An Evolutionary Approach for Learning Opponent's Deadline and Reserve Points in Multi-Issue Negotiation

R. Ayachi*, H. Bouhani, N. Ben Amor

Larodec laboratory, Tunis (Tunisia)

Abstract

The efficiency of automated multi-issue negotiation depends on the available information about the opponent. In a competitive negotiation environment, agents do not reveal their parameters to their opponents in order to avoid exploitation. Several researchers have argued that an agent's optimal strategy can be determined using the opponent's deadline and reserve points. In this paper, we propose a new learning agent, so-called Evolutionary Learning Agent (ELA), able to estimate its opponent's deadline and reserve points in bilateral multi-issue negotiation based on opponent's counter-offers (without any additional extra information). ELA reduces the learning problem to a system of non-linear equations and uses an evolutionary algorithm based on the elitism aspect to solve it. Experimental study shows that our learning agent outperforms others agents by improving its outcome in term of average and joint utility.

Keywords

I. Introduction

AUTOMATED negotiation aims to imitate the humans' negotiation process using intelligent agents. It is based on three components [1]: (1) the negotiation protocol defining the rules governing the interaction between the negotiating agents such as the number of agents, their actions, etc. When we deal with two agents, we talk about bilateral negotiation. If the negotiation concerns more than two partners, negotiation is then multilateral. (2) The negotiation object corresponding to the set of issues under negotiation. Issues are the characteristics of the negotiation item that are taken into account during the evaluation [7]. A negotiation can be either a single-issue or a multi-issue. (3) The negotiation strategy determining the agents plan for reaching a satisfactory agreement. It includes tactics and decision functions adopted by the negotiating agent.

In this work, we focus on bilateral multi-issue negotiation under a time constraint. In this context, several challenges can be derived out from the fact that negotiators do not reveal their private information (e.g., preference, deadline and reserve point) to their opponents for fear of being exploited.

Several researchers paid attention to the endowment of learning techniques into negotiating agents [1-3]. Most of the proposed learning methods require prior knowledge about the opponent. The challenge is to propose a learning method that only uses available information during the negotiation.

In this paper, we propose a learning agent, so-called Evolutionary Learning Agent (ELA), employing the evolutionary learning approach Differential Evolution Invasive Weed Optimization (DEIWO) [4] to learn its opponent's deadline and reserve points from only his counter-offers in a bilateral multi-issue negotiation. The use of DEIWO allows ELA enhancing its performance even with an important number of issues.

The remainder of this paper is as follows: Section 2 presents basic concepts of bilateral multi-issue negotiation and related work. The new learning approach is detailed in Section 3. The empirical evaluation and the analysis are presented in Section 4.

II. Bilateral Multi-issue Negotiation

As stated above, we are interested in a bilateral multi-issue negotiation framework as it is widely used in the agent's field [5]. In such environment, agents negotiate by exchanging offers and counteroffers, until they either reach a consensus that satisfies each party's private preferences, or withdraw from negotiation without any agreement. The communication between agents is made using the simple alternative offer protocol [6]. Formally, let I be a pair of negotiating agents (generally $I = \{B,S\}$ where B corresponds to a Buyer and S to a Seller). According to the negotiation process, the two negotiating agents act in conflictive roles. An agent receiving an offer at time t needs to decide whether (1) to accept or (2) to reject it and propose a counteroffer at time $t + 1$.

Counter-offers can be made until one agent's deadline (denoted τ^i for agent i) is reached as shown in Fig. 1. Since the negotiating agents have time constraint, they use time-dependent tactics (when preparing their offers) which model the fact that they concede faster as the deadline approaches [6].

Let $J = \{1,..,n\}$ be the set of issues under negotiation. For each issue $j \in J$, any agent $i \in I$ has a lower and upper reserve point, denoted by IP_j^i and RP_j^i, respectively and corresponding to the minimal and maximal utility the agent i is willing to accept during the negotiation.

* Corresponding author.

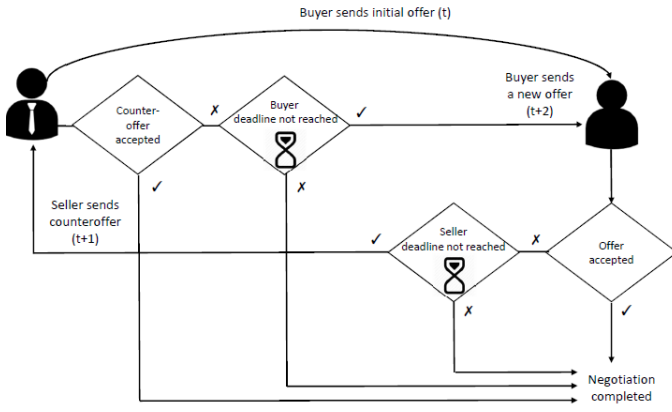E-mail address: raouia.ayachi@gmail.com

Fig. 1. Negotiation process.

A. Basics of Bilateral Multi-issue Negotiation

An agent i starts the negotiation (at $t = 0$) with its initial price (IP_j^i) and at $t = \tau^i$, concedes to its reserve point (RP_j^i). Formally, in each round t, agent i assigns to each issue $j \in J$ a value $x_j^i[t] \in [IP_j^i, RP_j^i]$ expressed by:

$$x_j^i[t] = IP_j^i + \left(RP_j^i - IP_j^i\right) * \left(\frac{t}{\tau^i}\right)^{\alpha_j^i} \tag{1}$$

where α_j^i is i's concession rate that quantifies the amount an agent concedes towards its opponent during the negotiation [7]. For simplicity reasons, we will use the notation α_j to denote the concession rate for either agent i or its opponent i'. Fig. 2 depicts the behavior of α_j on agent's offers curve. Clearly, the agent's tactics are classified into three classes [8], depending on the value of α_j , namely:

- *Boulware* ($\alpha_j < 1$): where an agent maintains the offered value until the time is almost exhausted.
- *Conceder* ($\alpha_j > 1$): where an agent concedes to its reserve point very quickly.
- *Linear* ($\alpha_j = 1$): where an agent makes a constant rate of concession.
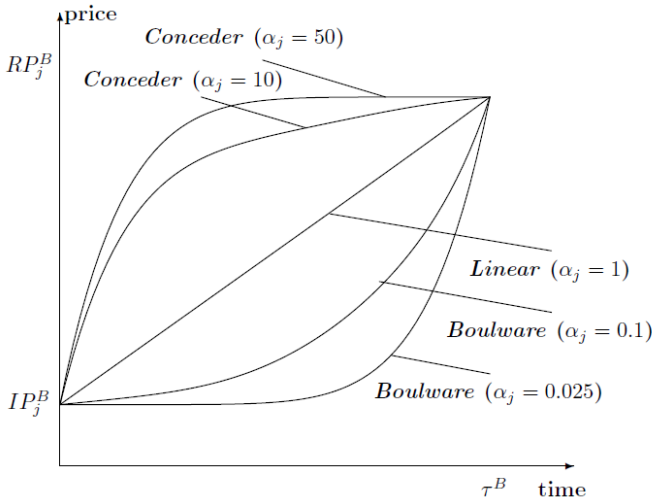

Fig. 2. The concession rate's impact on agent's offering curve.

