- *Process support models:* in the case of process support models, the humans participate in decision making step. Regarding to the final result, humans decide whether to accept or not the negotiation outcome. There are two types of process support models:
  - Communication oriented model: exchange of messages between the negotiations parties.
  - Document oriented model: exchange of documents between the negotiations parties.
- *Hybrid models:* Hybrid models is a combination between the previous models.

1.2.1.2 Orientation type

There are two types of Orientations: norm-oriented and goal-oriented negotiations. Norm-oriented negotiations are characterized by negotiators that have social commitments and obligations that they need to fulfill. On the contrary, the goal-oriented negotiators act on the basis of their goals and objectives.

1.2.1.3 Binding type

Automated negotiations can be a binding or non-binding type. Binding negotiations need an authentication of each participant in advance. In contrast, non-binding negotiations do not need that authentication.

1.2.2 Negotiation structure

Negotiation is a form of communication between two or more agents. In fact, this communication can be performed with different structures. In what follows we will detail each component of the negotiation structure.

1.2.2.1 Protocol category: It describes the number of negotiating agents and rules governing the interactions between themselves. There are three types of automated negotiation based on protocol categories:

- *Bilateral negotiation:* negotiation between two self interested partners or agents.
- *One sided negotiation:* negotiation between one buyer and many sellers or one seller and many buyers.
- *Double sided:* negotiation between many buyers and many sellers.

1.2.2.2 Distribution type: Automated negotiation can be distributed or integrated. In the distributed model, each negotiation agent tries to maximize his own profit (win-lose situation). In contrast, the aim of integrated model is to maximize the welfare of all agents (win-win situation).

1.2.2.3 Attribution type: The negotiation problem can be classified as single or multi-attribute negotiation according to the number of attribute. In the single attribute negotiation, the negotiators focus on specific attribute. However, in multi-attributed negotiation the negotiators take in consideration more than one-attributes.

1.2.2.4 Number of positions: The number of positions represents the quantity of independent objects in a single negotiation. A negotiation could be either single or multi-object.

1.2.2.5 Mediation type: There are two main negotiation types: mediated and non-mediated negotiations. The first one, there is broker (inter-mediator) negotiated on behalf of the participants, In the second one, the participants communicated directly between themselves.

1.2.2.6 Access type: A negotiation can be differentiated in a close or a public session. In a close session, the members of the negotiation process are fixed from the beginning. In the second one, a new member can take part dynamically.

1.2.3 Theoretical foundations

In the multi-agent Systems (MAS) literature, many decision and interaction mechanisms have been studied for automated negotiations, including game-theoretic approaches, heuristic approaches and argumentation-based approaches.

1.2.3.1 Game-theoretic approach

Game-theory is a branch of economics that tries to find the optimal strategy for the interactions between the self-interested economic agents by the analysis of the equilibrium conditions. But, the game-theoretic approaches have some limitation from the computational perspective (Dash, Jennings, & Parkes, 2003). Game theory assumes that the outcome space is known and that the agents have unlimited computational resources. However, these assumptions fail due to the fact that it is extremely hard for the humans to define their preference over outcomes (Jennings et al., 2001). Also, the communication and the processing capabilities are limited in the information systems of the realistic environment.

1.2.3.2 Heuristic approach

To overcome the aforementioned limitation of the game theory techniques, a number of heuristic approaches have being introduced. Instead of searching for an optimal solution, agents attempt to find acceptable sub-optimal solutions. In other words, heuristic approaches try to produce good enough solution rather than an optimal solution. Obviously, heuristic models overcome some of the game theoretic model drawback. However, it has some disadvantages, for example, the optimal solution may exist but can never be explored. To support the result of the heuristics technique and prove their robustness, we should evaluate this approach through empirical testing and evaluation.

1.2.3.3 Argumentation-based approach

Game-theory and heuristic approaches are limited on exchange of offers , counter offers, acceptance and rejection of offers. However, there is another approach with a more sophisticated form of interactions called *argumentation-based negotiation* ABN. This approach aims to overcome the limitations of game-theory and heuristic approaches by exchange of an additional information, called ”arguments”. An argument can be used to make or remind the opponent about a promise and to make a threat (for example walking away from the negotiation). Also, it can be used to attract and persuade the opponent by showing that the proposed offer is the best for his interest.

ABN framework is composed of external and internal elements:

- *External elements:* these elements that characterize the environment in which the agents interact. (e.g., the communication language, domain language and the interaction protocol).
- *Internal elements:* they are elements used by agents:
  - To evaluate, generate and select the arguments.
  - To alternate offers and counter offer in the negotiation process.

1.2.4 Negotiation restrictions

Negotiation is a difficult process that has many types of restrictions. But the information situation and time have huge influence on negotiation results.

1.2.4.1 Information situation There are three types of information situations in automated negotiations:

- *Complete information situation :* The negotiator has a prior knowledge about the preferences of their opponents (the preferences of their opponents are public).

- *Incomplete information situation :* The negotiator has no prior knowledge about the preferences of their opponents (the preferences of their opponents are private).
- *Information circumstances fraught with risk :* regarding the negotiation object, the opponent and the environment.

1.2.4.2 Time Automated negotiation can either have limited time (deadline) or illimited. In the first case, the agents have a deadline by when they must conclude the negotiations. For the latter case the agents don't have a deadline.

1.3 Automated negotiation components

The goal of automated negotiation is to create a software agent able to autonomously negotiate on behalf of humans. In order to understand the basic elements of an automated negotiation process, let us consider bilateral automated negotiations between self interested agents A and B. Figure 1.2 represents the main components of automated negotiation.

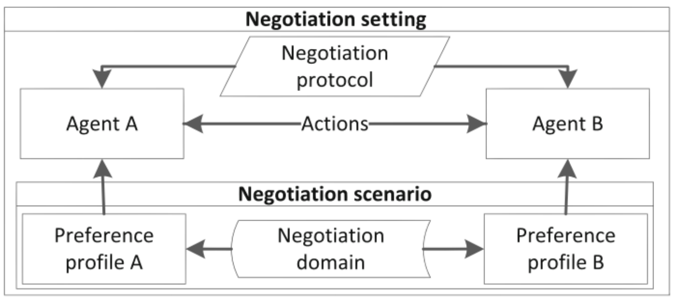


Figure 1.2: Overview of the defining elements of an automated bilateral negotiation (Baarslag et al., 2015).

The negotiation settings are composed of the negotiation protocol, which defines the rules governing the interaction between negotiating partners, and the negotiated scenario, that consists of the negotiation domain (also called outcome space) and preference profiles of the negotiating agents.

1.3.1 Negotiation domain

The negotiation domain denoted by Ω represents the set of all possible bids or negotiation outcomes or alternatives. The negotiation domain represents the space of agreement where the contract vector or contract offer exists. Figure 1.3 depicts the outcome space plot. The points represent the utility generated by agent's offers. the pareto optimal agreement is the outcome that satisfies all partners. In other words, there is no outcome beside pareto optimal outcome that is preferred by a partner without making another partner dissatisfies. The pareto frontier is the line that connects all the pareto optimal agreements.

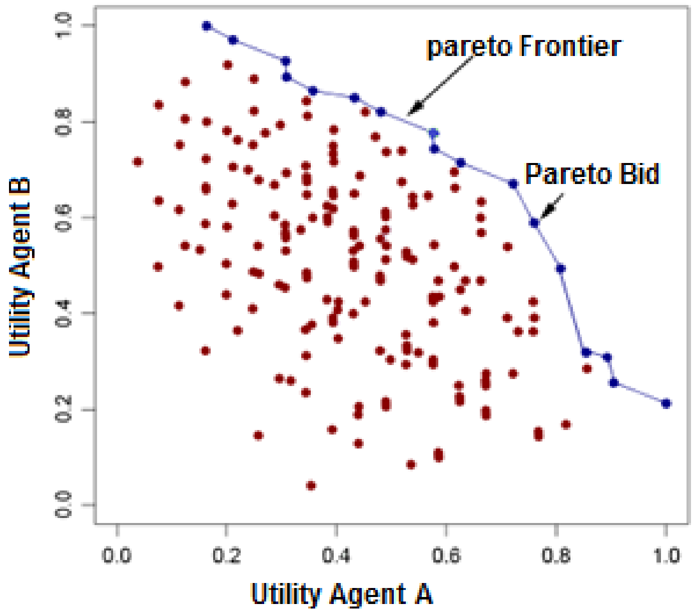


Figure 1.3: Negotiation domain (Baarslag et al., 2015).

1.3.2 The preference profile

The preference profile allows agents to rank the negotiation outcomes. Furthermore, by using it, the agents will be able to decide whether to accept or not the opponents proposals. The preferences of each agent are private information and they are modeled by an utility function. An utility function assigns a positive value to each negotiation outcome in the negotiation domain. For a negotiation alternatives $X =< x_1, \ldots, x_j >\in \omega$, $U(x)$ is expressed as follows:

$$U(X) = \sum_{j=1}^{n} w_j \times E_j(x_j) \tag{1.1}$$

where w_j are normalized weights (i.e, $\sum w_j = 1$) and E_j is the evaluation function for issue j.

1.3.3 The negotiation deadline

Time is a key factor in real world negotiations. As a matter of fact, time is limited due to the issues under negotiation, that could expire or one or more of the negotiators require a quick agreement. Therefore, without a deadline, agents may bargain for a long time and waste Unnecessary computation resource. In other words, without time pressure, the negotiators have no motive to accept an offer, and so the negotiation might go on forever. Also, with unlimited time an agent may simply try a large number of offers to learn the opponents preferences.

1.3.4 The negotiation strategy

The negotiation strategy correspond to the model used by agents to make decision and achieve their objectives (Radu, Kalisz, & Florea, 2013). It needs to be coherent to the negotiation protocols. There are three basic elements in the negotiation strategy:

- *The bidding strategy:* Also called the negotiation tactic or concession strategy, it is the set of functions that assigns a value for each issue in the offers. There are two type of concession strategy:
  - Time-dependent tactics: Let i be an agent and i' be his opponent. They are bargaining over an object characterized by J (J = $1 \ldots n$) issues. For each issue j ($j \in J$), both i and i' have a lower value called initial point (IP^i_j for agent i case and $IP^{i'}_j$ for agent i' case) and an upper value called reservation point (RP^i_j for agent i case and $RP^{i'}_j$ for agent i' case). More over, both agents have a deadline τ^i and $\tau^{i'}$ by which the negotiation must be concluded. The negotiation start at time $t = 0$ (first round), each agent gives his initial price then it is conceded in the next round $(t + 1)$ until $t = \tau$ when the agent gives his reservation value for each issue. The value of issue j ($x^i_j[t]$) is assigned according to this equation:

$$x^i_j[t] = IP^i_j + (RP^i_j - IP^i_j) * (\frac{t}{\tau^i})^{\alpha^i_j} \tag{1.2}$$

where α_j is the concession rate and $x^i_j[t] \in [IP^i_j, RP^i_j]$.

