



# POPPONENT: Highly accurate, individually and socially efficient opponent preference model in bilateral multi issue negotiations



Farhad Zafari<sup>a,b</sup>, Faria Nassiri-Mofakham<sup>a,\*</sup>

<sup>a</sup> Department of Information Technology Engineering, Faculty of Computer Engineering, University of Isfahan, 81746-73441, Hezar Jerib Avenue, Isfahan, Iran

<sup>b</sup> Faculty of Science, Engineering and Technology, Swinburne University of Technology, Melbourne, VIC 3122, Australia

## ARTICLE INFO

### Article history:

Received 4 December 2014

Received in revised form 27 March 2016

Accepted 1 April 2016

Available online 8 April 2016

### Keywords:

Bilateral multi issue negotiation

Opponent modeling

Bidding strategy

Acceptance strategy

Perceptron

Multi bipartite gradient descent

## ABSTRACT

In automated bilateral multi issue negotiations, two intelligent automated agents negotiate on behalf of their owners regarding many issues in order to reach an agreement. Modeling the opponent can excessively boost the performance of the agents and increase the quality of the negotiation outcome. State of the art models accomplish this by considering some assumptions about the opponent which restricts the applicability of the models in real scenarios. In this study, a less restricted technique where perceptron units (POPPONENT) are applied in modeling the preferences of the opponent is proposed. This model adopts the Multi Bipartite version of the Standard Gradient Descent search algorithm (MBGD) to find the best hypothesis, which is the best preference profile. In order to evaluate the accuracy and performance of this proposed opponent model, it is compared with the state of the art models available in the Genius repository. This results in the devised setting which approves the higher accuracy of POPPONENT compared to the most accurate state of the art model. Evaluating the model in the real world negotiation scenarios in the Genius framework also confirms its high accuracy in relation to the state of the art models in estimating the utility of offers. The findings here indicate that this proposed model is individually and socially efficient. This proposed MBGD method could also be adopted in other practical areas of Artificial Intelligence.

© 2016 Elsevier B.V. All rights reserved.

## 1. Introduction

Negotiation is the science and art of resolving any kind of disputes and reaching consensus among human parties. In an automated bilateral multi-issue version of negotiations, intelligent computer agents engage in a cooperative process on behalf of their beneficiaries with different and sometimes contradicting interests, with the objective of achieving an agreement on one or more issues. Recently, with the emergence of ANAC (an annual international Automated Negotiating Agents Competition) [2,3], many new negotiation strategies have been developed. Most of the existing sophisticated negotiation strategies typically consist of a set of fixed modules. In general, as observed in Fig. 1, three main components are distin-

\* Corresponding author. Tel.: +98 (0)31 3793 4510; fax: +98 (0)31 3668 2887.

E-mail addresses: f.zafari@eng.ui.ac.ir, fzafari@swin.edu.au, f\_z\_uut@yahoo.com (F. Zafari), fnasiri@eng.ui.ac.ir, fnasirimofakham@yahoo.com (F. Nassiri-Mofakham).

URL: <http://eng.ui.ac.ir/~fnasiri/> (F. Nassiri-Mofakham).

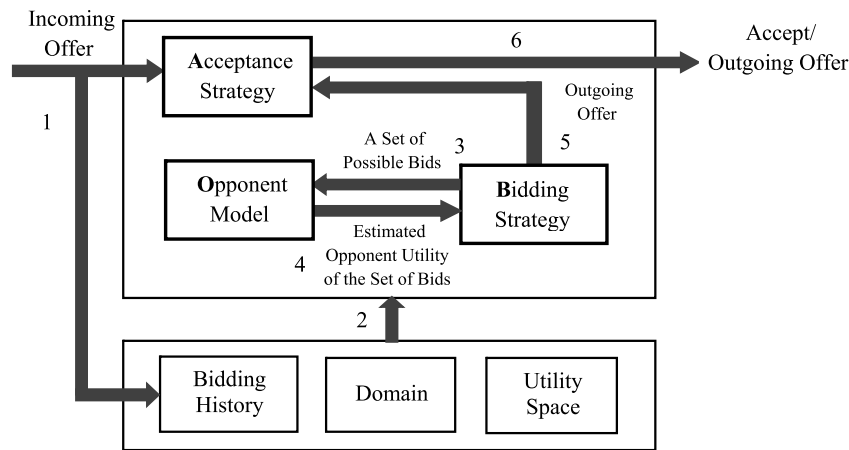


Fig. 1. The negotiation flow (adapted from the BOA framework [1]).

guished in a negotiating agent which work together within a BOA framework to accomplish the whole negotiation task in a collaborative manner [1]:

1. **Bidding Strategy:** this module may interact with the opponent model component by sending one or more candidate offers to the opponent model, and receive the estimated utility of those offers in the utility space of the opponent. Next, the bidding strategy component decides on one of those offers, as the selected candidate offer, to be sent to the opponent as the next proposal.
2. **Opponent Model:** this module constructs a model of the preference profile of the opponent through a learning technique. This model takes a number of offers and generates their estimated utilities.
3. **Acceptance Strategy:** this component receives the incoming offer from the opponent and the outgoing offer chosen by the bidding strategy component, and then determines whether the incoming offer is acceptable for the agent. If the received offer from the opponent is good enough to be accepted, then an *accept* message is provided and sent to the opponent in response. Otherwise, the outgoing offer, previously chosen by the bidding strategy component, is forwarded to the opponent as the response.<sup>1</sup>

According to the BOA framework, a negotiation strategy functions as follows: as soon as the agent receives a new offer<sup>2</sup> from the opponent, the bidding history and the opponent model are immediately updated (steps 1 and 2). This process assures that the agent has the most updated information regarding its opponent. Then in turn, the bidding strategy module generates a number of candidate bids with similar utilities for the agent and sends them to the opponent model. The opponent model then calculates the estimated utilities of the received bids and renders them to the bidding strategy component in response (steps 3 and 4). Next, the bidding strategy component chooses one of the candidate bids according to an opponent model strategy (for example, the best bid is chosen) and forwards this bid to the acceptance strategy module (step 5). Finally, the acceptance strategy decides on whether to accept this newly received bid from the opponent (in step 1) or to send the bid recently received from the bidding strategy component (in step 5).