The opponent's concession rate $\alpha_j^{i'}$ can be computed using $IP_j^{i'}$ and two successive offers as follows:

$$\alpha_j^{i'} = \log_{\frac{t}{t-1}}\left(\frac{x_j^i[t] - IP_j^{i'}}{x_j^i[t-1] - IP_j^{i'}}\right) \text{ where } t \geq 3 \tag{2}$$

A non-learning agent keeps the value of α_j unchanged until the negotiation ends while a learning one adjusts α_j in order to maximize its utility. Each agent employs an evaluation function $V_j^i : [IP_j^i, RP_j^i] \to [0,1]$ which assigns a normalized valuation to each possible value x_j. Formally:

$$V_j^i(x_j) = \frac{RP_j^i - x_j}{RP_j^i - IP_j^i} \tag{3}$$

Thus, the utility function modelling agent i's preferences is a linearly additive function defined by:

$$U^i(x) = \sum_{j=1}^{n} w_j^i \, V_j^i(x_j) \tag{4}$$

where w_j^i is the j's weight of the issue given by agent i. The utility function $U^i(x)$ is set to 1 at the beginning of the negotiation and decreases as the deadline approaches.

Since the negotiation is a sequence of alternative offers finishing with an *accept* or a *withdraw*. The response to an offer $x^{i'}[t]$ at time round t, denoted by Response ($t, x^{i'}$), is :

$$\text{Response}(t, x^{i'}) = \begin{cases} withdraw \; if \; t > \tau^i \\ accept \; if \; U^i(x^{i'}[t]) \geq U^i(x^i[t]) \\ x^i[t] \qquad otherwise \end{cases} \tag{5}$$

B. Related Work on Learning Opponent's Deadline and Reserve Point

In recent years, endowing agents with machine learning techniques has attracted automated negotiation community. Several researchers paid their attention to learn opponent's deadline and reserve point [1, 2, 9–12]. In what follows, we briefly review works related to the learning of opponent's deadline and reserve point. In fact, the learning problem has been deeply studied for bilateral single-issue negotiation. The first investigation was established by Hou [9] to learn the opponent's deadline and reserve point by employing non-linear regression. No mechanism has been used for adapting the concession strategy and this represents the major weakness of the method. Sim et al. [10] proposed a Bayesian Learning (BL) approach called BLGAN that only learns the opponent's reserve point and then employs a genetic algorithm to generate counter-offers. Sim et al. [11] proposed an improved version of BLGAN in which they incorporated a deadline learning method. Compared to BLGAN, Gwak et al. [2] exploited a new conditional probability to update the belief of the opponent's reserve point. In this framework, the concession rate adjustment mechanism is not efficient when opponent's deadline is greater than the learning agent's deadline. Yu et al. [12] proposed a combination of BL and regression analysis in order to estimate the opponent's deadline and reserve point. They defined a set of hypotheses about the values of the opponent's deadline and reserve point and then used the non-linear correlation coefficient to update agent's beliefs.

The research mentioned above only deals with bilateral single-issue negotiation. In contrast, only two works handle multi-issue negotiation. In fact, Zeng el al. [13] used a BL method called *Bazaar* which is a sequential decision making approach modelling beliefs of the opponent's reserve point. Their learning method does not include any mechanism to learn the opponent's deadline, in addition, it requires extra information about the opponent which is not available in most cases. Zhang et al. [14] improved the learning approach proposed in [12] to deal with multi-issue negotiation. To this end, they use strong assumptions about the opponent's preferences since they only deal with conflicting and equally weighted issues. By conflict issues they mean that increasing the value of an issue will help agents to raise their utilities but to decrease the opponents' utilities.

From another side, Coehoorn et al. [15] have used the Kernel Density Estimation (KDE), a non-parametric method for learning opponent's preferences where the initial distribution is based solely on the training data. However, this method was applied in a context of making negotiation trade-offs in bilateral encounters in which an agent concedes on one issue and demands more on another, which falls outside the scope of this paper.

From this review on bilateral single and multi-issue negotiation, it is clear that existing works strongly depend on a set of hypothesis. These assumptions must be made about the underlying distribution function. In what follows, we propose a novel approach to learn opponent's deadline and reserve point for multi-issue negotiation able to learn simultaneously multiple reserve points for multiple issues without any extra information.

III. A NOVEL APPROACH FOR LEARNING OPPONENT'S DEADLINE AND RESERVE POINTS IN MULTI-ISSUE NEGOTIATION

The basic idea of the new negotiation approach is to model the learning process as an optimization problem. To this end, the learning problem should be expressed in mathematical terms, i.e., reduced to a system of non-linear equations, in order to take advantage of recent results on optimization. In fact, the Differential Evolution Invasive Weed Optimization (DEIWO) [4] algorithm will be used to solve the system. Afterwards, our learning agent so called *Evolutionary Learning Agent* (ELA) should be able to adjust its concession strategy according to the estimated value of opponent's deadline and reserve points.

A. Transformation of the Learning Problem

As mentioned in Section 2, deadline and reserve point are inter-dependent terms, i.e., if the deadline is learned, the reserve point can be easily estimated. This is explained by the fact that agents concede to their reserve points when the deadline is reached. Based on (1), we can easily express the relationship between deadline and reserve point as expressed by (6) and (7).

$$RP_j^{i'} = \frac{x_j^{i'}[t] - IP_j^{i'}}{\left(\frac{t}{\tau^i}\right)^{\alpha_j^i}} + IP_j^{i'} \tag{6}$$

$$\tau^{i'} = t.\left(\frac{RP_j^{i'} - IP_j^{i'}}{x_j^{i'}[t] - IP_j^{i'}}\right)^{\frac{1}{\alpha_j^{i'}}} \tag{7}$$

The proposed approach aims to learn the opponent's deadline and reserve points by reducing the learning problem to a system of non-linear equations problem. To this end, we will use (7) in order to build our system. Moreover, the learning agent ELA needs to find the parameters that best fit its opponent's historical offers. Formally, for each issue under negotiation, the following error function should be minimized:

$$f_j\left(\widehat{RP}_j^{i'}[t], \hat{\tau}^{i'}[t]\right) = \sum_{k=1}^{t}(t/\left(\frac{x_j^{i'}[k] - IP_j^{i'}}{\widehat{RP}_j^{i'}[t] - IP_j^{i'}}\right)^{\frac{1}{\alpha_j^{i'}}} - \hat{\tau}^{i'})^2 \tag{8}$$

where $\hat{\tau}^{i'}$ and $\widehat{RP}_j^{i'}[t]$ are the learned deadline and j's reserve point at time round t, respectively.