– Behavior-dependent tactics: In this type of tactics each agent adopts it concession strategy depending on his opponent behavior.

- *The acceptance strategy:* Negotiating agent accepts the opponent's offer only if the utility generated from that offer is equal or greater than the counter offer generated by the agent at the current round.

- *The opponent modeling:* Information is a key factor in automated negotiation but it is difficult to collect. Negotiators keep their parameters (e.g., preference profile, reserve value, etc.) secret, to make sure that the opponents don't exploit them. As a consequence, an agent needs to learn about its opponents. In negotiation opponent modeling is circled around three concerns (Baarslag, Hendrikx, Hindriks, & Jonker, 2016) :

  1 What does the opponent want?

  2 What will the opponent do?

  3 What type of player is the opponent?

1.3.5 Opponent modeling related work

The learning problem has been deeply studied in automated negotiation specially for the case of bilateral negotiation with a single issue. For example, (Hou, 2004) presented a learning agent that is able to learn the opponent's reserve value and deadline using nonlinear regression. (Sim et al., 2007) proposed a learning agent called BLGAN that predicts the opponent's deadline using Bayesian Learning (BL) than apply a genetic algorithm to generate a counter proposal. (Yu, Ren, & Zhang, 2013) proposed a learning agent that use a combination between Bayesian Learning (BL) and regression analysis in order to predict the deadline and the reserve point. They defined a set of hypotheses using the Bayesian Learning and updated those hypotheses based on the distance generated by regression analysis between the hypotheses about the reserve value and historical offers made by the opponent. (Zeng & Sycara, 1998) proposed a sequential decision making agent called BAZAAR that use a Bayesian Learning to model the opponent beliefs. but this agent does not have any mechanism to predict the opponent's deadline.

1.4 Conclusion

It is clear from the state of the art review, that there is no universal approach to deal with the automated negotiation problem. However, there is a set of approaches that try to find a solution to the problem of automated negotiation based on their assumptions concerning the environment in which agents interact. The next chapter discusses the major multilateral negotiation protocols and overview literature relative to it.

Chapter 2

Multilateral automated negotiations protocols

2.1 Introduction

Multi-agent systems are distributed systems, composed of a number of interacting entities (Jennings, Sycara, & Wooldridge, 1998). These entities are called software agents. Engineering those software agents means rigorously specifying the communication among themselves (Chopra & Singh, 2011). These communications are governed by negotiation protocols. The negotiation protocol defines the rules governing the interaction between the negotiating agents (Rosenschein & Zlotkin, 1994). It specifies the number of software agents and actions they are allowed to perform.

To the best of our knowledge, there are two different types of protocol categories that governs the negotiation between the self-interested agents, namely the bilateral negotiation and the multilateral negotiation protocols.

The remainder of this chapter is organized as follows. Section 2.2 gives the automated negotiation definition and the difference between bilateral and multilateral automated negotiation. Section 2.3 focuses the related work in multilateral automated negotiation.

2.2 Definitions

Automated negotiation can be defined as a discussion between two or more software agents with conflicting issues that try to reach a mutually acceptable agreement (Lomuscio, Wooldridge, & Jennings, 2001).

A bilateral negotiation may be viewed as a game, that is restricted to two self-interested agents (one buyer and one seller). In contrast, multilateral negotiation may be viewed as a game between more than two self-interested agents. In a multilateral negotiation, there are two types of protocol categories: one sided and double sided.

In the case of one-sided multilateral negotiation, there is the auction mechanism characterized by one buyer and many sellers or by one seller and many buyers. Moreover, there is another one-sided multilateral negotiation : the concurrent negotiation or simultaneous one-sided negotiation characterized by one seller (respectively one buyer) negotiated simultaneously with many buyers (respectively many sellers) using the multi-thread system.

Double sided multilateral negotiation is characterized by many buyers and many sellers negotiating with each other. Our focus in this chapter is multi-issues multilateral automated negotiation among multiple agents systems.

2.3 Related work in multilateral negotiation

This section discusses closely related works to multilateral automated negotiation, namely many to many negotiations, one to many negotiation and concurrent negotiation. In what follows we will present some of the most used negotiation protocols for multilateral negotiation.

2.3.1 One to many negotiation

As mentioned in the previous section, one-to-many negotiation consists of one seller negotiating with many buyers or one buyer negotiating with many sellers as we can see in (Figure 2.1). Several existing works in the literature uses one-to-many negotiation or auction mechanism. (Wurman, Wellman, & Walsh, 1998) proposed a configurable auction server called the Michigan internet auction bot. It is an internet-based auction system, that provides interfaces for both human and software agents. The Michigan auction bot supports single-resource auction

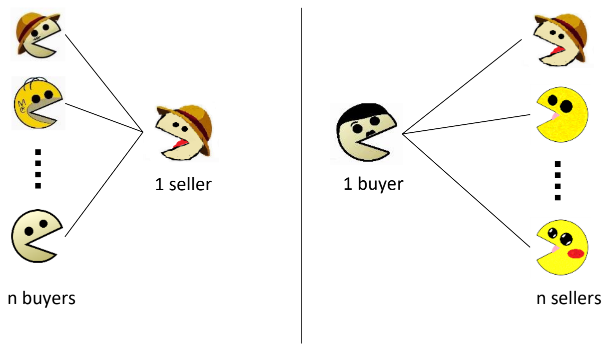


Figure 2.1: One-sided multilateral negotiation

mechanism. (Marsa-Maestre, Lopez-Carmona, Velasco, & de la Hoz, 2009) proposed a mediated auction based protocol for non-linear preference spaces (non-linear preference space is negotiation with multiple issues) generated using weighted constraints. However, according to (Tsuruhashi & Fukuta, 2015), auction approach can handle a considerable number of parties but can not be used to find agreement in multi-issues negotiations.

2.3.2 Concurrent negotiation

To secure a good deal, an agent may engage in multiple simultaneous negotiations, hence for this to be effective, the negotiating agents need to be carefully coordinated. In the case of the auction mechanism, the buyer proposes a set of offers and the sellers choose from that given set (one buyer and many sellers case). However, as outlined in the work of (Nguyen & Jennings, 2004) concurrent negotiation gives the buyer agent the opportunity to adopt different strategies in each thread because each thread represents an independent bilateral negotiation. Moreover, the outcome of a single negotiation can influence the behavior of the remaining concurrent negotiation. Concurrent negotiation is extensively studied in the literature. (Sim & Shi, 2010) presented dynamic simultaneous one-sided negotiation, that coordinates the negotiation based on the calculation of the expected utility of the resource provider proposals. Also, the consumers can break out the contract after paying off a penalty to his provider. (Sim, 2013) adopted the previous model for the case of cloud computing with some modifications in the coordination model and the concession strategy (they use bargaining position estimation strategy to modify the concession strategy). This modification enhanced the agent response to the market conditions. (Nguyen & Jennings, 2004) introduced a heuristic model to coordinate concurrent negotiation based on information obtained from previous encounters. (Tsuruhashi & Fukuta, 2015) proposed a negotiation framework called ”Necotiator” for one-to-many simultaneous negotiation. ”Necotiator” is a new framework for automated negotiation, that can handle Simultaneous Negotiations and other type of automated negotiation protocols (see Figure 2.2). (Yu, Wong, & Li, 2017) presented a negotiation protocol for multi-product suppliers selection problem. Furthermore, the proposed negotiation protocol is able to handle the purchasing of multiple products simultaneously.

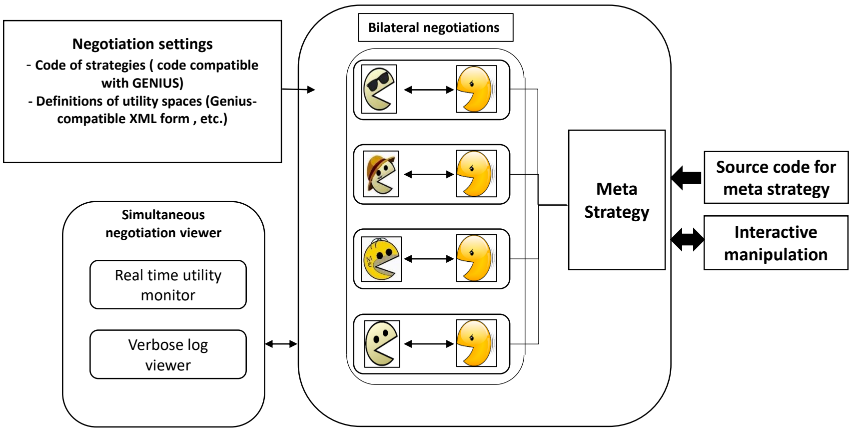


Figure 2.2: Architecture of the Necotiator framework (Tsuruhashi & Fukuta, 2015)

2.3.3 Many to many negotiation

In this subsection we are going to describe various types of many to many negotiation protocols, namely mediated negotiation protocols and non-mediated negotiation protocols.

2.3.3.1 Mediated negotiation

In the mediated negotiation, there is an inter-mediator that negotiates on the behalf of the software agents. In literature, researchers introduced many mediated negotiation protocols. For example, (Shojaiemehr & Rafsanjani, 2014) introduced a fuzzy system based automated negotiation called FANA (see Figure.2.3). FANA s composed with two components, namely FOM component and Decision maker component, FOM is responsible of generating the offers and the counteroffers for the agents. Decision maker component decides whether to accept or not the opponent's offers.

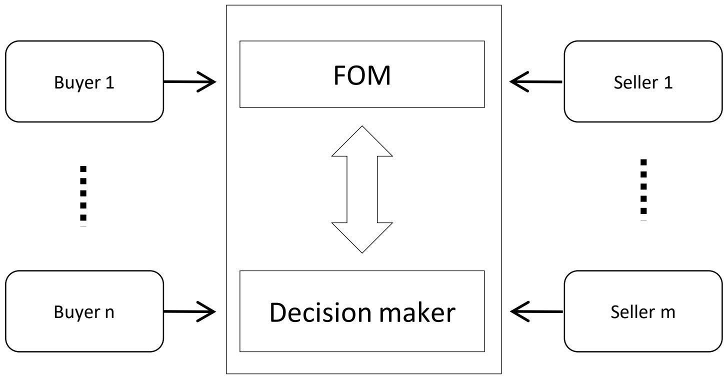


Figure 2.3: FANA model architecture

(Deochake et al., 2012) proposed a multi-threading system based automated negotiation. This system implemented negotiation protocol using the concept of weighted utility and is composed of two components, namely advertisement repository and condition-checker. The advertisement repository component stores all the agent information and the condition-checker component performs conditional matching between the buyers and the sellers. (Vij, Patrikar, Mukhopadhyay, & Agrawal, 2015) proposed linear programming and pattern matching based on many to many multilateral negotiation systems (see Figure.2.4). In this system, each user describes its preference profile in the beginning. After that, the system affects an agent to each user. Next, the offer generator component generates the offers. Finally, decision component decides whether to accept or not the offer.