Identifying offers that are mutually beneficial, avoiding non-agreement offers, and earlier agreements are the benefits of applying an opponent modeling [7]. Despite the variety in opponent modeling techniques, most of the current models rely on a small and common set of learning techniques [7]. It is believed that this is due to the restricting nature of the negotiation problem. The core of the opponent modeling is a learning technique. Moreover, traditional learning techniques are mainly comprised of two non-overlapping learning and classification or prediction phases. Since in a single negotiation session all bids (i.e., training examples) are not available at the same time in advance, traditional learning techniques are not easily applied. An opponent model that is able to learn incrementally and update itself once the new training examples (i.e., offers) arrive during a negotiation session is of necessity. These learning techniques are collectively referred to as adaptive models [8], where both of the aforementioned phases are performed in parallel. Another problem in the negotiation setting is that the training instances lack the output variable (i.e., the variable which contains the utility values of the received bids in the opponent's view). That is, since the agent is not aware of the preference profile of the opponent, it cannot calculate the values of that output variable. Therefore, specific opponent models are required that are capable of modeling the preferences of the opponent with no need for the value of the output variable. Strictly speaking, the model must either use an unsupervised learning technique or somehow pre-estimate the utility values of the received bids from the opponent

<sup>1</sup> An elaborated analysis of the state of the art acceptance strategies and their performance as well as optimal acceptance mechanisms has been carried out by Baarslag et al. [4,5] and Baarslag and Hindriks [6], respectively.

<sup>2</sup> The terms "offer" and "bid" are used interchangeably throughout this article.

and use the estimated values as the output label for the training examples. In order to overcome this limitation, all the existing opponent models use a subset of the following 13 assumptions to extract the preferences of the opponent [9]:

**Assumption 1.** *The bidding strategy of the opponent follows a concession based pattern.* That is, the bidding strategy of the agent generates a sequence of bids in the order of the agent's preference over those bids. This assumption reduces the opponent modeling problem to the problem of estimating the utility of all the bids in the outcome space, so that the estimated utilities are as close to the real utilities as possible.

**Assumption 2.** *The first bid sent by the opponent has the highest possible utility in the opponent's utility space.* That is, the opponent sends its most preferable bid at the beginning of the negotiation session. The first bid embodies the best possible values for each of the negotiable issues. As all rational agents try to maximize their own utilities, it is not surprising that this assumption holds true for most of the rational agents.

**Assumption 3.** *There is an inverse relation between the importance of an issue and the number of times its value changes throughout the negotiation session.* In other words, agents rarely change the value of issues of great importance; the more important the issue, the lower the tendency of the agents to change its value [10,11]. Obviously, the validity of this assumption depends on the agent's bidding strategy and domain.

**Assumption 4.** *There is a direct relation between the importance of an issue value and the number of times it is offered.* Here, the agents seek to offer an issue value of greater importance (greater evaluation value) in the successive offers they send in a negotiation session.

Here, too, the validity of this assumption depends on the agent's bidding strategy and domain.

**Assumption 5.** *The evaluation functions of the negotiation issues are defined by a number of functions with pre-defined shapes (e.g., as in the model proposed by Hinkriks and Tykhonov [12]).* Clearly, the objective of applying this assumption is to limit the number of hypotheses in the opponent's hypothesis space.

**Assumption 6.** *The negotiation issue weight values are simply calculated according to their rankings among all negotiation issue weights (e.g., as in the model proposed by Hinkriks and Tykhonov [12]).* Obviously, like Assumption 5, the objective here is to reduce the opponent's hypothesis space size.

**Assumption 7.** *The utility values of the bids offered by the opponent throughout a negotiation session are distributed around a constant and specific number (such as 1).* Since the rational agents' objective is to maximize their utilities, this assumption is assumed to be valid for rational agents.

**Assumption 8.** *All negotiation issues have equal importance for the opponent.* In other words, all the negotiation issues are of equal weight values [13]. Obviously, the purpose of making this assumption is to limit the size of the opponent's hypothesis space.

**Assumption 9.** *The negotiation issues are conflicting between the two agents.* Here, the evaluation of a negotiation issue value for the opponent is given by 1 minus the evaluation of that negotiation issue value for the agent (e.g., the models proposed by Jazayeriy et al. [14] and Zhang et al. [15]). Obviously, the objective of this assumption is to limit the space size of the opponent's hypothesis.

**Assumption 10.** *The utility values of the offers received from the opponent are known for the agent (e.g., the models proposed by Hinkriks and Tykhonov [12] and Williams et al. [16]).*

**Assumption 11.** *The utility function of the opponent is completely known for the agent.* In other words, the agent has perfect information regarding the preference profile of the opponent.

**Assumption 12.** *The utility value of an offer received from the opponent equals one minus the utility of that offer for the agent.* In other words, the preference profile of the opponent is the opposite of the agent's preference profile.

**Assumption 13.** *The utility value of an offer for the opponent equals the utility of that offer for the agent.* In other words, the preference profile of the opponent is the same as that of the agent.