To take advantage of the issues' multitude and (8), we reduce the learning problem to a system of nonlinear equations as follows:

$$\begin{cases} f_1\left(\widehat{RP}_1^{i'}[t], \hat{\tau}^{i'}[t]\right) = 0 \\ \quad ... \\ f_J\left(\widehat{RP}_J^{i'}[t], \hat{\tau}^{i'}[t]\right) = 0 \end{cases} \tag{9}$$

The deadline's search area is restricted as it remains the same for all cases. While when we deal with multiple reserve points (i.e., one for each issue) we have a good approximation of the opponent's deadline (i.e., close to the exact value) but inaccurate estimations of the opponent's reserve points. To make the results rigorous, the opponent's reserve points should be recomputed using the relation in (6).

Solving a system of non-linear equations corresponds to a multi-objective optimization problem in which all the functions should be minimized. Formally,

$$min \sum_{j=1}^{J} f_j\left(\widehat{RP}_j^{i'}[t], \hat{\tau}^{i'}[t]\right) \tag{10}$$

B. DEIWO Optimization Algorithm

There exists an extensive literature centered on solving optimization problems [16, 17]. Within the panoply of existing methods, we focus on a recent evolutionary algorithm *Differential Evolution Invasive Weed Optimization* (DEIWO) [4] that proved its efficiency for solving systems of non-linear equations. DEIWO has the abilities to overcome local optimal solutions and obtain global optimal solutions and is based on two popular global optimization algorithms, namely: the Invasive Weed Optimization (IWO) and the Differential Evolution (DE).

The first part of DEIWO employs the IWO algorithm which is a numerical optimization method inspired from colonizing weeds [18]. In IWO, weeds refers to feasible solutions of the given problem. They are spread over the search area and are allowed to produce seeds (i.e., new solutions) depending on their fitness. After some iterations, the number of population reaches its maximum, consequently, a mechanism for eliminating plants with poor fitness activates. Compared to the Genetic Algorithm (GA) [19], IWO employs a different way to disperse new individuals. In fact, the generated seeds are randomly dispersed over the search space by normally distributed random numbers with mean equal to zero; but varying variance [18]. The second part of DEIWO exploits DE which provides means for accelerating the optimization [20]. It is based on three operators: *mutation, crossover* and *selection*. These operators keep population diversity and avoid premature convergence.

To improve the quality of the DEIWO output at time round t, we incorporate its output at time round $t-1$. Thus, if the previous output stills optimal, it is not necessary to readjust the opponent's learned parameters. Formally, the improved DEIWO procedure is outlined in Algorithm 1.

Algorithm 1 the Differential Evolution Invasive Weed Optimization Algorithm

1: **input**: Solution at t-1 + the opponent's historical offers + DEIWO parameters;
2: **output**: Best solution at t;
3: Set the generation counter g=0;
4: Initialize the population size;
5: Initialize the population P (g); /*Initialization step*/
6: **while** g < maximum iteration **do** /*IWO phase*/
7: Compute the fitness for each weed in P (g);
8: Produce new seeds based on fitness; /*Reproduction step*/
9: Add new seeds to the population P (g); /*Spatial dispersal step*/
10: **if** population size > maximum population **then**
11: Eliminate weeds with poor fitness; /*Competitive exclusion step*/
12: **end if** /*DE phase*/
13: Perform mutation on P (g);
14: Perform crossover on P (g);
15: Perform selection on P (g);
16: Increment g by 1;
17: **end while**
18: **return** best solution;

In our problem, each weed represents a solution for the system in Eq. (9) and is expressed as follows:

$$w_n = \left(\hat{\tau}^{i'}, \widehat{RP}_1^{i'}, \widehat{RP}_2^{i'}, \ldots, \widehat{RP}_J^{i'} \right) \tag{11}$$

where $\hat{\tau}^{i'}, \widehat{RP}_1^{i'}, \widehat{RP}_2^{i'}, \ldots, \widehat{RP}_J^{i'}$ are feasible solutions for (9). Basically, weeds with the lowest fitness are closest to the optimal solution. To evaluate the quality of a solution, we consider the following fitness function:

$$f(w_n) = \sum_{j=1}^{J} f_j \left(\widehat{RP}_j^{i'}[t], \hat{\tau}^{i'}[t] \right) \tag{12}$$
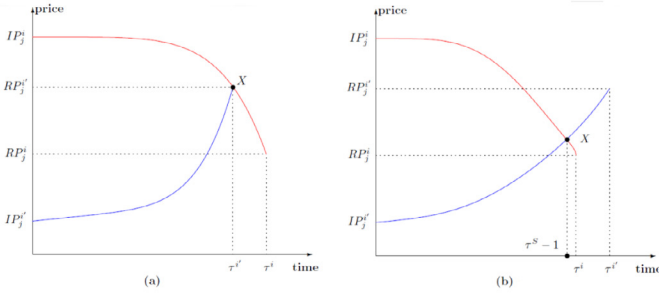

Fig. 3. Two cases of concession strategy: (a) $\hat{\tau}^{i'} < \tau^i$ and (b) $\hat{\tau}^{i'} > \tau^i$.

C. Concession Rate Adjustment

Learning the opponent's deadline and reserve points is necessary for finding ELA's optimal strategy. In each round, the learning agent uses the predicted values of deadline and reserve points in order to adjust its bidding strategy. Henceforth, it could improve its outcome and avoid disagreements at the end of the negotiation.

Before laying out with the concession rate, let us adapt the offering function. ELA generates counter offers using the following equation proposed in [11]:

$$x_j^i[t] = x_j^i[t-1] + \left(RP_j^i - x_j^i[t-1] \right) * \left(\frac{1}{\tau^i - t + 1} \right)^{\alpha_j} \tag{13}$$

Compared to (1), ELA treats its previous offer $x_j^i[t-1]$ as its new initial point at time round t.

The opponent's concession rate $\alpha_j^{i'}$ can be computed using $IP_j^{i'}$ and two successive offers as follows:

$$\alpha_j^{i'} = log_{\frac{t}{t-1}} \left(\frac{x_j^{i'}[t] - IP_j^{i'}}{x_j^{i'}[t-1] - IP_j^{i'}} \right), \text{ where } t \geq 3 \tag{14}$$

A non-learning agent keeps the value of α_j unchanged until the negotiation ends while a learning one adjusts α_j in order to maximize its utility.