Besides those protocols, Genius (JAVA framework for automated negotiations) offers some predefined implementation of a set of existing negotiation protocols (e.g., Simple Mediator Based Protocol, Mediator Feed Back Based Protocol,...etc) (Baarslag, Pasman, et al., 2016).

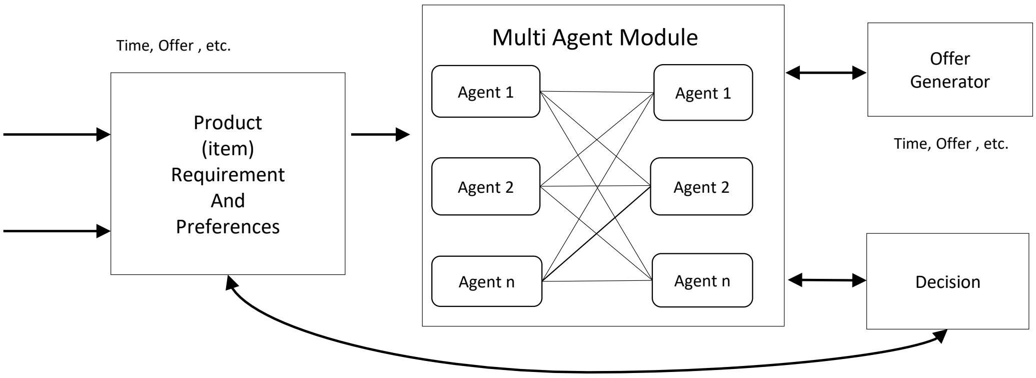


Figure 2.4: Architecture for multilateral automated negotiation

2.3.3.2 Non mediated negotiation

For the case of non-mediation negotiation, there are many approaches that have been proposed in the literature. (Aydogan, Festen, Hindriks, & Jonker, 2016) Proposed two multilateral turn-taking protocols called: Stacked alternating protocol (SAOP) and Alternating Multiple Offers Protocol (AMOP). In SAOP and AMOP each participant gets a turn per round. SAOP has one phase for each round but in AMOP there are two phases for each round Bidding phase and voting phase. In SAOP, an agent can walk a way from the negotiation if he didn't accept the bids of the other agent, but it is not the case in AMOP, the agent can leave only if the negotiation is finished. (Wong & Fang, 2010) proposed multilateral negotiation in supply chain management called ECNPro (the Extended Contract-Net-like multilateral Protocol) for handling buyer and seller bargaining in supply chain management. (Zheng, Dai, Sycara, & Chakraborty, 2016) proposed a generalization of the alternation-offer protocol, called sequential-offer protocol (alternation-offer protocol: an agent A proposes an offer and another agent B response to it by accepting or by proposing a counteroffer and they repeat the same process until a stopping criterion is satisfied).

2.4 Qualitative comparison

Table 2.1 represents the qualitative comparison between the negotiation protocols presented above.

Table 2.1: Qualitative comparison between the stated negotiation protocols

Protocol	Bilateral negotiation	One-to-many negotiation	Concurrent negotiation	Multi-issues	Many-to-many negotiation	Mediated many-to-many negotiation	Supports a meta-strategy
(Rubinstein, 1985)	×	-	-	×	-	-	-
(Wurman et al., 1998)	-	×	-	-	-	-	-
(Nguyen & Jennings, 2004)	×	-	×	×	-	-	×
(Marsa-Maestre et al., 2009)	-	×	-	×	-	-	-
(Wong & Fang, 2010)	×	×	-	-	×	-	-
(Deochake et al., 2012)	×	×	×	-	×	-	-
(Shojaiemehr & Rafsanjani, 2014)	×	×	-	-	-	×	-
Necotiator (Tsuruhashi & Fukuta, 2015)	×	-	×	×	×	-	×
(Vij et al., 2015)	×	-	-	×	-	×	-
(Aydogan et al., 2016)	×	×	-	×	×	-	-
SAOP (Aydogan et al., 2016)	×	×	-	-	×	-	-
AMOP (Aydogan et al., 2016)	×	-	-	×	×	-	-
(Zheng et al., 2016)	×	-	-	×	×	-	-
Genius (Baarslag, Pasman, et al., 2016)	×	-	-	×	×	×	-
(Yu et al., 2017)	×	-	×	-	-	-	×

2.5 Conclusion

In this chapter, we illustrated some of the existing negotiation protocols for the case of multilateral automated negotiation among software agents. On the basis of background presented in this chapter, in chapter 3, we will propose a new learning agent that is capable of adjusting its negotiation strategy based on bargaining position and the information learned from its opponent's proposals in the context of concurrent one-sided multilateral negotiation.

Part II
Contributions

Chapter 3

New Concurrent Evolutionary Automated Negotiation

3.1 Introduction

As mentioned in the previous chapter, there is a growing interest in many to many automated negotiations, especially concurrent one-sided automated negotiation in the case of cloud computing and resources allocation market. In a such negotiation, agents may simultaneously engaged multiple agents. However, to ensure the effectiveness of this approach, an agent should change his behavior dynamically. Therefore, agents need to carefully coordinate their negotiations strategies using coordinator that contains a meta strategy. In this chapter,we propose a generic new learning agent with a coordination strategy for concurrent automated negotiation, so called Concurrent Evolutionary Learning Agent(CELA), able to negotiate simultaneously with many agents and adjust it concession strategy based on the multi-thread system and on the information learned form its opponents.

This chapter is composed of 3 parts: Section 2 presents the new learning agent. Then, in Section 3 we will expose, the bargaining position.

3.2 Evolutionary approach for concurrent negotiation

In concurrent automated negotiation one agent engaged multiple opponents agents simultaneously using multi-thread system. Each negotiation thread represents an independent bilateral automated negotiation. Moreover, the software agent creates a sub-agent with the same preference profile to negotiate with the opponant agent. Finally, the main agent coordinates all the sub-agents using a meta-strategy.

Meta-strategy takes into consideration all simultaneous bilateral negotiations that the buyer agent performs. It can be used to manage certain resources of the buyer agent. For example, the budget, the total price of the agent purchased goods or services should be less or equal to the specified budget. In this work, we do not make any assumptions about the specific meta-strategy used by the agent. Which means that, the agent can use any meta-strategy.
The multi-Thread system allows software agents to do simultaneous actions in the same time. Using this mechanism a software agent can engage in multiple concurrent negotiations.

As shown in Figure 3.1, the architecture of the concurrent evolutionary learning agent (CELA) is composed of 4 components:

1 **Sub-agent creator**: This component is responsible for creating the sub-agents that are going to negotiate on behalf of the agent in the concurrent negotiation.

2 **Sub-agent**: The sub-agent component is a software agent that negotiates on behalf of his main agent or his master in independent negotiation thread (multi-issues bilateral negotiation). Each sub-agent is able to predict the deadline and the reservations points of his opponent and adjusts his bidding strategy.

3 **Coordinator**: The coordinator component is responsible of coordinating the negotiation threads based on bargaining estimation strategy presented by (Sim, 2013) in the context of could computing for multiple interrelated e-Markets.

4 **Historical offers**: The opponent historical offers contain the counteroffers

made by all opponents. The historical offers are the only information that is available about the opponents. Besides, it is necessary to predict their private's information.

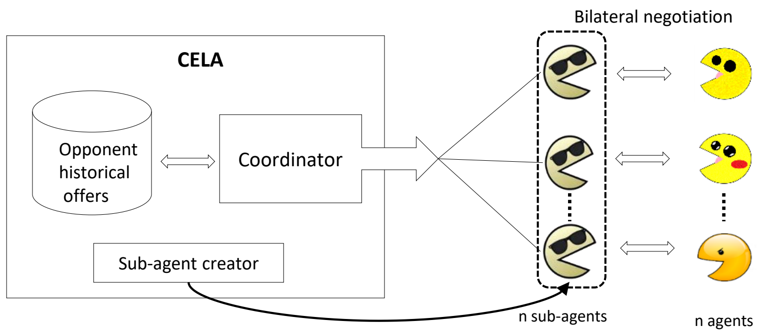


Figure 3.1: Architecture of CELA.

The concurrent evolutionary learning agent (CELA) uses a hybrid approach that combines between time dependency strategy and bargaining position. The time dependency tactic estimates the deadline and reservations points of an opponent agent and the bargaining position estimation adjusts the concession rate learned by the time dependency strategy.
The deadline and reservations points learning problem is expressed in term of non-linear equations system in order to benefit from the recent researchers in optimization literature. To solve this learning problem, we will use the Differential Evolution Invasive Weed Optimization (DEIWO) (Zhou et al., 2013). This learning agent called Evolutionary Learning Agent (ELA) adjusts its bidding strategy according to the predicted reservations values and deadline.

3.3 Concurrent evolutionary learning agent (CELA)

Figure 3.2 depicts the overall process executed by our proposed agent CELA. When an opponent agent wants to engage our agent in new negotiation, CELA agent creates a new thread and assigns to it a new sub-agent that negotiates with the opponent on behalf of the main agent. Then, the sub-agent applies the time-dependent strategy to predict the deadline and reserves points to adjust his concession strategy. After that, the sub-agent sends his concession rate to the main agent to adjust it according to the bargaining position.

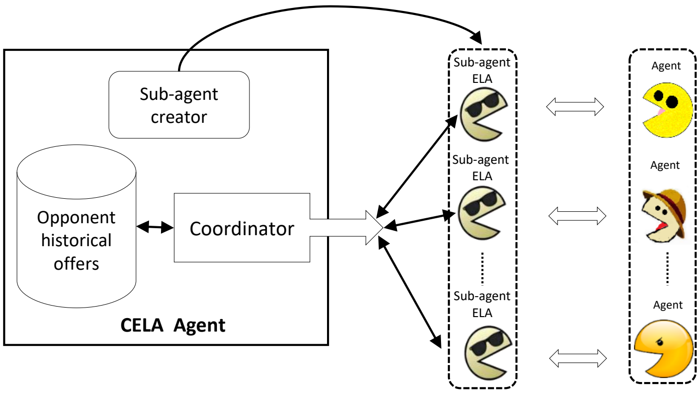


Figure 3.2: CELA strategy adjustment

3.3.1 Evolutionary Learning Agent

Evolutionary learning agent (ELA) uses an evolutionary learning approach to predict the competitor deadline and reserves points in a multi-issue bilateral negotiation and it only requires the information exchanged during the negotiation. ELA uses time-dependent tactics presented in (section 1.3.4) to adjusts his bidding strategy (see Equation 1.2). Moreover, ELA uses Differential Evolution Invasive Weed Optimization (DEIWO) algorithm to predict the deadline and reserves points of his opponent.