Another difficulty with modeling an opponent's preferences in bilateral negotiations is related to the time factor. In discounted negotiations [2,3,17], where the utility of an offer is decreased as the negotiation time passes, reaching an agreement as early as possible is of extreme importance. Since constructing an opponent model is costly in the computational sense, the agent should make a trade-off between "non-application of an opponent model, hence saving time" and

“applying one and increasing its utility gained by using the opponent model”. In other words, by non-application the agent would save more time to better explore the outcome space, by looking for an agreement which would result in a gain in his utility. On the other hand, applying an opponent model would assist the agent to make more appropriate agreements. This trade-off is referred to as the time/exploration trade-off [9]. The post event analysis of ANAC tournaments [2,3,17] also confirms that the *computational complexity* of the opponent models and the *poor accuracy* are the two main factors that degrade the performance of the agents applying these models [9]. In particular, the time factor is of paramount importance in online opponent models. In these models, the participating agents usually exchange a limited number of offers before the negotiation deadline is met, therefore, they do not contain enough information to accurately train an opponent model [12,18]. Consequently, the ability of the model to extract the most information possible from the training bids it receives is highly essential. Due to the time restriction and the limited number of offers the agents exchange in their negotiations, a *proper opponent model* is one with the following features:

1. Functionality based on the least assumptions, which would make it more robust against the opponents not following these assumptions.
2. Having the lowest possible computational cost (time/exploration trade-off).
3. Extracting the most information from the least bids (especially important in online opponent models).
4. Incremental training capability.

To learn the issue weight values and individual utility function, a two layered architecture would be essential. Thus, in order to overcome the aforementioned difficulties and to justify the features outlined above, a new model based on *perceptron* units is proposed here for an agent with incomplete information, in order to model the preference profile of the opponent in bilateral multi issue negotiations. Moreover, to be more applicable in real world negotiations, fewer and more realistic assumptions than that of the state of the art models are applied in this study. This is obtained by proposing an opponent model named POPPONENT, based on an adapted version of the standard gradient decent search (named the Multi Bipartite Gradient Decent Search), which applies fewer assumptions. The model shows success in the practical AI area of modeling the user's preferences in bilateral multi issue negotiations with incomplete information.

The remainder of this paper is organized as follows: Section 2 contains the literature review; in Section 3, the general negotiation setting is described; in Section 4, the mathematical concepts underlying perceptron based learning, and the proposed opponent model based on perceptron units are introduced; in Section 5, the experimental setup used in evaluating the proposed model, the carried-out examination, is discussed and the accuracy of this is compared proposed model with some of both the classic and the state of the art opponent models. Next, the efficiency of this model in practice (with real examples) is presented. Finally, in Section 6, the paper is concluded.

## 2. Literature review

In this section, the opponent models from the general and technical perspectives are reviewed in Sections 2.1 and 2.2, respectively. State of the art opponent models currently available in the Genius<sup>3</sup> Repository will be discussed in Section 2.3.

### 2.1. General classification of opponent models

There are many aspects related to the opponent which could be learnt in a negotiation session. Actually, an opponent considers several attributes, based on which they try to learn the existing opponent modeling techniques. These are generally categorized into the following [7]:

**Category 1.** Models that try to estimate *the reservation value* of the opponent [19,20].

**Category 2.** Models that try to estimate *the deadline* of the opponent [19,20].

**Category 3.** Models that try to estimate *the order of the opponent's preferences over different negotiation issues* (the exact value or at least the order of issue weights) [14,15,21,22].

**Category 4.** Models that try to estimate *the order of the opponent's preferences over different negotiation outcomes* (the exact value or at least the order of the utilities of the offers) [12,16,23–26].

**Category 5.** Models that try to learn *the opponent's bidding strategy* [20,27–29].

**Category 6.** Models that try to learn *the opponent's acceptance strategy* [30,31].

The function of the models in the first category is based on the assumption that the agents usually stop conceding close to their reservation values (i.e., the minimum acceptable utility for the agent) and that they exhibit this behavior when approaching negotiation deadline. This means that in the beginning of the negotiation, the agents will offer bids with

far greater utilities than their reservation values [7]. For example, in the model proposed by Gwak and Sim [19] where a simple single issue negotiation is only based on price, the Bayesian method is adopted to estimate the reservation price of the opponent. They assume reservation prices and negotiation deadlines of the agents to be private knowledge. In their model, a number of hypotheses (in the form of  $rv = v_i$ ) are considered as possible values ( $v_i$ ) for the reservation price ( $rv$ ) of the opponent. Next, the Bayesian learning process is applied to update the probability values of the hypotheses in the hypothesis space<sup>3</sup> using the offers received from the opponent during the negotiation session. Then, the weighted average of all hypotheses is calculated and considered as the estimated reservation value of the opponent. In this model, the bidding strategy of the opponent is assumed to follow some kind of a *concession* function. Using the opponent's estimated reservation price (reservation value), the buyer/seller bidding prices are generated in ascending/descending order. Assuming that the opponent uses a pre-known bidding strategy, it is illustrated that the agent could easily compute the opponent's reservation price given its deadline, and vice versa. Given these two value estimations, the agent would then be able to determine the optimal bidding strategy. In their model, the opponent is assumed to have discrete reservation prices, among which each one is a fraction of the difference between the maximum and minimum possible prices considered in that negotiation. This model cannot be adopted in the real world, because real world negotiations are more complex, and often contain more issues than just one price issue. Furthermore, the agents in more complex domains may exhibit more complicated behavior than just simply following a typical concession strategy; accordingly, we need to design a model to operate in more complex and realistic scenarios. As another example, in the model proposed by Yu et al. [20] (similar to the model by Gwak and Sim [19]) the opponent model is used for learning the reservation value and deadline of the opponent (which is assumed to be private information) in a single issue bilateral negotiation. In their article, a model is proposed based on Bayesian learning and regression analysis. This shows how the method is adopted in adapting the agent's strategy to the opponent. Similar to the previous model, this model assumes that the opponent follows a pre-known *concession based strategy* with unknown parameters; that is, this model is not applicable in realistic scenarios where multiple issues are subject to negotiation and the opponent follows more complex bidding strategies.