Finding the optimal strategy for the learning agent can be analyzed by considering two cases:

- **Case 1** ($\hat{\tau}^{i'} < \tau^i$): In this case, the opponent's deadline is smaller than agent i's deadline. Fig.3 (a) depicts the behavior of the learning agent in this case. When the two curves intersect, the negotiating agents reach an agreement. That is why, ELA needs to negotiate with its opponent's as long as possible in order to catch the opponent's best offers. Therefore, the optimal strategy for agent i is to make its offers curve cross the opponent's offer curve at t=$\hat{\tau}^{i'}$ (point X). Hence, the optimal strategy can be computed as follows [11]:

$$\alpha_j^i[t] = \left| log_{\frac{\hat{\tau}^{i'} - t + 1}{\tau^i - t + 1}} \left(max \left\{ 0, \left| \frac{x_j^{i'}[t-1] - \widehat{RP}_j^{i'}}{x_j^{i'}[t-1] - RP_j^i} \right| \right\} \right) \right| \tag{15}$$

where $\hat{\tau}^{i'}$ and $\widehat{RP}_j^{i'}$ are, respectively, the learned opponent's deadline and issue j's reserve point.

- **Case 2** ($\hat{\tau}^{i'} > \tau^i$): In this case, the opponent's deadline is greater than agent i's deadline. Fig. 3(b) depicts the behavior of the learning agent in this case.

Proposition 1 *let X be the optimal point that maximizes the learning agent's utility. The optimal strategy for agent i in this case is to make its offers curve cross the opponent's offer curve at $t = \tau^i - 1$ since at $t = \tau_t$ agent i concedes to its reserve point which is the worst case. The proposed adjusting formula is as follows:*

$$\alpha_j^i[t] = log_{\frac{\tau^i - t}{\tau^i - t + 1}} \left(\frac{IP_j^i - x_j^i[t-1]}{RP_j^i - x_j^i[t-1]} + \frac{RP_j^{i'} - IP_j^i}{RP_j^i - x_j^i[t-1]} \left(\frac{\tau^i - 1}{\tau^{i'}} \right)^{\alpha_j^{i'}} \right) \tag{16}$$

Proof For each issue under negotiation, crossing the opponent's offers curve at $t = \tau^i - 1$ is expressed as follows:

$$x_i^{\tau^i - 1} = x_{i'}^{\tau^i - 1}$$

$$\overset{so}{\Rightarrow} IP_j^i + (RP_j^i - IP_j^i) \left(\frac{\tau^i - 1}{\tau^i} \right)^{\alpha_j^i} = IP_j^{i'} + (RP_j^{i'} - IP_j^{i'}) \left(\frac{\tau^i - 1}{\tau^{i'}} \right)^{\alpha_j^{i'}}$$

$$\overset{so}{\Rightarrow} \left(\frac{\tau^i - 1}{\tau^i} \right)^{\alpha_j^i} = \frac{IP_j^{i'} - IP_j^i}{RP_j^i - IP_j^i} + \frac{RP_j^{i'} - IP_j^i}{RP_j^i - IP_j^i} \left(\frac{\tau^i - 1}{\tau^{i'}} \right)^{\alpha_j^i}$$

$$\overset{so}{\Rightarrow} \alpha_j^i = log_{\frac{\tau^i - 1}{\tau^i}} \left(\frac{IP_j^{i'} - IP_j^i}{RP_j^i - IP_j^i} + \frac{RP_j^{i'} - IP_j^i}{RP_j^i - IP_j^i} \left(\frac{\tau^i - 1}{\tau^{i'}} \right)^{\alpha_j^{i'}} \right) \blacksquare$$

Note that when the opponent's deadline is greater than agent i's deadline (case 2), there is one special case that have to be considered.

If the estimated $\widehat{RP}_j^{i'}$ for agent i (B or S) at time round $\tau_i - 1$ is lower (respectively greater) than RP_j^i, the learning agent does not adjust its concession strategy because it is still possible that the learned reserve point may change in later time rounds.

By considering (15) and (16) as the offering tactic, the adjusting formula becomes as follows:

$$\alpha_j^i[t] = \begin{cases} \left| log_{\frac{\hat{\tau}^{i'} - t + 1}{\tau^i - t + 1}} \left(max \left\{ 0, \left| \frac{x_j^{i'}[t-1] - \widehat{RP}_j^{i'}}{x_j^{i'}[t-1] - RP_j^i} \right| \right\} \right) \right| & \hat{\tau}^{i'} < \tau^i \\ log_{\frac{\tau^i - t}{\tau^i - t + 1}} \left(\frac{IP_j^i - x_j^i[t-1]}{RP_j^i - x_j^i[t-1]} + \frac{RP_j^{i'} - IP_j^i}{RP_j^i - x_j^i[t-1]} \left(\frac{\tau^i - 1}{\hat{\tau}^{i'}} \right)^{\alpha_j^{i'}} \right) & \hat{\tau}^{i'} > \tau^i \end{cases} \tag{17}$$

Equality: when the deadline of the opponent is equal to the agent deadline we will use case 2 because if the agent reaches the deadline he will give his reservation value so we need to get a deal before deadline is met at round (deadline - 1).

D. Illustrative Example

In this section, we give an illustrative example of the proposed negotiation model. Let us consider two agents, one buyer (B) and one seller (ELA), negotiating over a service provided by ELA. The issues under negotiation are the service's price and duration. It is obvious that B (respectively, ELA) wants to reduce its costs as much as possible. Hence, B wants to quickly execute the task with the lower price. On the other hand, ELA does not want to waste its computational resources but at the same time wants to maximize the selling price. Table I summarizes agents' parameters for the two issues.

TABLE I. Agent's Parameters for the Illustrative Example

	Price (DT)			Duration (sec)			Deadline
	IP	RP	α	IP	RP	α	τ
Seller (ELA)	100	55	1	30	10	1	5
Buyer	40	70	1	5	20	1	7

Each agent starts the negotiation with its initial price. Using Eq. (13), ELA's offer at $t = 1$ is computed as follows: $x_{price}^{ELA}[1] = 100 + (55 - 100)\left(\frac{1}{5}\right)^{1} = 91$. With the same equation, ELA computes its offer for the second issue. After two rounds, ELA will be able to compute B's concession rate. The sequence of counter offers for both agents is shown in Table II.

TABLE II. Agent's Offering for the Illustrative Example: Elements in the Offer Vector Refer to the Price and Duration, Respectively

Round	0	1	2
ELA	[100, 30]	[91, 26]	[82, 26.22]
Buyer	[40, 5]	[44.29, 7.14]	[48.57, 9.28]

Using (14), ELA computes B's concession rate as follows: $\alpha_{price}^{B} = log_2\left(\frac{48.57-40}{44.29-40}\right) = 1$.