3.3.2 Transformation of the Problem

It is clear that the reserve point and deadline are inter-related terms because the agent give his reservation value by the deadline. furthermore, if we predicted the deadline of an agent, we can estimate his reservation values. Based on the Equation 1.2, the dependency between the deadline and reserve point are illustrated by the following equations.

$$RP_j^{i'} = \frac{X_j^{i'}[t] - IP_j^{i'}}{(\frac{t}{\tau^{i'}})^{\alpha_j^{i'}}} + IP_j^{i'} \tag{3.1}$$

$$\tau^{i'} = t/(\frac{X_j^{i'}[t] - IP_j^{i'}}{RP_j^{i'}[t] - IP_j^{i'}})^{\alpha_j^{i'}} \tag{3.2}$$

The proposed approach reduces the learning problem to a system of non-linear equations problem. This system of non-linear equations is buildup on the basis of Equation 3.2. Furthermore, the learning agent ELA should find the parameters that minimize the difference between the estimated value and the opponent's historical offers. Formally, we should minimize the following error function for each issue j under negotiation as follows.

$$f_j(\widehat{RP}_j^{i'}[t], \widehat{\tau}^{i'}[t]) = \sum_{k=1}^{t}(t/(\frac{X_j^{i'}[t] - IP_j^{i'}}{\widehat{RP}_j^{i'}[t] - IP_j^{i'}})^{\frac{1}{\alpha_j^{i'}}} - \widehat{\tau^{i'}})^2 \tag{3.3}$$

Equation 3.4 shows the reduction of the problem to a non-linear equations system:

$$\begin{cases} f_1(\widehat{RP}_1^{i'}[t], \widehat{\tau}^{i'}[t]) & = & 0 \\ f_2(\widehat{RP}_2^{i'}[t], \widehat{\tau}^{i'}[t]) & = & 0 \\ & \cdot & \\ & \cdot & \\ & \cdot & \\ f_J(\widehat{RP}_J^{i'}[t], \widehat{\tau}^{i'}[t]) & = & 0 \end{cases} \tag{3.4}$$

To solve a multi-objective optimization problem with the non-linear equations system, we need to find the parameters that minimize the following function:

$$min \sum_{j=1}^{J} f_j(\widehat{RP}_j^{i'}[t], \widehat{\tau}^{i'}[t]) \tag{3.5}$$

3.3.3 DEIWO Optimization Algorithm

Differential Evolution Invasive Weed Optimization algorithm is a recent evolutionary algorithm for solving non-linear equations system presented by (Zhou et al., 2013). DEIWO is a combination of two global optimization algorithms, namly: Invasive Weed Optimization algorithm IWO (Mehrabian & Lucas, 2006) and Differential Evolution algorithm DE (Storn & Price, 1997). In what follows, we will introduce DEIWO, his component and his steps. The first part of DEIWO is IWO and second part is DE.

- *IWO Optimization Algorihm*
  Invasive Weed Optimization algorithm is a numerical optimization method inspired from colonizing weeds and it follows the 4 following steps:

  1 *population initialization* : An initial population is randomly created. This population is composed with n initial feasible solution of the given problem also referred as a weeds. The weed structure represented as follows.
  $weed_n = (\widehat{\tau}^{i'}, \widehat{RP}^{i'}_1, \widehat{RP}^{i'}_2, .., \widehat{RP}^{i'}_J)$
  where $\widehat{\tau}^{i'}, \widehat{RP}^{i'}_1, \widehat{RP}^{i'}_2, .., \widehat{RP}^{i'}_j$ are the possible solutions for Equation 3.4.

  2 *reproduction* : In this step, weeds are allowed to generate new seeds (new solutions) depending on their fitness. The number of seeds produced by each weed is computed as follows:
  $seed_k = \frac{f - f_{max}}{f_{min} - f_{max}}(S_{max} - S_{min}) + S_{min}$
  where S_{min} and S_{max} are the minimum and the maximum of seeds that a weeds allows the generate. f is the fitness of the weed, f_{min} and f_{max} are, the minimum and the maximum fitness in the current generation.

  3 *spatial dispersal* : The produced seeds are normally distributed into the search area with mean equal to zero and varying variance to ensure that they remain close to their parent. The variance is reduced from a generation to another as follows:
  $\sigma_{curr} = \frac{(itr_{max} - itr)^n}{(itr_{max})^n}(\sigma_{initial} - \sigma_{final}) + \sigma_{final}$

  4 *competitive exclusion* : If the number of weeds is higher than its maximum number of population P_{max}. In that iteration, all the weeds are allowed to produce seeds. Thereafter, a mechanism for eliminating plants with poor tness activates.

- *DE Optimization Algorithm*
  The second part of DEIWO algorithm is Differential Evolution algorithm (DE). It is based on three operators summarized as follows:
  We begin with the mutations and crossover which allow as to create a new

generation for a specific population and we finish with the selection which decides to keep the new solution or not on the basis of its fitness.

- *Mutation:* The new mutant vector $WM_{k,G+1}$ is generated as follows:
  $WM_{k,G+1} = WM_{k,G} + F * (W_{best,G} - W_{k,G}) + F * (W_{r1,G} - W_{r2,G})$
  where $k = 1..P_{max}$ and $W_{k,G}$, $W_{r1,G}$ and $W_{r2,G}$ are a parents weeds. $W_{best,G}$ is the current generation best weed in term of fitness. F is a real and constant factor $\in [0 , 2]$ which instruments the expansion of the differential variation.

- *Crossover:* The crossover operator used to randomly merge mutant weeds $WM_{kd,G+1}$ with parent weeds $WM_{kd,G}$ to generate new seeds $WC_{kd,G+1}$.

  $$C_{kd,G+1} = \begin{cases} WM_{kd,G+1}, & if(rand_d = [0,1] \leqslant CR) \\ & or \quad d = d_{rand} \\ WM_{kd,G}, & otherwise \end{cases}$$

  where $d_{rand} \in [1, nd]$ is random value and nd is the dimension of the weed W. CR is a crossover rate constant between 0 and 1.

- *Selection:* The selection of the individual for the next generation is based on the fitness function $f(X)$ is done as follows:

  $$WC_{kd,G+1} = \begin{cases} WC_{kd,G+1}, & if \quad f(WC_{k,G+1} \leqslant WC_{k,G}) \\ WM_{k,G}, & otherwise \end{cases}$$

- *Concession Rate adjustment*
  In time-dependent strategy, there are two cases. In the first case, the deadline of the opponent is smaller than the agent deadline. In the second case, the deadline of the opponent is greater than the agent deadline. For each case, the agent uses different formula to adjust his concession rate α.

  - **Case 1** ($\widehat{\tau^{i'}} < \tau^i$): The concession rate $\alpha^i_j[t]$ is computed as follows:

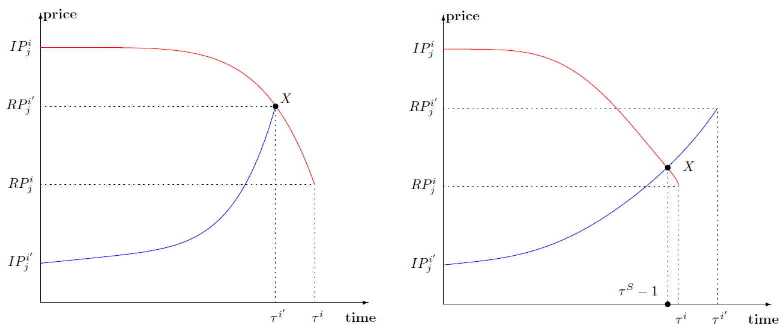
    $$\alpha^i_j[t] = \left| log_{\frac{\widehat{\tau^{i'}}-t+1}{\tau^i-t+1}} \left(max \left\{ 0, \frac{x^{i'}_j[t-1] - \widehat{RP}^{i'}_j}{x^{i'}_j[t-1] - RP^i_j} \right\} \right) \right|$$

    Figure 3.3(a) represents the concession rate behavior of both agents i and i' when agent deadline is bigger than the opponent deadline. Like shown in the Figure an agreement is meet only if the two curves of concession intersect. In this case, the best optimal strategy adopted by the learning agent is to sit and wait until the deadline of the opponent agent i' is met. The agent i should give to his opponent an offer with an utility equal to or higher than it reservation utility RU.

- **Case 2** ($\widehat{\tau}^{i'} > \tau^i$): The concession rate $\alpha^i_j[t]$ is computed as follows:

$$\alpha^i_j[t] = log_{\frac{\tau_i - t}{\tau_i - t + 1}} \left(\frac{IP^{i'}_j - x^{i'}_j[t-1]}{RP^i_j - x^{i'}_j[t-1]} + \frac{\widehat{RP^{i'}_j} - IP^{i'}_j}{RP^i_j - x^{i'}_j[t-1]} \left(\frac{\tau_i - 1}{\widehat{\tau_{i'}}} \right)^{\alpha^{i'}_j} \right)$$

Figure 3.3(b) represents the concession rate behavior of both agents i and i' when agent deadline is smaller than the opponent deadline. In this case the agent should have an agreement at round ($t = \tau - 1$) because at ($t = \tau$) the agent gives his reservation value.



(a) Case 1:($\widehat{\tau}^{i'} < \tau^i$) (b) Case 2:($\widehat{\tau}^{i'} > \tau^i$)

Figure 3.3: Concurrent evolutionary learning agent.

3.4 Bargaining position

Bargaining position gives the possibility to model the market and the competition between negotiating agents in the market (Sim, 2013). In other words, bargaining position denoted by B_p, gives us an idea about the position of the agent in the negotiation. In fact, the agent could be in an advantageous position (favorable market) or in a disadvantageous position (unfavorable market). Therefore, the negotiating agent adjusts his concession strategy according to his bargaining position. Notations used in this sub-section are given in Table 3.1.

Table 3.1: Notations

Symbols	**Descriptions**
t	The current round $t \in \{0..n\}$.
$P^i(t)$	Opponent i proposal in round t.
$\Delta^i(t)$	Difference between the initial and the current proposal.
$\delta^i(t)$	Difference between the previous and the current proposal.