The second category is run on models that try to estimate the deadline of the opponent. Obviously these models are specifically adopted in scenarios where the opponent has a private deadline. However, in the famous settings like ANAC, it is assumed that the negotiating parties have a common deadline to reach an agreement. For example, both the models proposed by Gwak and Sim [19] and Yu et al. [20] are of this category.

The third category includes the models that try to learn the order of the opponent's preferences over different negotiation issues. In other words, they try to estimate the weights of the issues according to the opponent's view. Faratin et al. [21] believe that obtaining information about the preference order of the negotiation issues from the opponent's perspective would be sufficient to improve the outcome in a negotiation session. Therefore, the problem of learning the opponent's utility function is simply reduced to learning the order of issue weights in the opponent's context. In these models, each possible permutation of the issue weight order is considered as a hypothesis in the hypothesis space, and then the Bayesian formula is applied to update the probability of each hypothesis once a new bid is received from the opponent. For example, the models devised by Niemann and Lang [22], Jazayeriy et al. [14], and Zhang et al. [15] fall into this category. These models also assume restricted types of bidding behaviors for the opponent. For example, the model proposed by Jazayeriy et al. assumes that the opponent follows one of the three following concession strategies: (1) *Boulware*, (2) *Fixed Concession*, and (3) *Conceder*. Moreover, they assume that the agent knows which one of the three aforementioned strategies its opponent follows. Here it is assumed that all of the negotiation issues are monotonic and conflicting, meaning that increasing the utility value of the agent for an issue would cause the opponent's utility for that issue to decline. Obviously, these assumptions do not generally hold true in the real world of negotiations. Contrary to the model presented by Jazayeriy et al. [14], which assumes all the issues are conflicting, in the model by Gwak and Sim [19], it is rather assumed that the agents announce their preference direction as well as the acceptable value ranges for each issue before starting the negotiation session. In other words, the agents' preference directions are considered as common knowledge information in the negotiation. For example, the buyer agent announces that the range of its acceptable prices is from \$100 to \$200. It also declares that there is an inverse relation between the price and its utility. In other words, higher price values will have lower utility values, and vice versa; consequently, both negotiating parties would know the conflicting issues, and would only negotiate on these issues. That is, the agents would be able to easily pick the values with maximum utilities for non-conflicting issues. Thus, there would be no need to negotiate on the values of such issues. It is also assumed that there is an opposite correlation between the weight of an issue and the number of times its value is changed. In the model proposed by Zhang et al. [15], the agent analyzes the history containing the offers received from the opponent and applies Bayesian Learning to learn the opponent's preferences over negotiation issues. The learnt orders of the issue weights are then incorporated into a counter-offer proposition algorithm to enable the agent to propose the offers which are mutually beneficial for both the agent and the opponent. Similar to the model proposed by Jazayeriy et al. [14], in this model it is also assumed that all issues are conflicting. With this assumption, the hypothesis space of possible utility functions will be limited to the possible orders of negotiation issue weights. It is also assumed that the opponent follows a time-dependent concession strategy [33] with known parameters. The counter-offer proposition mechanism used by the agent compensates for the issues of high importance for the opponent. By using this trade-off mechanism, the agent tries to find bids with greater utilities than both the

<sup>3</sup> The term "hypothesis space" is used by Anderson et al. in their Machine Learning book [32] in reference to the space containing all the hypotheses.



agent and the opponent's target utilities. In other words, the agent will try to propose offers that are mutually acceptable for both the agent and the opponent. It is obvious that these assumptions do not necessarily hold true in the real world, so when the opponent does not follow these assumptions, these models are prone to failure.

The models in the fourth category estimate the preference order of the existing bids from the outcome space in the opponent's view. In fact, the models in this category try to estimate the utility function of the opponent and then use this function to calculate the utility of a given bid in the opponent's utility space. Some of the models (e.g. [12,16,23–26]) that are based on the Bayesian formula [34] are classified in this category. These models consider the hypothesis space as the preference order of the negotiation issues as well as evaluation values of each negotiation issue. They update the probability of each hypothesis using the Bayesian formula in a consecutive manner as each offer is received from the opponent through the negotiation session. For example, the model proposed by Hindriks et al. [12] that in fact is the basic model underlying some other Bayesian models (e.g. [16,24,26]) estimates the utility function of the opponent (which includes the estimated values for the weights of negotiation issues and estimated evaluation values for each negotiation issue value) through a Bayesian formula. This model is extremely similar to the model proposed by Zhang et al. [15] but it is different in two ways: first, the hypothesis space contains both the evaluation functions and orders of issue weights. However, in the model proposed by Zhang et al. [15] the hypothesis space is reduced by assuming that all negotiation issues are conflicting, so only the possible ranking of issue weights is assumed as the hypothesis space. Second, in the model proposed by Hindriks et al. [12], a normal distribution is used to calculate the probability of observing a bid given the condition that a hypothesis holds (or  $P(D|h)$ ), but in the model proposed by Zhang et al. [15], upon receipt of a new offer, the best hypothesis which most accurately estimates the utility of this new offer is calculated, and the resultant value is used to estimate the value of  $P(D|h)$  for each of the hypotheses.