After that, ELA employs the proposed DEIWO based Learning method to predict B's deadline and reserve points. In this illustrative example, we do not aim to explain how the weeds are generated. However, we clarify how the best solution is selected from a population of weeds.

Let us consider a population containing three weeds, w_1, w_2 and w_3 shown in Table III. To determine w_1's fitness, we need first to compute $f_{price}(w_1)$ and $f_{duration}(w_1)$ using Eq. (8) and then perform their sum.

TABLE III. A Population Containing Three Weeds

	$\hat{\tau}^B$	$\widehat{RP}_{price}^{B}$	$\widehat{RP}_{duration}^{B}$	Fitness
w_1	6	66	17	0.1587
w_2	7	70	20	0.0002
w_3	8	74	22	0.0161

For example,

$$f_{price}(w_1) = f_{price}(66.6) = \left[1/\left(\left(\frac{44.29-40}{66-40}\right)^{\frac{1}{1}}\right) - 6\right]^2$$
$$+ \left[2/\left(\left(\frac{48.57-40}{66-40}\right)^{\frac{1}{1}}\right) - 6\right]^2 = 0.0046$$

.

and $f_{duration}(w_1) = 0.1547$. Now, (12) is used to compute w_1's fitness as follows: $f(w_1) = 0.0046 + 0.1541 = 0.1587$. The same calculations are made for the other weeds.

The weed $w_2 = (7, 70, 20)$ is selected as the best solution because it has the lowest fitness of the population. Therefore, B's deadline (equal to 7 from w_2) is greater than ELA's deadline (equal to 5). Hence, using (17), ELA adjusts its concession rate for the issue price as follows:

$$\alpha_{price}^{ELA}[3] = log_{\frac{5-3}{5-3+1}}\left(\frac{40-82}{55-82} + \frac{70-40}{55-82}\left(\frac{5-1}{7}\right)^1\right) = 0.204$$

With the same manipulation, S adjusts its concession strategy for the second issue.

In the next rounds, ELA repeats the same procedure. Since ELA's deadline is less than B's deadline, the agreement much be reached at $t = (\tau^{ELA} - 1)$. In a such situation, ELA's offer at $t = (\tau^{ELA} - 1)$ is $X^{ELA}[4] = [57.143, 13.57]$ and B's offer is $X^B[4] = [57.143, 13.57]$. Thus,

an agreement is reached with $U(X^{ELA}) = 0.1131$ instead of $U(X^{ELA}) = 0$.

IV. Experimental Study

To evaluate a heuristic-based negotiation model, simulations need to be performed. In this section, we evaluate our new learning agent ELA through multiple simulations and scenarios.

Experiments were implemented in Java 7 language, compiled using the Eclipse Java Mars environment and ran on windows 10-64 bits equipped with an Intel Core i7-4750QM (3 GHz) and 16 GB of RAM. We start with describing the experimental protocol then we compare agents.

A. Experimental Protocol

We will evaluate our negotiation model by comparing ELA to the following agents:

1. An agent with complete information which adapts its concession strategy based on available information.

2. A no learning agent that does not learn its opponent's parameters and its concession strategy remains fixed during the negotiation.

3. The Bayesian Learning Agent (BLA) based agent [14] that learns its opponent's reserve utility and deadline in order to adjust its concession strategy.

We propose to study four scenarios (Incomplete, Complete, ELA, BLA) as detailed in Table IV. For each of scenario, 1000 random runs were carried out to show the generality and the robustness of our negotiation model. In each run, S and B were programmed using the same parameters. Table V defines the negotiation settings. Since BLA and ELA employ different learning techniques, they have additional parameters shown in Table VI.

TABLE IV. Four Negotiation Scenarios

Scenario	Seller	Buyer	
Incomplete	Incomplete information	Incomplete	information
Complete	Complete information	Incomplete	information
ELA	S learns the opponent's deadline and RPs using ELA	Incomplete	information
BLA	S learns the opponent's deadline and RU using BLA	Incomplete	Information

TABLE V. 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	$[RP_j^B, maxj]/ [minj, RP_j^S]$
RP_j	$[minj + 5, minj + (maxj - minj)/2]/ [RP_j^S + 10, maxj - 5]$
α_j	$[0.1, 5]$
τ	$[10, 100]$

TABLE VI. BLA & ELA PARAMETERS

BLA parameters									
Parameter	V_{max}		V_{min}			Number of cells			
Value	1		0.2			100			
ELA parameters									
Parameter	P_{init}	P_{max}	σ_{init}	σ_{final}	S_{min}	S_{max}	n	F	CR	MaxIt
Value	20	30	5	1	5	15	3	1	0.5	30

In order to evaluate our negotiation model, we will use four common performance measures [1]:

- **The Average Utility (AU)** considered as the most popular performance measure to gain the utility of outcomes.

$$AU = \frac{1}{N_{success}}\sum_{i=1}^{n} U(X) \qquad (18)$$

where $N_{success}$ is the number of deals.

- **The Success Rate (SR)** represents the ability of the negotiation model to reach an agreement.

$$SR = \frac{N_{success}}{N} \qquad (19)$$

where N is the number of negotiation.

- **The Average Negotiation Speed (ANS)** measuring the average duration of negotiation. A small ANS reduces costs but also affects the quality of the outcome of an agent.

$$ANS = \frac{1}{N_{success}}\sum_{i=1}^{N_{success}} t^{i} \qquad (20)$$

where t^{i} is the time round of the agreement if an agreement is reached.

- **The Joint Utility (JU)** measuring the fairness of the outcome. In a competitive environment, JU is better when it is minimal.

$$JU = \sqrt{US * UB} \qquad (21)$$

The testbed consists of two negotiation agents with conflicting interests (i.e., one buyer and one seller). Each agent is an instance from a super class Agent. Besides their private information (RP and deadline), the agents incubate three main procedures:

- A procedure for recording the opponent's historical offers.
- A procedure for learning the opponent's parameters.
- A procedure for generating offers based on the learning results.

Fig. 4 depicts a simple negotiation simulation using the Eclipse environment. It illustrates a negotiation between one buyer and one seller under the incomplete information scenario and a deadline range of [9-10]. The negotiation ends by B accepting S's offer. The iteration number corresponds to the number of runs.