An agent bargaining position B_p can be determined in 3 steps as follows.

- **Step 1:** In the first step we measure the difference between the initial offer and current offer of an agent i.

$$\Delta^i(t) = P^i(0) - P^i(t) \tag{3.6}$$

- **Step 2:** In the second step we measure the difference between the previous offer and current offer of an agent i.

$$\delta^i(t) = P^i(t-1) - P^i(t) \tag{3.7}$$

- **Step 3:** In the last step the bargaining position is measured by the factor of the average of $t * \delta^i(t)$ and the average of $\delta^i(t)$, more formally:

$$B_p(t) = avg(\frac{t * \delta^i(t)}{\Delta^i(t)}) \tag{3.8}$$

In this work, we will measure the bargaining position from a buyer point of view.

3.4.1 The bargaining position estimation BPE

The sub-agent adjusts it concession rate based on the learned private information (deadline and reservation values) of his opponent. Then, the bargaining position estimation adjusts the concession rate α_j^{ELA} for each issue j ($j \in J$) based on B_p. If $B_p(t) \gg 1$, then the agent is in advantage situation so he does not need to modify his concession strategy. If $B_p(t) \ll 1$, then agent is in disadvantage situation, so we need to adjust his concession strategy. Let $\delta(B_p(t) - B_p(t-1))$ be the difference between the bargaining position from t to $(t-1)$.

- If $\delta(B_p(t) - B_p(t-1)) > 1$, then the agent should adopt slower bidding strategy because he is in favorable position.
- If $\delta(B_p(t) - B_p(t-1)) < 1$, then the agent should adopt faster bidding strategy because he is in an unfavorable position .

Bargaining position estimation is outlined as follows.

$$\delta^t = B_p(t) - B_p(t-1) \tag{3.9}$$

where δ^t is the change in the bargaining position B_p

$$\alpha^{t+1} = \begin{cases} MAX(\alpha^{ELA}, \alpha^{ELA} + \delta^t * (\alpha^{ELA} - \alpha^{min})) & 0 < \alpha^t \leqslant 1 \\ \alpha^{ELA} + \delta^t & 1 < \alpha^t \leqslant \infty \end{cases} \tag{3.10}$$

where α^{ELA} is the concession rate adjusted by the sub-agent. α^{t+1} is the concession rate for the next round. α^t is the concession rate of the opponent.

- **Case of** $(1 < \alpha^t \leqslant \infty)$**:** In this case, the opponent agent is adopting a boulware or conservative strategy see Figure 3.4. In other words, he gives his reservation value only when the time is almost exhausted. An agent adjusts his strategy based on δ^t if the agent is gaining its B_p then δ^t is positive and it will adjust α^{t+1} to a slower concession strategy, else if the agent is losing its B_p then δ^t is negative and it will adjust α^{t+1} to a faster concession strategy.
- **Case of** $(0 < \alpha^t \leqslant 1)$**:** When the opponent agent is a conceder, it gives his reservation value very quickly to his opponent but when $\delta^t = 1$ then the agent is adopting a linear strategy. Therefore, the agent concedes using the same concession rate during the negotiation see Figure 3.4.
  The concession rate of the opponent is computed as follows:

  $$\alpha^t = log_{\frac{t}{t-1}}\left(\frac{x_j^{opp}[t] - IP_j^{opp}}{x_j^{opp}[t-1] - IP_j^{opp}}\right), t \geqslant 3 \tag{3.11}$$

  where $x_j^{opp}[t]$ is the value of issue j proposed by opponent at round t and IP_j^{opp} is initial value of issue j proposed by opponent.

The overall algorithm of the bargaining position is outlined in Algorithm 3.1.

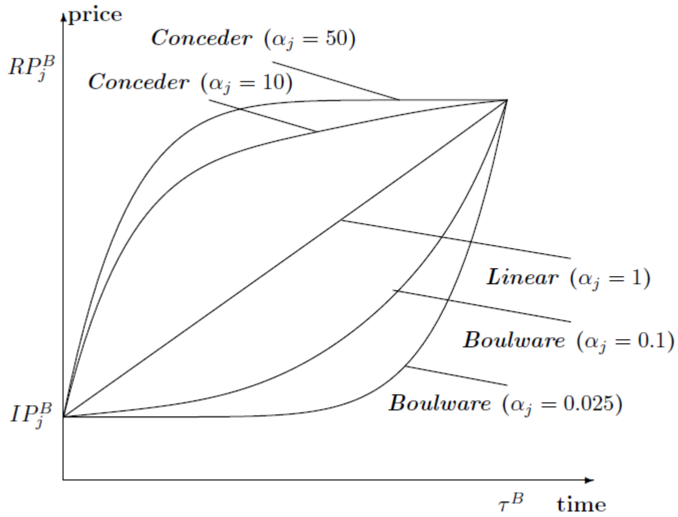


Figure 3.4: The concession rate's impact on agent's offering curve

Algorithm 3.1 The bargaining position estimation Algorithm

Require: the opponent's historical offers, set of the α^{ELA} determined by sub-agent ELA;
Ensure: Concession rate α^{t+1} at round $t+1$;

Compute the bargaining position $B_p(t)$ (see Equation 3.8);
if $B_p(t) \gg 1$ **then**
  return α^{ELA} ;
end if
if $B_p(t) \ll 1$ **then**
  /* Estimate the bargaining position */
  $\delta^t \leftarrow B_p(t) - B_p(t-1)$;
  if $0 < \alpha^t \leqslant 1$ **then**
    $\alpha^{t+1} \leftarrow MAX(\alpha^{ELA}, \alpha^{ELA} + \delta^t * (\alpha^{ELA} - \alpha^{min}))$;
  end if
  if $1 < \alpha^t \leqslant \infty$ **then**
    $\alpha^{t+1} \leftarrow \alpha^{ELA} + \delta^t$;
  end if
end if
return α^{t+1};

3.4.2 Illustrative example

In order to illustrative example of the bargaining position estimation, we will consider a bilateral negotiation between a sub-agent buyer created by CELA and a seller agent negotiated about the price of a product. Table 3.2 gives the parameters of the negotiation and Table 3.3 gives the historical offers made by the two

agents. The buyer sub-agent engages seller in bilateral negotiation in separate thread. The sub-agent ELA computes the seller concession rate as follows:
$\alpha_{price} = log_{\frac{2}{1}}\left(\frac{7.99-8}{7.97-8}\right) = 4.$
After that, ELA employs the proposed DEIWO based learning method to predict the seller deadline and reserve points. Let us consider a population containing three weeds, w_1, w_2,...,w_n . To determine w_1's fitness, we need first to compute $f_{price}(w_1)$. For example, we take the value of the price in the w_i equal to 7 and the value of deadline equal to 6.
$f_{price}(w_1) = f_{price}(7, 6) = \left[1/\left(\left(\frac{7.99-8}{7-8}\right)^{\frac{1}{1}}\right) - 6\right]^2 + \left[2/\left(\left(\frac{7.97-8}{7-8}\right)^{\frac{1}{1}}\right) - 6\right]^2$
The sub-agent chooses the weed with the best fitness. To compute the new concession rate α as stated in the subsection(3.3.1). Then the buyer sub-agent sends the concession rate to the coordinator to compute the $B_p(t)$. BPE computed as follows: $B_p(t) = avg(\frac{1*(8-7,99)}{8-7.99} + \frac{3*(7.99-7.97)}{8-7.97}) = 0.83 < 1$ so the buyer agent is in unfavorable position. Then we compute δ^t as follows:
$\delta^t = B_p(1) - B_p(3) = -0.17$
The seller concession rate is $1 < 4 \leqslant \infty$. For this reason, α^{t+1} computed using the second case of the equation 3.10.
$\alpha^{t+1} = \alpha^t + \delta^t = 0.53$

Table 3.2: Agent parameters.

Agent	deadline	reserve price	initial price	Concession rate α
Seller	7	4	8	4
Buyer	5	7	3	0.7

Table 3.3: Agent's historical offers.

Round	0	1	2
Seller	[8]	[7.99]	[7.97]
Buyer	[3]	[4.2]	-

3.5 Conclusion

In this chapter, we proposed a model that can be easily implemented that combines between behavior and time dependency tactic. The presented model is able to engage multiple agents in simultaneous negotiation and coordinates carefully the sub-agent in each independent negotiation thread. In next chapter, we will present experimental simulation results in order to evaluate its performances.

Chapter 4

Experimental study

4.1 Introduction

In order to evaluate the performance of the proposed CELA agent, we carry out a set of experiments on several simulated data set and we compare it to an agent with incomplete information which adapts its concession strategy based on bargaining position estimation BPE used by (Sim, 2013) and the Bayesian Learning Agent (BLR) of(Zhang, Ren, & Zhang, 2014) in terms of combined negotiation outcome (CNO), agent average utility (AU), agent joint utility (JU) and average negotiation speed (ANS).

This chapter is composed of 2 parts: Section 2 is dedicated to the experimental study. then, Section 3 will present the integration of CELA in Genius.

4.2 Experimental scenarios

To evaluate our concurrent negotiation model, simulations need to be performed. In this section, we will evaluate our new learning agent CELA through multiple simulations and scenarios, without making any assumptions about the meta-strategy. Therefore, our model can be implemented with many different meta-strategies.
The tests were performed on a Windows 10Pro 64-bit Operating System com-

Table 4.1: The negotiation parameters

Issues' parameters		
min_j	1	
max_j	100	
Number of issues	4	
Preferences	0.25	
Agents' parameters		
Parameter	*Buyer*	*Seller*
IP_j	$[RP_j^B, max_j]$	$[min_j, RP_j^S]$
RP_j	$[min_j + 5, min_j + (max_j - min_j)/2]$	$[RP_j^S + 10, max_j - 5]$
α_j	$[0.1, 5]$	
τ	$[10, 100]$	

puter equipped with an Intel Core i3-2328M CPU 2.20 GHz and 4GB(3.90 GB)of RAM.

We have conducted a set of experiments and each experiment corresponds to a comparison between CELA and other agents using different evaluation measures.

- **Experiment 1:** In order to evaluate our negotiation model, we propose to compare CELA to the following agent:
  - The Bayesian Learning Agent (BLR) based agent (Zhang et al., 2014) that learns its opponent's reserve utility and deadline in order to adjust its concession strategy(see Appendix A1). After extending it to concurrent BLR (Hybrid approach that combines time dependency strategy BLR and behavior dependency strategy BPE).