The fifth category comprises the models that try to learn the bidding strategy of the opponent. The bidding strategy is defined as a specific function which maps a negotiation state to a bid [7]. In other words, a bidding strategy generates a sequence of successive bids, which are going to be presented to the opponent in every round of the negotiation. This function could be either a *time dependent* or *behavior dependent* (or imitative) function [33]. The strategy proposed by Krimpen et al. [10] is an example of a time dependent function in the bidding strategy presented by Fatima et al. [35]. Moreover, the bidding strategy used by Baarslag et al. [24] is of the behavioral function types that compensates every nice move from the opponent. The models in this category are classified in two classes: (1) *Regression analysis* models, and (2) *Time series forecasting* models [7]. Regression analysis models assume that the bidding strategy of the opponent could be defined by a formula with unknown parameters; that is, the problem of estimating the bidding strategy of the opponent could be easily reduced to the regression analysis on the utility values of the received offers from the opponent in the agent's own utility space. The model proposed by Yu et al. [20] falls into this category. On the other hand, in time series forecasting models, the bidding strategy of the opponent is totally unknown; that is, the agent does not even have any information on the bidding strategy formula. Therefore, the agent uses a time series in order to forecast the utility values of the upcoming received offers in the agent's own utility space. In other words, these models receive a list of timely ordered utility values as the input and generate as the output the probable next utility values. For example, the models devised by Williams et al. [27,29] and Carbonneau et al. [28] fall in this category.

The models in the sixth category try to estimate the acceptance strategy of the opponent. The acceptance strategy of an agent is defined as a Boolean function, which receives a typical bid as the input and produces a Boolean value showing whether the bid would be acceptable by that agent or not. It is important for the agent to learn this function for improving the negotiation outcome, since it could determine the best bid for the agent which is still acceptable for the opponent [7]. One of the works that deals with proposing an acceptance strategy learning method is the article by Lau et al. [30]. In this model, the offers sent by the agent to the opponent are considered as unacceptable bids and the offers received from the opponent are considered as acceptable ones. These bids are then fed into a Bayesian learning model as training examples through which the probability of accepting a candidate bid by the opponent would be determined. The model proposed by Aydogan and Yolum [31] also falls in this category. This model estimates the acceptability of a model through inductive learning. In their model, the sent offers are considered as the negative instances and the received offers are considered as the positive instances, and both are applied in learning the target function.

## 2.2. Technical classification of opponent models

In the general classification, the models are classified based on the aspects of the opponent which are to be learnt. However, according to the underlying scheme in which they are applied to extract the opponent's preferences, the existing state of the art opponent models (which use a common negotiation setting for estimating the preference order of the negotiation outcomes and are designed and implemented in the Genius framework [36,37] in the ANAC tournaments) could be categorized into the following [38]:

1. **The Bayesian models:** estimate the preference profile of the opponent by generating a set of candidate profiles (hypotheses) and then using Bayesian learning to update the probabilities of each hypothesis. The models here make some assumptions on the bidding strategy of the opponent.
2. **The Frequency models:** estimate the *issue weights* and *evaluation values* by considering the changing frequency of the value of each issue between successive bids and the frequency at which each issue value is offered, respectively. Unlike

the Bayesian models, which make strong assumptions about the bidding strategy of the opponent, the models of this group make assumptions on the value distribution of the issues in the opponent's preference profile and impose weak restrictions on the bidding strategy of the opponent [9].

3. **The Value models:** are the same as frequency models except that they assume equal and constant values for issue weights and focus on estimating the evaluation values of each issue value instead. Though assuming constant and equal values for issue weights would degrade the model's accuracy, it would also make the agent free of the need for estimating the issue weights.

Bayesian opponent modeling techniques, allegedly the most prominent and popular probabilistic approaches in opponent modeling [7], generate hypotheses on the opponent's preferences. Upon receiving a new offer from the opponent, they update the probabilities associated with each one of these hypotheses [9,12]. These models are mostly based on the first assumption, that is, these models make strong assumptions on the bidding strategy of the opponent.

On the other hand, *frequency* models (and *value* models) are mostly reliant on the third and fourth assumptions (and in some cases on the second assumption), and rely less on the first assumption. That is, these models usually make assumptions on the value distribution of the issues in the opponent's preference profile and depend less on the opponent's bidding strategy. However, it is not clear whether the opponent would stick to the underlying assumptions of these models. Therefore, due to the uncertainty about the validity of these assumptions through negotiations with different opponents, these modeling techniques are potentially subject to failure, since they strongly count on the opponent to somehow follow a subset of these assumptions [9].

### 2.3. State of the art opponent models

All three classes in the technical classification try to estimate the utility function of the opponent and correspond to the fourth category in the general classification (see Section 2.1). In this article, the frequency models and value models are not differentiated, and the value models are considered as a part of the frequency models. In fact, frequency models are value models which also extract the issue weight values from the opponent's behavioral and verbal reflection. In Section 5, the accuracy of this newly proposed model is compared with the state of the art frequency (and value) models together with Bayesian and classic models, which are available in the Genius repository. These models are briefly explained below:

1. *AgentLG Freq. Model:* each evaluation value is simply estimated based on the frequency at which that value is offered. Issue weights are simply assumed to be uniform [36,37].
2. *AgentX Freq. Model:* a more complex variant of HardHeaded Freq. model, where the tendency of the opponent to repeat bids is of concern. The weight of an issue is computed based on the number of times its value changes [36,37].
3. *CUHKAgent Freq. Model:* the evaluation values are computed based on the frequency at which they are offered. The utility value of a bid is estimated by calculating the sum of the evaluation values for all issues of that bid and dividing it by the best possible score to normalize it into a utility value. Only the first 100 unique bids are used in this model for its estimation. Issue weights are simply assumed to be uniform [39].
4. *HardHeaded Freq. Model:* the evaluation values are calculated based on the frequency at which they are offered. The weight of an issue is calculated based on the number of times its value changes [10].
5. *InoxAgent Freq. Model:* the evaluation values are calculated based on the frequency at which they are offered. The more important an issue value is, the more it is repeated in successive bids. The weight of an issue is calculated based on the number of times its value changes. The higher the importance of an issue, the more unchanged its value remained [36,37].
6. *Nash Freq. Model:* computes the evaluation values based on the frequency at which they are offered. The weight of an issue is calculated based on the number of times the best assumed value of the issue is changed [36,37].
7. *Smith Freq. Model:* the evaluation values are computed based on the frequency at which they are offered. The weight of an issue is calculated based on the distribution of the values of that issue [11].
8. *The Fawkes Freq. Model:* the evaluation values are determined based on their frequency through successive bids. The issue weights are determined based on the number of times their value changes. This model is very similar to *InoxAgent Freq. Model* with a slight difference: the unchanged issue values are determined by comparing the values of the last and first bids received from the opponent [40].
9. *IAMHaggler Bayes. Model:* is an efficient implementation of the Bayesian Scalable model, and is similar to the Scalable Bayesian model; here, it is also assumed that the opponent uses a particular time-dependent concession function [16].
10. *TheNegotiatorReloaded Bayes. Model:* is very similar to the *IAMHaggler Bayes. Model* with a slight difference: it uses different parameters for the supposed concession function of the opponent [40].
11. *Scalable Bayes. Model:* the issue weights and evaluation values are estimated using Bayesian learning. First, it initializes the hypothesis space of all possible preference profiles of the opponent. Next, it repeatedly updates each probability value based on the assumption that the opponent concedes according to a linear function [12].
12. *Perfect IAMHaggler Bayes. Model:* the same as the *IAMHaggler Bayes. Model* except that it is equipped with perfect information [16].
13. *Perfect Scalable Bayes. Model:* the same as the *Scalable Bayes. Model* except that it is equipped with perfect information [12].

**Table 1**

Classification of the state of the art opponent models in Genius repository.

| Opponent model               | Type           | Estimating    |                   | Assumptions |
|------------------------------|----------------|---------------|-------------------|-------------|
|                              |                | Issue weights | Evaluation values |             |
| AgentLG Freq.                | F <sup>a</sup> | No            | Yes               | 8, 4        |
| AgentX Freq.                 | F              | Yes           | Yes               | 4, 3        |
| CUHKAgent Freq.              | F              | No            | Yes               | 8, 4        |
| HardHeaded Freq.             | F              | Yes           | Yes               | 4, 3        |
| InoxAgent Freq.              | F              | Yes           | Yes               | 4, 3        |
| Nash Freq.                   | F              | Yes           | Yes               | 4, 3, 2     |
| Smith Freq.                  | F              | Yes           | Yes               | 4, 3        |
| TheFawkes Freq.              | F              | Yes           | Yes               | 4, 3, 2     |
| IAMHaggler Bayes.            | B              | Yes           | Yes               | 6, 5, 1     |
| TheNegotiatorReloaded Bayes. | B              | Yes           | Yes               | 6, 5, 1     |
| Scalable Bayes.              | B              | Yes           | Yes               | 6, 5, 1     |
| Perfect IAMHaggler Bayes.    | B              | Yes           | Yes               | 10, 6, 5, 1 |
| Perfect Scalable Bayes.      | B              | Yes           | Yes               | 10, 6, 5, 1 |
| Perfect Model                | C              | Yes           | Yes               | 11          |
| No Model                     | C              | Yes           | Yes               | 13          |
| Worst Model                  | C              | Yes           | Yes               | 11          |
| Opposite Model               | C              | Yes           | Yes               | 12          |

<sup>a</sup> Note: F, B, and C stand for Frequency based, Bayesian, and Classic.

14. *Perfect Model*: The agent has perfect knowledge regarding the preferences of the opponent. In other words, the opponent's utility function is known to the agent [36,37].
15. *No Model*: the agent is without any model, hence the estimated utility of a given bid equals the utility of that bid in the opponent's utility space [36,37].
16. *Worst Model*: This model is the opposite of the Perfect Model, that is, the utility of a given bid using the Worst Model equals 1 minus the real utility of the same bid obtained through the Perfect Model [36,37].
17. *Opposite Model*: it is assumed that the estimated utility of a given bid equals 1 minus the utility of the same bid in the agent's own utility space [36,37].

All the models in the Genius repository, together with the assumptions that each adopts in modeling the opponent's preferences, are tabulated in Table 1 (see Section 1). As observed, eight out of the seventeen opponent models are Frequency based. Moreover, except AgentLG and CHUKAgent which only estimate evaluation values and not issue weights, all others estimate both the issue weights and evaluation values.

With respect to the technical classification (Section 2.2), this newly proposed model does not fit into any of the three categories. Hence, a new class is defined, named Perceptron-Based Models, where this proposed model could fit in. Moreover, this proposed model fits best into the fourth category described in the general classification (Section 2.1). Nonetheless, to the best of our knowledge, there is no proposed model in the fourth category which specifically uses neural networks (or neuron units) as the underlying learning technique. Our model is the first proposed model in the fourth category that uses perceptron units as the underlying mathematical basis for opponent modeling.

### 3. Negotiation setting

The negotiation setting here corresponds to the state of the art models' setting in the field of automated negotiations (e.g., [1–3,17,36,38]) and that of the ANAC 2010–2013.<sup>4</sup> Automated agents alternatively exchange offers and compete against each other to reach a joint agreement on a set of issues in bilateral negotiations. The issues and possible values for each issue constitute a *domain*. For each domain there could be two *preference profiles* (one for each side of the negotiation) which together with the domain construct a *negotiation scenario*.