B. Results and Analysis

The goal of the proposed negotiation model is to achieve results close to the best scenario which is the complete information scenario. Empirical results were recorded from *S*'s perspective and are shown in Fig. 5 and Fig. 6. Stacked lines are used to represent results, the x-axis indicates deadline ranges and y-axis corresponds to performance measures.
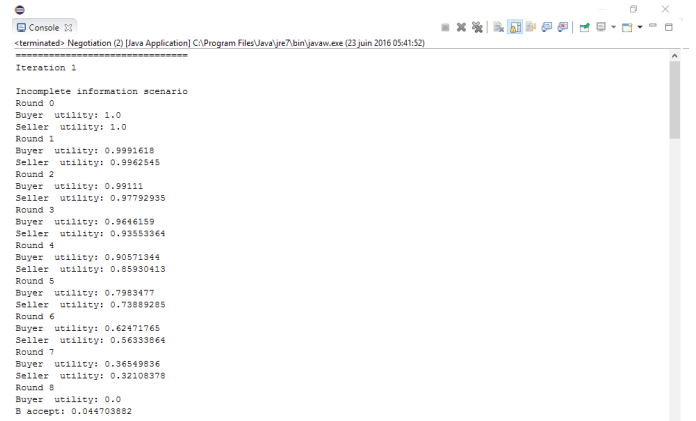

Fig. 4. Eclipse Toolbox.

Fig. 5 depicts results of the comparison between ELA and the complete and incomplete scenarios. Obviously, ELA achieves very close results to the optimal scenario. For example, we pinpoint that ELA's average utility is equal to 0.618 for the deadline range [30 − 40], which is very close to the best scenario (equal to 0.625). ELA also achieves little faster ANS than the complete information agent (Fig. 5(c)). This is due to the approximation value of opponent's deadline and reserve points. Compared to the incomplete scenario, ELA always achieves much better AU and JU (Fig. 5(a) and Fig. 5(d)). We can also observe that our learning agent always reaches an agreement (SR = 1) (Fig. 5(b)). In contrast, in the incomplete scenario, *S* does not always reach agreements, especially for short deadlines (i.e., 10 to 40 rounds).

Fig. 6 presents different results relative to the comparison between ELA and BLA agents. Clearly, ELA outperforms BLA. In fact, ELA achieves much better AU and JU (Fig. 6(a) and Fig. 6(c)) since BLA achieves faster ANS (Fig. 6(b)) affecting the quality of its outcome. For example, ELA attains 58.938 of ANS (deadline range [60 − 70]), which is higher than the one of BLA (equal to 49.788). In term of SR, the two agents always reach agreements.

To test the cut-off points of ELA, we proposed to increase the number of issues from 4 to 5, then boosted it to 10 and 20. The results of this experiment are shown in Fig. 7 and Fig. 8.

Fig. 7 represents the success rate of ELA, BLA, complete and incomplete scenarios. It is clear that ELA's success rate follows the same behavior as the number of issues increases. Indeed, ELA achieved 0.984 successful negotiations for 5 issues, then it reached the value of 0.995, which is very close to the complete scenario, when we upgraded the number of issues to 20. While BLA failed from 10 issues since the computational complexity raises by boosting the number of prediction cells used by the Bayesian learning.

The average utility of the four negotiation strategies is depicted by Fig.8. Clearly, ELA provides better average utility than BLA since the first one adjusts the concession strategy of each issue based on the learned reserve point by DEIWO. Unlike to BLA that uses a single concession rate to adjust the concession strategy for all issues based on the learned reservation utility.
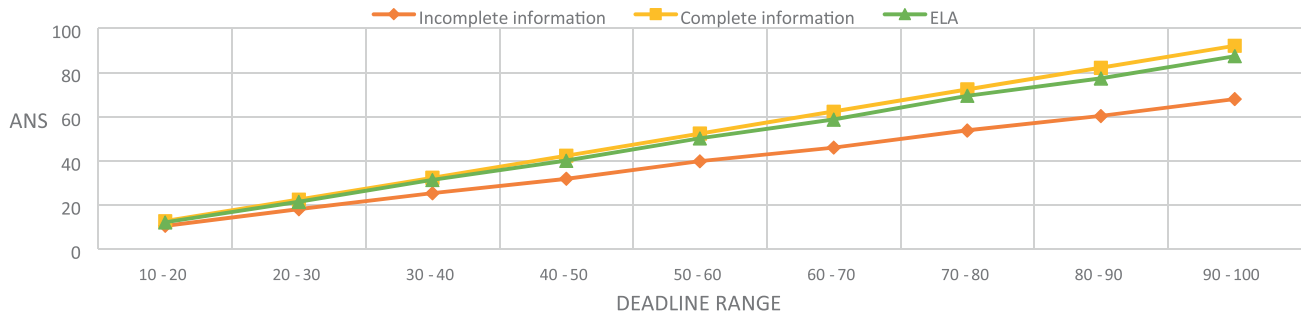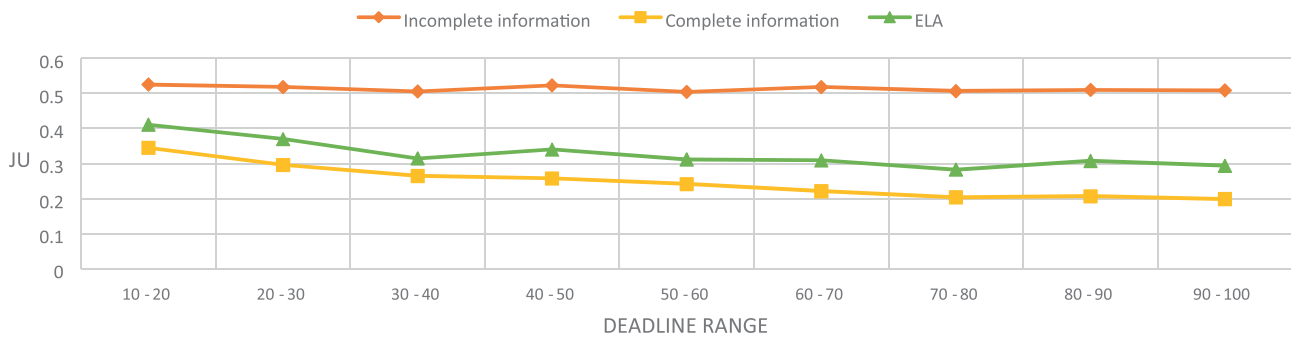
UTILITY
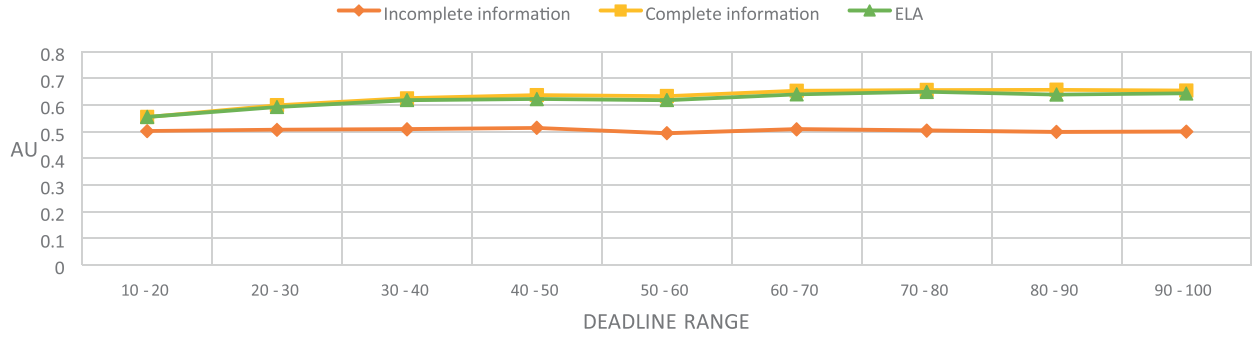
(a)