  We run the experiment 50 times with randomly generated scenarios. In each concurrent negotiation, a sub-agent engages in 10 different agent simultaneously negotiations on behalf of its main agent. Each sub-agent negotiates in an independent negotiation thread. The randomly generated experimental settings are shown in Tables 4.1. The result of the experiment are shown in Figure 4.1. This figure depicts the comparison between CELA and the concurrent BLR scenarios in term of average utility. In fact, CELA outperforms concurrent BLR because it can learn multiple reservation values for multiple issues simultaneously. In another word, CELA determines for each issue its own concession rate α. However, BLR learns the reservation utility of his opponent to determine a single concession rate α for each issue. In the case of CELA, the coordinator adjusts each α_j for each issue j according to the bargaining position of the agent in the negotiation round. In the other hand in the case of concurrent BLR, the coordinator adjusts a single α for all issues according to the bargaining position of the agent in that round.

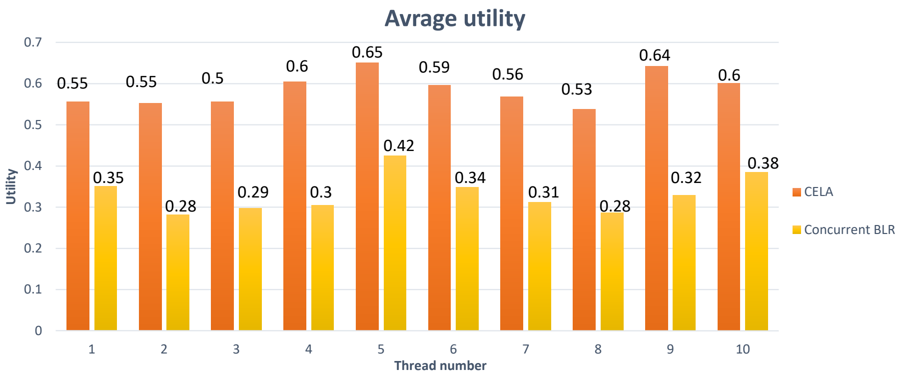


Figure 4.1: The average utility achieved by CELA and Concurrent BLR

- **Experiment 2:**
  In order to evaluate our negotiation model, we propose this time to compare CELA to the following agents:

  1 An agent with incomplete information which adapts its concession strategy based on bargaining position estimation BPE used by (Sim, 2013).

  2 The Bayesian Learning Agent (BLR) based agent (Zhang et al., 2014) that learns its opponent's reserve utility and deadline in order to adjust its concession strategy after extending it to concurrent BLR (Hybrid approach that combines time dependency strategy BLR and behavior dependency strategy BPE).

The experiment is repeated 50 times with randomly generated scenarios. Each negotiated agent is engaged in 4 different simultaneously negotiations. Each sub-agent negotiates on behalf of its main agent in an independent negotiation thread. Tables 4.1 outlines the parameters used in our experimental study.

In this experiment, we will use the combined negotiation outcomes (CNO) as a performance measure, in order to evaluate our negotiation model. CNO is measured as follows:

$$CNO = \frac{(AU * SR)}{ANS} \tag{4.1}$$

where AU is the average utility, ANS[1] is the average negotiation speed, where t^i is the time of the agreement. and SR[2] is the success rate.
The results of the experiment are shown in Figure 4.2. In fact, CELA outperforms concurrent BLR because it can learn multiple reservation values for multiple issues simultaneously. In another word, CELA determines for each issue its own concession rate α. However, BLR learns the reservation utility of his opponent to determine a single concession rate α for each issue. After that CELA coordinator adjusts each α_j for each issue j according to the bargaining position of the agent in the negotiation round. In the other hand in the case of concurrent BLR, the coordinator adjusts a single α for all issues according to the bargaining position of the agent in that round. CELA outperforms BPE because CELA does not only rely on the bargaining position to adjusts the agent's concessions rate.

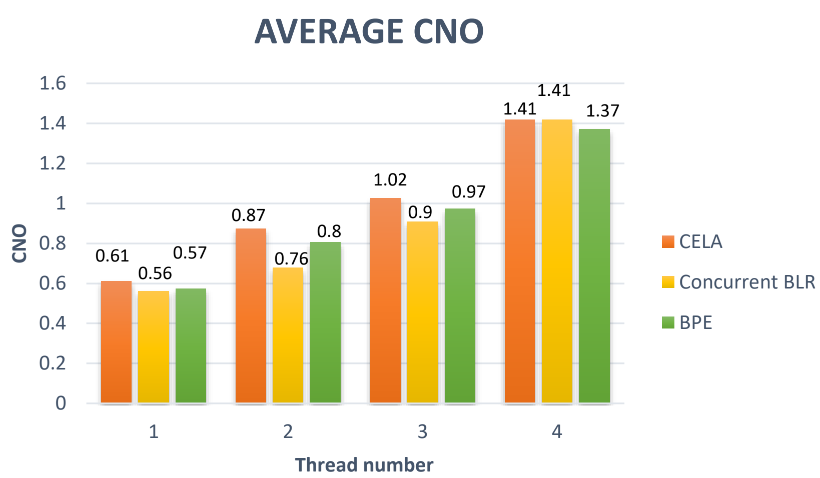


Figure 4.2: CNO of CELA, concurrent BLR and BPE

---

[1] $ANS = \frac{1}{N_{success}} \sum_{i=1}^{N_{succes}} t^i$
[2] $SR = \frac{N_{success}}{N}$ where N is the total number of negotiations.

- **Experiment 3:**
  In this experiment, we will use average utility and average negotiation speed ANS as a performance measure, in order to evaluate our negotiation model. Figure 4.3 represents the comparison between CELA, the concurrent BLR and BPE scenarios in term of average utility.

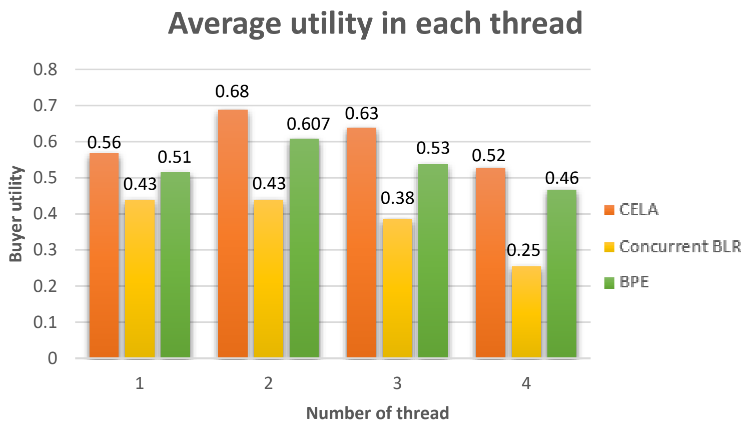


Figure 4.3: Buyer average utility in each thread

CELA outperforms BPE because CELA uses hybrid approach not only adjusts its concession tactic based on the learned deadline and reserve points also it uses the bargaining position to adjust the learned concession rate α based on the position of the agent in the negotiation.
CELA outperforms concurrent BLR because the research mechanism used by CELA explores the outcome space more effectively than the concurrent BLR. Figure 4.4 represents the comparison between CELA, the concurrent BLR and BPE scenarios in term of average negotiation speed. The average negotiation speed (ANS) measures the average negotiation duration.

Figure 4.5 represents the comparison between CELA, the concurrent BLR and BPE scenarios in term of joint utility. The joint utility is the average utility achieved by the master agent, after all, sub-agents finishes negotiating with their opponents. CELA achieves better joint utility than the concurrent BLR and BPE. There are several possible explanations for this result. First of all, our model uses hybrid approach that combines between time dependency tactic and behavior dependency tactic (Bargaining Position Estimation BPE).

Also, our approach uses learning mechanism that models the negotiation problem on a non-linear equation that allows as to adjust the concession rate for each issue independently. Finally, it uses recent optimization algorithm that proves his effectiveness in solving the non-linear equation system.

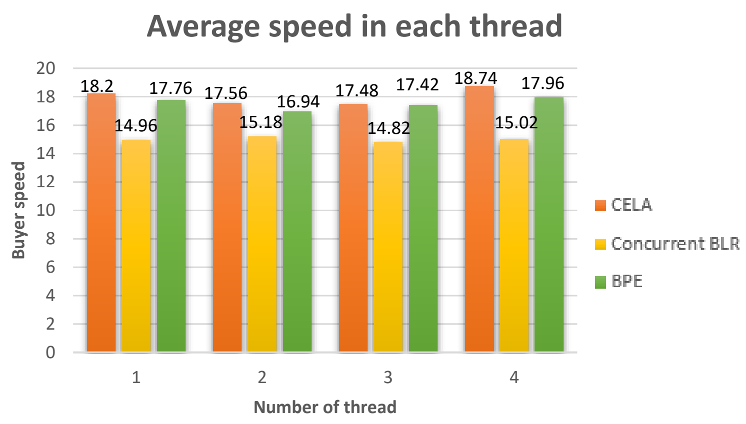


Figure 4.4: Buyer speed in each thread

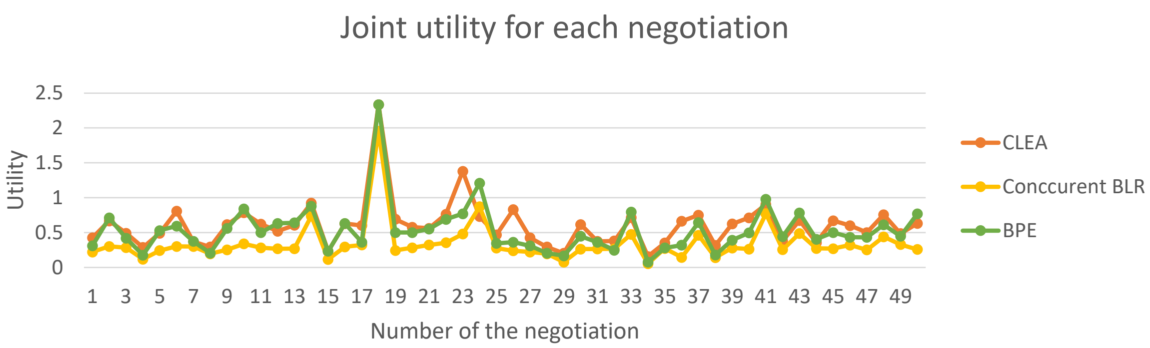
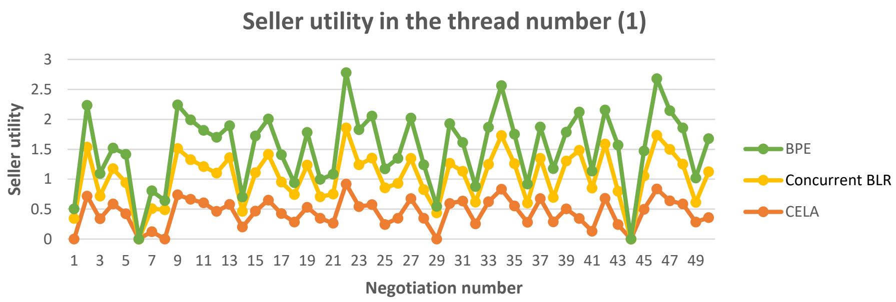