The interaction between negotiating parties is regulated by a *negotiation protocol* that defines the rules of how and when proposals can be exchanged [17]. In this setting, the *alternating offers protocol* is applied [42]. Negotiation is bilateral, that is, exactly two parties are negotiating over one or a set of *issues*. Each issue is associated with a set of possible *values*. The agents repeatedly exchange offers in successive rounds, so as to reach a mutually acceptable outcome. The negotiation deadline is reached after a specified number of rounds  $N$  are passed. This type of negotiation setting is commonly referred to as a *round based setting*. Each agent tries to take advantage of the other party for gaining a maximum utility for its own. A negotiation break-off causes both negotiating parties to obtain their *reservation values*. Therefore, the agents try to reach an agreement before the deadline. A negotiation session takes place in a *negotiation scenario*, which consists of a *negotiation domain* (or, alternatively, an *outcome space*) and two *preference profiles* (or, alternatively, *utility space*) one for each negotiating agent.

<sup>4</sup> In 2014, another setting has been added for negotiations under nonlinear utilities and in ultra large domains. ANAC 2015 considered multi-party negotiations [41].



The negotiation domain  $\Omega$  specifies all possible offers  $\vec{\omega}$  that the agents can send or receive. Each offer or possible outcome is a vector  $\langle \omega_1, \dots, \omega_n \rangle$ , where each component is the mapping of every issue  $i$  to a value  $\omega_i \in \{v_{i1}, \dots, v_{im_i}\}$ , where,  $m_i$  is the number of possible values for issue  $i$ ,  $i = 1, \dots, n$  [17]. A preference profile  $\{\langle \vec{\omega}, U(\vec{\omega}) \rangle \mid \vec{\omega} \in \Omega\}$ , on the other hand, consists of a utility function  $U(\vec{\omega})$  which maps each possible offer  $\vec{\omega} \in \Omega$  to a value in the  $[0, 1]$  range based on the overall relative value of that offer for the agent. In multi-issue negotiations, the common assumption is that the utility of an offer can be computed as a weighted sum of the utilities associated with the values for each issue [12,43–47]. Accordingly, in the negotiation setting here, the agents use the linear additive utility function shown in Equation (1),<sup>5</sup> defined by a set of weights  $w_i$ , and the corresponding evaluation functions or evaluation values  $eval_i(\omega_i)$ ,  $i = 1, \dots, n$ , for the issue value  $\omega_i$  of a given offer  $\vec{\omega}$ :

$$U(\vec{\omega}) = \sum_{i=1}^n w_i \cdot eval_i(\omega_i) \quad (1)$$

In other words, the preference profile of an agent is modeled as a linear combination of a set of weights (which measures the relative importance of the negotiation issues) and a number of *individual utility functions* (or *evaluation functions*) which calculate the utility of a possible value for a negotiation issue. Unlike the negotiation domain which is publicly known for both the negotiating parties, the preference profile is private for each agent, so the agents are not aware of the weights and evaluation values associated with the preference profile of one another. Without losing the applicability of this newly proposed model, for the sake of simplicity in the negotiation setting here it is assumed that each issue value  $\omega_i$  in an offer only takes a finite set of discrete values.

This negotiation setting is online, that is, the agent is only allowed to use the offers exchanged during a single negotiation session to model the preferences of the opponent, so that learning between sessions is not authorized. Unlike *offline opponent models* where negotiation information from different negotiation sessions with similar opponents is applied in modeling the preferences of an opponent, in *online models* (e.g., [10,12]) no history of the previous negotiations is provided for the opponent model [7,9].

#### 4. POPPONENT: perceptron-based opponent model

As explained in Section 2, state of the art models fall into the categories of either Frequency-based or Bayesian models, and they suffer from the dependency on the category-based restricting set of assumptions to model the opponent. In this study, an efficient model is designed based on perceptron units which are less dependent on the assumptions of the opponent. In fact, unlike non-classic state of the art models (Table 1) which depend on a number of restricting assumptions, this model relies only on one assumption, either 1 or 10. In real negotiation problems, all of the training examples (i.e., bids) are not always given in advance, but are gradually acquired one at a time. Therefore, a model is required which is updated right after each bid is encountered. For this purpose, the proposed perceptron-based opponent model (POPPONENT) adopts an *incremental gradient descent* or *stochastic gradient descent* algorithm to efficiently learn the preference profile of the opponent in linear scenarios. The mathematical details behind this approach are presented in Section 4.1. The algorithm of the perceptron-based opponent model is proposed in Section 4.2.

##### 4.1. Mathematical justifications

###### 4.1.1. Perceptron-based learning

In order to enable the agent to learn the preference (issue weight values and individual utility functions) of an opponent during a negotiation session in bilateral multi-issue negotiations, the standard simple perceptron units are adopted, which are used as the basic units that construct the ANN [32,48–50], Fig. 2.

A simple perceptron unit takes a vector of real valued inputs  $(x_1, \dots, x_n)$  and calculates a linear combination of their values through a linear additive function as Equation (3),

$$O(x_1, \dots, x_n) = \begin{cases} 1 & \text{if } w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n > 0 \\ -1 & \text{otherwise} \end{cases} \quad (3)$$

where edge weights  $w_i$  are the real valued parameters that determine the contribution of inputs  $x_i$  to the perceptron output  $O$ , and the quantity  $-w_0$  is a specific threshold (such as 0). When all the inputs are received in a perceptron unit, if the weighted inputs' combination exceeds the threshold, the perceptron produces 1 as an output; otherwise, it produces  $-1$ . The weights are not known in advance, so they have to be somehow adjusted through some *training instances*, a specific *learning algorithm* (like *gradient descent*) and a specific *training rule* (like the *delta rule*) [32]. The error between the real

<sup>5</sup> In discounted scenarios [2,3,17], the utility of an offer decreases over time according to a discount factor. Discounted utility  $U_D$  could be as follows:

$$U_D(\vec{\omega}) = U(\vec{\omega}) \times d^t \quad (2)$$

where,  $d$  ( $0 \leq d \leq 1$ ) and  $t$  ( $0 \leq t \leq 1$ ) are the discount factor and the negotiation time, respectively.