SUCCESS RATE
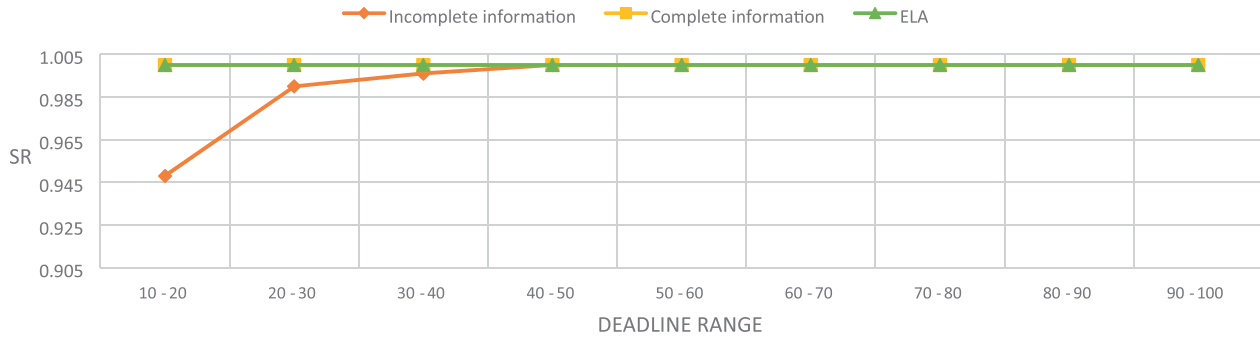
(b)

AVERAGE NEGOTIATION SPEED

(c)

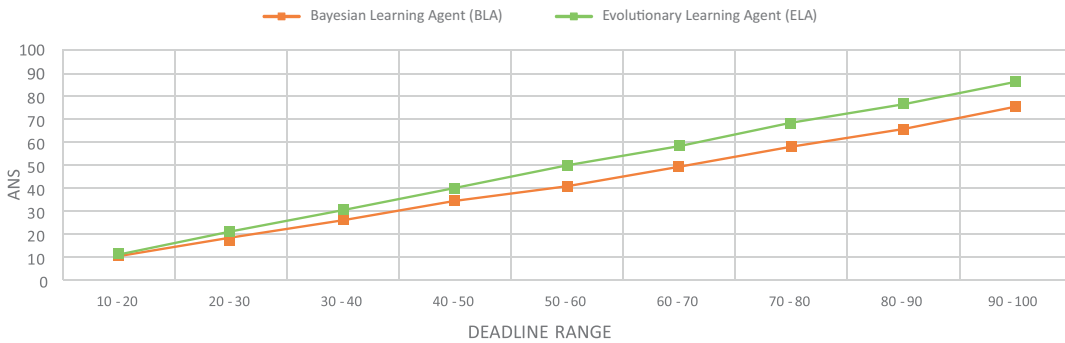## JOINT UTILITY

(d)

Fig. 5. ELA vs complete and incomplete information agents.
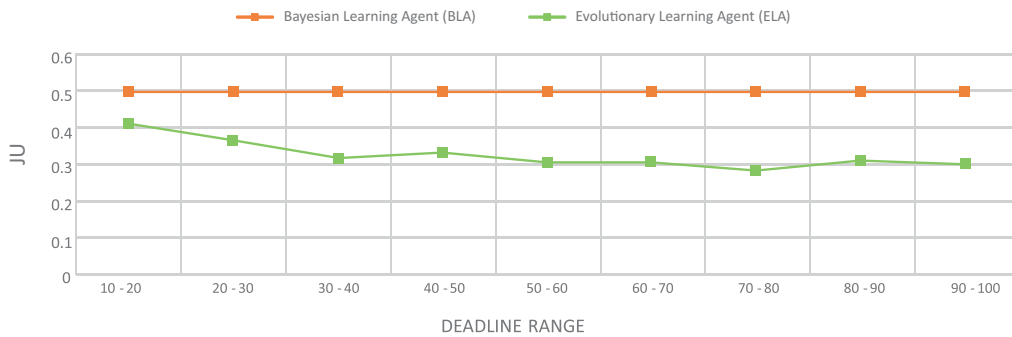
AVERAGE UTILITY

(a)

AVERAGE NEGOTIATION SPEED

(b)

JOINT UTILITY
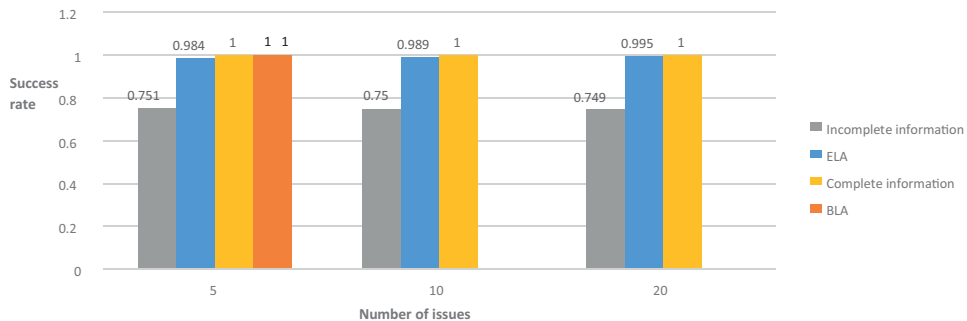

(c)

Fig. 6. ELA vs BLA.

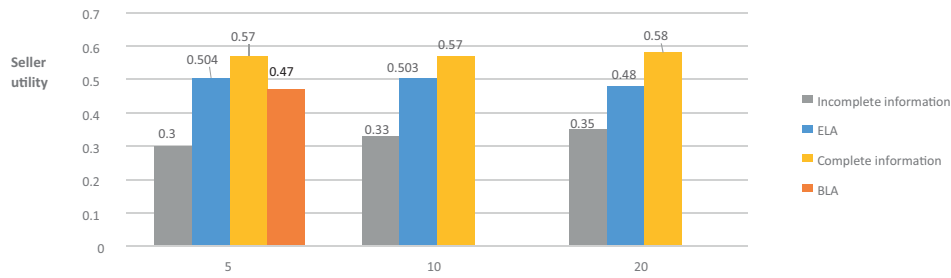Fig. 7. Success rate for 5, 10 and 20 issues.