Figure 4.5: Joint utility

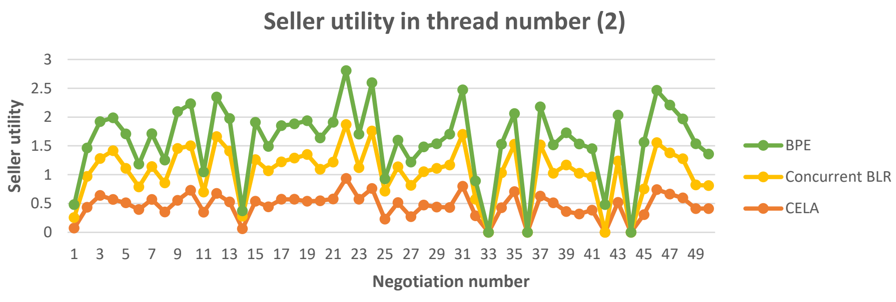
- **Experiment 4:**
  In this experiment, we will use seller utility (Opponent utility) in each thread

as a performance measure, in order to evaluate our negotiation model. Figure 4.6 highlights the comparison between CELA, the concurrent BLR and BPE scenarios in term of seller utility. It is clear that our method improves potential outcome of the negotiation, we can see that in the figure there are a remarkable difference in achieved seller utility between CELA and BPE and concurrent BLR(the opponent agent achieves better utility when it negotiates with concurrent BLR or BPE).

Those result achieved because our model doesn't change the concession strategy only base on the estimated bargaining position it uses a hybrid model that relied on time dependency tactic and bargaining position estimation. Concurrent BLR achieves better utility the BPE but CELA achieved better than it because CELA adjusts the concession strategy of each issue based on the learned reserve point by DEIWO. Unlike to BLR that adjusts in concession strategy base learned reservation utility learned by combining the Bayesian learning and regression analyses.



(a)



(b)

Figure 4.6: Seller utility in thread 1, 2, 3 and 4

4.3 Integration of CELA IN Genius

Genius (Lin et al., 2014) is a negotiation environment that implements an open architecture for heterogeneous negotiating agents, it can be used to implement, or simulate, real life negotiations. Genius offers some predefined implementation of a set of existing negotiation protocols namely, Stacked Alternating Oers Protocol, Alternating Multiple Oers Protocol, Alternating Majority Consensus Protocol, Simple Mediator Based Protocol, Mediator Feedback Based Protocol and Beyond the Protocol. Moreover, genius had Basic GUI Components that allows as to visualize the result of the experiment and also, give as possibility of Creating a domain and a Preference prole and we can run the simulation using GUI Component.

In this section, we will illustrate how we integrate in Genius simple agent the learning mechanism used by CELA. Genius has a set of predefined methods namely, getUtility, init, ReceiveMessege, choseAction and getName (see Figure 4.7).

`double getUtility(Bid bid)`	
A convenience method to get the utility of a bid taking the discount factor into account.	
`void init()`	
Informs the agent about beginning of a new negotiation session.	
`void ReceiveMessage(Action opponentAction)`	
Informs the agent which action the opponent did.	
`Action chooseAction()`	
This function should return the action your agent wants to make next.	
`String getName()`	
Returns the name of the agent. Please override this to give a proper name to your agent.	

Figure 4.7: The most important methods of the genius Agent class(Lin et al., 2014).

The method chooseAction chooses whether to accept or not the opponent proposal. We integrate our learning mechanism in this method if agent refuses the opponent offer and the current round ($t \leq \tau$) then the agent generates a counter proposal using our learning mechanism see the source code in (Figure 4.8). first

```
private Bid getBid() throws Exception {
        List<Issue> issues = utilitySpace.getDomain().getIssues();
        Random randomnr = new Random();
        DEIWO deiwo = new DEIWO(bids,bids.get(0), utilitySpace.getDomain().getIssues().size(), (int) this.round);
        Weed w = deiwo.learnDEIWO(issues);
        HashMap<Integer, Value> values = new HashMap<Integer, Value>();
        Bid bid = null;
        do {
            for (Issue i : issues) {
                switch (i.getType()) {
                case REAL:
                    IssueReal lIssueReal =  (IssueReal)i;
                    double v=0.0,a=0.0,b=0.0,c=0.0,d=0.0,x=0.0,p=0,lp=0;
                    Bid bi=this.bids.get(this.bids.size());
                    Bid ip=this.bids.get(0);
                    p=((ValueReal)ip.getValue(i.getNumber())).getValue();
                    lp= ((ValueReal)bi.getValue(i.getNumber())).getValue();
                    a=((ValueReal)bi.getValue(i.getNumber())).getValue()-((ValueReal)w.Variables.get(i.getNumber()))
                    b=((ValueReal)bi.getValue(i.getNumber())).getValue()-utilitySpace.getReservationValue();
                    if(w.deadline< (new Deadline()).getValue()){
                        v=Math.max(0, (a/b));
                        double secondTerm = (double) (w.deadline - this.round + 1) /
                                ((new Deadline()).getValue() - this.round + 1);
                        x= (Math.log(v))/(Math.log(secondTerm));
                    }else{
```

Figure 4.8: Source code of the learning mechanism.

of all the agent learns the reservation points and the deadline of its opponent by calling the method learnDEIWO, this method returns the best weed of the population in term of fitness then the agent computes for each issue the concession rate. After that, we will measure the next value of each issue in the new offer using equation presented in (section 1.3.4).

4.4 Conclusion

In this chapter, we evaluate CELA with concurrent BLR and BPE. Our experimental results confirm that CELA achieves better results than bargaining position estimation and concurrent BLR. The comparison was mad in terms of agent average utility, agent joint utility and combined negotiation outcome. We also, show that our approach ensures that opponent agent achieves less utility then the other approaches.

Conclusion

In this study, we develop a new intelligent agent called CELA that implements new negotiation mechanism for concurrent negotiation by combining time dependency strategy and behavior time strategy. The presented model is able to engage multiple agents in simultaneous negotiation and carefully coordinates the subagent in each independent negotiation thread.
CELA uses a hybrid approach inspired from two different mechanisms. The First one called evolution learning agent ELA able to adjust its concession strategy based on learning the private information of the opponent. The second one called bargaining position estimations BPE that estimates the bargaining position of the agent and adjusts its concession strategy in terms of the experimental study revealed that our model CELA enhances it negotiation outcome. In terms of combined negotiation outcome CNO, average utility AU and joint utility JU.

The present finding might help to solve problem resource allocation and the cloud provider problem and also it can be used in other negotiation problems that incorporates concurrent negotiations for multiple interrelated e-Markets. Our model can be easily applied to many domains like resource allocation, grid computing and cloud computing.

As a future work, we plan to extend our work to the case of double sided multilateral negotiation, where more than two agents negotiated independently to solve conflicts of interests and to find mutual agreement. Also, we will enhance the acceptance strategy of our agent by modeling the agent's preference profile. Additionally, we will try to implement our approach in genius (Lin et al., 2014) and participate in the International Automated Negotiating Agents Competition.

Appendix

Appendix A

Appendix 1

A.1 Bayesian learning agent (BLR)

A.1.1 Introduction

Bayesian learning agent (BLR) proposed by (Zhang et al., 2014). It uses the combination of the Bayesian Learning (BL) and regression analysis. To learn the reservation utility of the opponent agent. The learning process of (BLR) consists of two parts: bayesian-based prediction of negotiation deadline and reservation utility and regression analysis.

A.1.2 Prediction of Negotiation Deadline and Reservation Utility

BLR divides the outcome space into n equal cells called the prediction cell or the prediction region. Each prediction cell is bonded by maximum reservation utility, minimum reservation utility, minimum time and maximum time.
The prediction of negotiation deadline and the opponent reservation utility. Process it as follows, the learning mechanism assigns a probability X_i to each prediction cell i in the outcome space (X_i represents the probability that the reservation utility in the cell i). The cell with the highers probability is more likely to be the partner real reservation utility. After that, The regression analysis is used to up-

date the choose it hypothesis about opponent agent deadline and reservation utility according to the historical offers made the opponent see Figure. A.1.

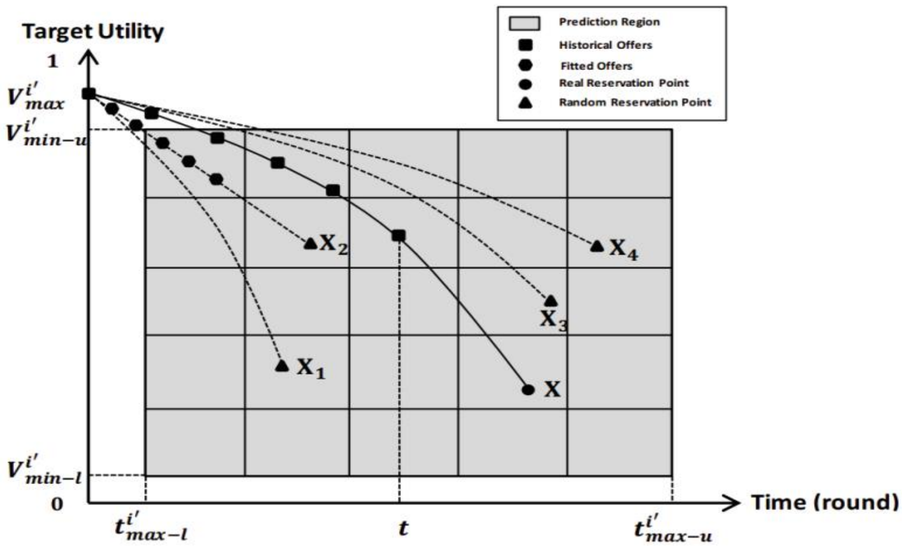


Figure A.1: Prediction Process Demonstration (Zhang et al., 2014)

where $V^{i'}_{min}$ is the opponent's min reservation utility, $V^{i'}_{max}$ is the opponent's max reservation utility, $t^{i'}_{min}$ is the opponent's min deadline and $t^{i'}_{max}$ is the opponent's max deadline.

A.1.3 Concession rate adjustment

Bayesian learning agent (BLR) updates its concession rate according to the predicted deadline and reservation utility. The optimal strategy can be find it by considering two cases see Figure. A.2
Figure. A.2 represents the two case that the agent can faces during the learning process. t^{i}_{max} is the agent deadline and t^{k}_{max} is the opponent predicted deadline. V^{i}_{max} is the agent maximum reservation utility and V^{k}_{max} is the opponent predicted reservation utility.