Fig. 8. Seller average utility for 5, 10 and 20 issues.

We have also compared the execution time of BLA and ELA as shown in Table VII. In fact, the execution time of BLA is equal to 3000 seconds for the case of 7 issues, while with ELA, the execution time is only equal to 3 seconds. This confirms that DEIWO used by ELA explores the outcome space faster than the combination of regression analysis and Bayesian learning used by BLA.

TABLE VII. EXECUTION TIME OF ELA AND BLA

Number of issues	BLA	ELA
1	0.093 ms	0.141 ms
2	0.150 ms	0.125 ms
3	0.39 ms	0.903 ms
4	0.851 ms	1 s : 32 ms
5	0.469 ms	0.185 ms
6	145 s	2442 ms
7	3000 s	3 s

V. CONCLUSION AND FUTURE WORK

In this paper, we introduced the ELA agent which learns its opponent's deadline and reserve points in a bilateral multi-issue negotiation. ELA employs an evolutionary optimization algorithm in order to learn its opponent's parameters. A new concession strategy adjustment is performed to improve an agent's outcome. Empirical results showed that ELA gets very close outcomes to the best scenario. Also, we test the limit of our model in order to find out how much it can handle in term of number of issues. Our future work consists in studying in depth multi-issue negotiation and proposing a new negotiation model based on multi-criteria methods such as ANP [21] for inter-dependent issues and extend our negotiation model to multilateral negotiation. Another perspective is to extend our approach to the case of concurrent one side multilateral negotiation in which an agent may engage simultaneously many agents in multiple bilateral negotiation.

REFERENCES

[1] T. Baarslag, M. J. C. Hendrikx, K. V. Hindriks, and C. M. Jonker, "Learning about the opponent in automated bilateral negotiation: a comprehensive survey of opponent modeling techniques," Autonomous Agents and Multi-Agent Systems, vol. 30, no. 5, pp. 849–898, 2016.

[2] J. Gwak and K. M. Sim, "Bayesian learning based negotiation agents for supporting negotiation with incomplete information," in Proceedings of the international multiconference of engineers and computer scientists, vol. 1, 2011, pp. 163–168.

[3] C. C. Lee and C. Ou-Yang, "A neural networks approach for forecasting the suppliers bid prices in supplier selection negotiation process," Expert Systems with Applications, vol. 36, no. 2, pp. 2961–2970, 2009.

[4] Y. Zhou, Q. Luo, and H. Chen, "A novel differential evolution invasive weed optimization algorithm for solving nonlinear equations systems," Journal of Applied Mathematics, vol. 2013, no. 3, pp. 1-18, 2013.

[5] C. Siqi and W. Gerhard, "An efficient and adaptive approach to negotiation in complex environments," in ECAI 2012 - 20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012), p. 228-233, 2012.

[6] A. Rubinstein, "A bargaining model with incomplete information about time preferences," Econometrica: Journal of the Econometric Society, vol.53, no, 5, pp. 1151–1172, 1985.

[7] T. Baarslag, K. Hindriks, and C. Jonker, "Towards a quantitative concession-based classification method of negotiation strategies," in: Kinny D., Hsu J.Y., Governatori G., Ghose A.K. (eds) Agents in Principle, Agents in Practice PRIMA 2011. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, vol. 7047, pp. 143–158, 2011.

[8] P. Faratin, C. Sierra, and N. R. Jennings, "Negotiation decision functions for autonomous agents," Robotics and Autonomous Systems, vol. 24, no. 3, pp. 159–182, 1998.

[9] C. Hou, "Predicting agents tactics in automated negotiation," in Intelligent Agent Technology, 2004.(IAT 2004). Proceedings. IEEE/WIC/ACM International Conference on. IEEE, 2004, pp. 127–133.

[10] K. M. Sim, Y. Guo, and B. Shi, "Adaptive bargaining agents that negotiate optimally and rapidly," in Evolutionary Computation, 2007. CEC 2007. IEEE Congress on. IEEE, 2007, pp. 1007–1014.

[11] B. Sim, Kwang Mong andShi and Y. Guo, "Blgan: Bayesian learning and genetic algorithm for supporting negotiation with incomplete information," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 39, no. 1, pp. 198–211, 2009.

[12] C. Yu, F. Ren, and M. Zhang, "An adaptive bilateral negotiation model based on bayesian learning." In: Ito T., Zhang M., Robu V., Matsuo T. (eds) Complex Automated Negotiations: Theories, Models, and Software Competitions. Studies in Computational Intelligence, Springer, Berlin, Heidelberg, vol.435, pp.75-93, 2013.

[13] D. Zeng and K. Sycara, "Bayesian learning in negotiation," International Journal of Human-Computer Studies, vol. 48, no. 1, pp. 125–141, 1998.

[14] J. Zhang, F. Ren, and M. Zhang, "An innovative approach for predicting both negotiation deadline and utility in multi-issue negotiation," in PRICAI 2014: Trends in Artificial Intelligence. Springer, 2014, pp. 1076–1088.

[15] R. M. Coehoorn and N. R. Jennings, "Learning on opponent's preferences to make effective multi-issue negotiation trade-offs," in Proceedings of the 6th international conference on Electronic commerce. ACM, 2004, pp. 59–68.

[16] R. Marler and J. Arora, "Survey of multi-objective optimization methods for engineering," Structural and Multidisciplinary Optimization, vol. 26, pp. 369–395, 2004.

[17] G. Venter, Review of Optimization Techniques. John Wiley and Sons, Ltd, 2010.

[18] A. R. Mehrabian and C. Lucas, "A novel numerical optimization algorithm inspiredfrom weed colonization," Ecological informatics, vol. 1, no. 4, pp. 355–366, 2006.

[19] J. H. Holland, Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. U Michigan Press, 1975.

[20] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," Journal of global optimization, vol. 11, no. 4, pp. 341–359, 1997.

[21] T. L. Saaty and L. G. Vargas, "Decision making with the analytic network process", International Series in Operations Research & Management Science, 95, Springer US, 2006.

R. Ayachi

R. Ayachi is an assistant professor at Ecole Supérieure des Sciences Economiques et Commerciales de Tunis (ESSEC Tunis), Tunisia and member of LARODEC laboratory (ISG Tunis). Her research interests include decision and reasoning issues.

H. Bouhani

H. Bouhani is a master student at Institut Supérieur de Gestion (ISG Tunis), Tunisia. His research interests are automated negotiation and Multi-agent systems.

N. Ben Amor

N. Ben Amor is a professor at Institut Supérieur de Gestion (ISG Tunis), Tunisia. Her research interests include reasoning and decision making under uncertainty, graphical models, and machine learning.