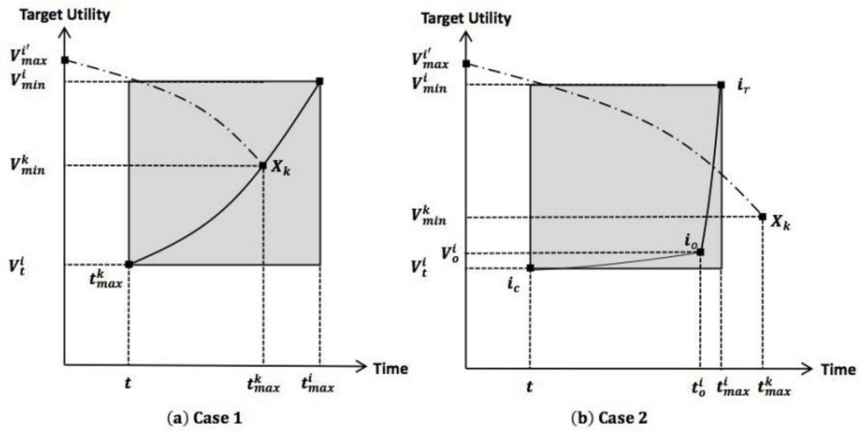


Figure A.2: Prediction Process Demonstration (Zhang et al., 2014)

- (case 1: $t^k_{max} < t^i_{max}$ and $V^k_{max} \leqslant V^i_{max}$) BLR adjust it concession rate α as follows.

$$\alpha = \frac{{t^k_{max}}^{\frac{V^k_{min}-V^i_{max}}{V^i_{min}-V^i_{max}}}}{t^i_{max}} \tag{A.1}$$

- (case 2:$t^k_{max} \geqslant t^i_{max}$ and $V^k_{max} \leqslant V^i_{max}$) BLR adjust it concession rate α as follows.

$$\alpha = \lim_{(t^i_0,v^i_0)\to(t^i_{max},v^i_t)} \frac{t^i_0}{t^i_{max}}^{\frac{V^i_0-V^i_{max}}{V_{min}}} \tag{A.2}$$

References

Aydogan, R., Festen, D., Hindriks, K. V., & Jonker, C. M. (2016). Alternating offers protocols for multilateral negotiation. *Modern Approaches to Agent-based Complex Automated Negotiation. Springer*.

Baarslag, T., Hendrikx, M. J., Hindriks, K. V., & Jonker, C. M. (2015). Learning about the opponent in automated bilateral negotiation: a comprehensive survey of opponent modeling techniques. *Autonomous Agents and Multi-Agent Systems*, 1–50.

Baarslag, T., Hendrikx, M. J., Hindriks, K. V., & Jonker, C. M. (2016). A survey of opponent modeling techniques in automated negotiation. In *Proceedings of the 2016 international conference on autonomous agents & multiagent systems* (pp. 575–576).

Baarslag, T., Pasman, W., Hindriks, K., Tykhonov, D., Visser, W., Hendrikx, M., & Feirstein, D. (2016). Negotiation user guide.

Buttner, R. (2006). A classication structure for automated negotiations.

Chopra, A. K., & Singh, M. P. (2011). Multiagent systems.

Dash, R. K., Jennings, N. R., & Parkes, D. C. (2003). Computational-mechanism design: A call to arms. *IEEE intelligent systems*, *18*(6), 40–47.

Davis, R., & Smith, R. G. (1983). Negotiation as a metaphor for distributed problem solving.

Deochake, S., Sarode, S., Kanth, S., Potdar, V., Chakraborty, S., & Mukhopadhyay, D. (2012). Henri: high efficiency negotiation-based robust interface for multi-party multi-issue negotiation over the internet. In *Proceedings of the cube international information technology conference* (pp. 647–652).

Hou, C. (2004). Predicting agents tactics in automated negotiation. In *Intelligent agent technology, 2004.(iat 2004). proceedings. ieee/wic/acm international* *conference on* (pp. 127–133).

Jennings, N. R., Faratin, P., Lomuscio, A. R., Parsons, S., Wooldridge, M. J., & Sierra, C. (2001). Automated negotiation: prospects, methods and challenges. *Group Decision and Negotiation*, *10*(2), 199–215.

Jennings, N. R., Sycara, K., & Wooldridge, M. (1998). A roadmap of agentresearch and development.

Lin, R., Kraus, S., Baarslag, T., Tykhonov, D., Hindriks, K., & Jonker, C. M. (2014). Genius: An integrated environment for supporting the design of generic automated negotiators. *Computational Intelligence*, *30*(1), 48–70.

Lomuscio, A. R., Wooldridge, M., & Jennings, N. R. (2001). A classication scheme for negotiation in electronic commerce.

Marsa-Maestre, I., Lopez-Carmona, M. A., Velasco, J. R., & de la Hoz, E. (2009). Effective bidding and deal identication for negotiations in highly nonlinear scenarios.

Mehrabian, A. R., & Lucas, C. (2006). A novel numerical optimization algorithm inspired from weed colonization. *Ecological informatics*, *1*(4), 355–366.

Nguyen, T. D., & Jennings, N. R. (2004). Coordinating multiple concurrent negotiations.

OxfordUniversityPress. (2017). *oxforddictionaries.* `https://en.oxforddictionaries.com/definition/process`. (Online; accessed 25 May 2017)

Radu, S., Kalisz, E., & Florea, A. M. (2013). A model of automated negotiation based on agents profiles. *Scalable Computing: Practice and Experience*, *14*(1), 47–56.

Rahwan, I., Ramchurn, S. D., Jennings, N. R., Mcburney, P., Parsons, S., & Sonenberg, L. (2003). Argumentation-based negotiation. *The Knowledge Engineering Review*, *18*(4), 343–375.

Rosenschein, J. S., & Zlotkin, G. (1994). Rules of encounter: designing conventions for automated negotiation among computers.

Rubinstein, A. (1985). A bargaining model with incomplete information about time preferences. *Econometrica: Journal of the Econometric Society*, 1151–1172.

Shojaiemehr, B., & Rafsanjani, M. K. (2014). A fuzzy system approach to multilateral automated negotiation in b2c e-commerce. *Neural Computing and Applications*, *25*(2), 367–377.

Sim, K. M. (2013). Complex and concurrent negotiations for multiple interrelated e-markets. *IEEE transactions on cybernetics*, *43*(1), 230–245.

Sim, K. M., Guo, Y., & Shi, B. (2007). Adaptive bargaining agents that negotiate optimally and rapidly. In *Evolutionary computation, 2007. cec 2007. ieee congress on* (pp. 1007–1014).

Sim, K. M., & Shi, B. (2010). Concurrent negotiation and coordination for grid resource coallocation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *40*(3), 753–766.

STONE, P., & VELOSO, M. (2000). Multiagent systems: A survey from a machine learning perspective.

Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, *11*(4), 341–359.

Tsuruhashi, Y., & Fukuta, N. (2015). Toward a framework for implementing and analyzing meta-strategies on simultaneous negotiations. , 3–18.

Vij, S., Patrikar, M., Mukhopadhyay, D., & Agrawal, A. (2015). A smart and automated negotiation system based on linear programming in a multilateral environment. *Smart CR*, *5*(6), 540–552.

Wong, T., & Fang, F. (2010). A multi-agent protocol for multilateral negotiations in supply chain management. *International Journal of Production Research*, *48*(1), 271–299.

Wooldridge, M. (1997). Agent-based software engineering. *IEE Proceedings-software*, *144*(1), 26–37.

Wurman, P. R. ., Wellman, M. P., & Walsh, W. E. (1998). The michigan internet auctionbot:a configurable auction server for human and software agents. *In Proceedings of the second internatonal Confernece on Autonomous Agents(Agents-98).*

Yu, C., Ren, F., & Zhang, M. (2013). An adaptive bilateral negotiation model based on bayesian learning. In *Complex automated negotiations: Theories, models, and software competitions* (pp. 75–93). Springer.

Yu, C., Wong, T., & Li, Z. (2017). A hybrid multi-agent negotiation protocol supporting supplier selection for multiple products with synergy effect. *International Journal of Production Research*, *55*(1), 18–37.

Zeng, D., & Sycara, K. (1998). Bayesian learning in negotiation. *International Journal of Human-Computer Studies*, *48*(1), 125–141.

Zhang, J., Ren, F., & Zhang, M. (2014). An innovative approach for predicting both negotiation deadline and utility in multi-issue negotiation. In *Pricai 2014: Trends in artificial intelligence* (pp. 1076–1088). Springer.

Zheng, R., Dai, T., Sycara, K., & Chakraborty, N. (2016). Automated multilateral negotiation on multiple issues with private information. *INFORMS Journal on Computing*, *28*(4), 612–628.

Zhou, Y., Luo, Q., & Chen, H. (2013). A novel differential evolution invasive weed optimization algorithm for solving nonlinear equations systems. *Journal of Applied Mathematics*, *2013*.

Résumé:

La négociation automatique est un processus dans lequel les agents interagissent pour résoudre les conflits entre eux. Les agents recherchent conjointement des accords acceptables . Dans ce travail, nous nous concentrerons sur la négociation automatique simultanée, où un agent logiciel négocie avec plusieurs agents simultanément. Dans un tel environnement, un agent peut effectuer simultanément plusieurs négociations bilatérales à l'aide du système multi-thread. Ce travail présente un nouvel agent de négociation pour coordonner la négociation automatique simultanée appelée CELA. En fait, CELA est une approche hybride pour la négociation automatique qui utilise la combinaison de la stratégie de négociation temporelle et la bargaining position.

Mots-clés:

Négociation automatique, Négociation automatique simultanée, Apprentissage des délais et des points de réservation, position de négociation, systèmes basés sur les agents.

Abstract:

Automated negotiation is a process in which self-interested agents interact to resolve conflicts among themselves. Agents search jointly for mutually acceptable agreements in a multidimensional space formed by negotiable issues. In this work, we will focus on concurrent automated negotiation, where a software agent negotiates with multiple agents simultaneously. In a such environment, an agent can perform simultaneous multiple bilateral negotiations using the multi-thread system. To this end, this work introduces a novel negotiation agent for coordinating concurrent automated negotiation called CELA. In fact, CELA is a hybrid approach for automated negotiation that uses combination of time dependency strategy and the bargaining position.

Keywords: Automated negotiation, Concurrent automated negotiation, deadline and reservation points learning, bargaining position, agent-based systems.