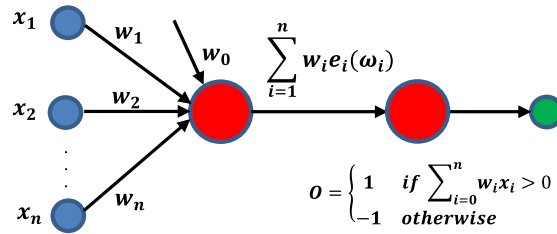


Fig. 2. A simple unit of perceptron [32].

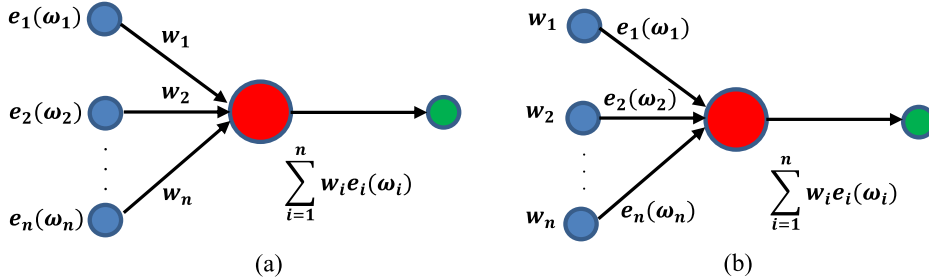


Fig. 3. The Perceptron unit: (a) to learn the issue weights, and (b) to learn the evaluation values.

output and the output produced by perceptron unit is minimized by giving each training example in the training set into the perceptron unit, and adjusting the edge weights according to a training rule (like the delta rule) in a successive manner.

#### 4.1.2. Multi bipartite incremental gradient descent search

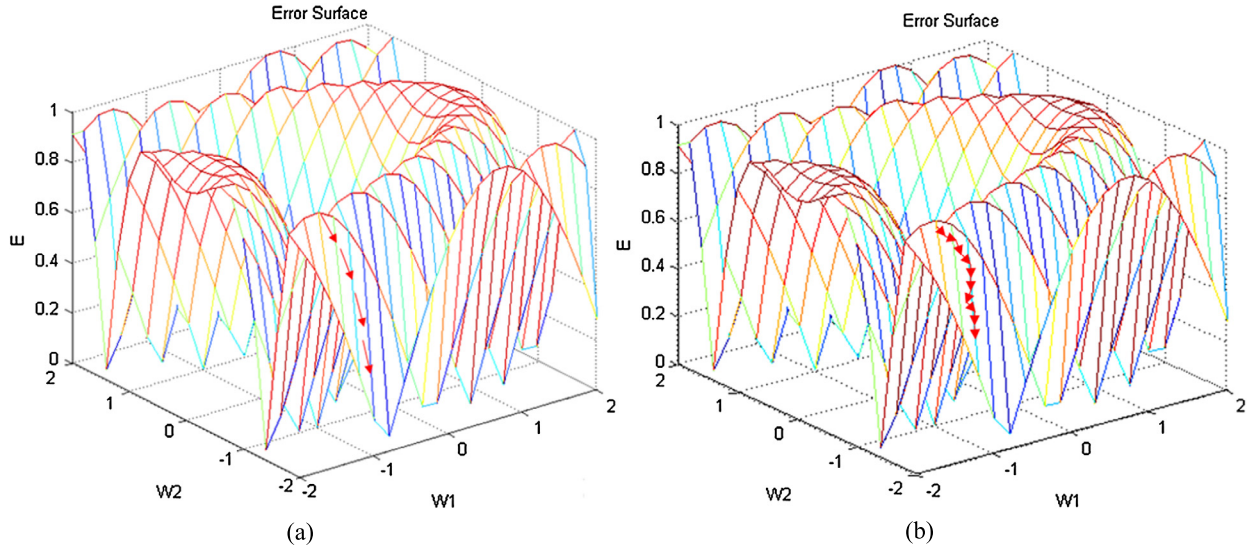
As mentioned in Section 3, in bilateral multi-issue negotiations, the preference profile of an opponent is private to the agent. Therefore, the agent has to learn it through offers as *training instances* received from the opponent and through a specific *learning algorithm*. In linear scenarios, a preference profile is comprised of a number of negotiation issue weights  $w_i^{OP}$  or priorities as well as the evaluations  $eval_i^{OP}(\omega_i)$  corresponding to each possible value  $\omega_i \in \{v_{i1}, \dots, v_{im_i}\}$ , where,  $m_i$  is the number of possible values for issue  $i$  ( $i = 1, \dots, n$ ). Since both the issue weights and evaluations of the opponent's preference profile are unknown to the agent, this proposed model applies *two types of perceptron units* [32] for learning the opponent's preferences through a multi bipartite incremental gradient descent search. The two constituting parts of this algorithm execute two separate overlapping steps: in the first step, Fig. 3(a), the evaluation values and the issue weights are assumed to have become the inputs to the perceptron unit and the weight vector elements of the perceptron, respectively. Next, the gradient descent is applied to adjust the issue weights (perceptron type 1); in the second step shown in Fig. 3(b), the issue weights and the evaluation values are assumed to have become the inputs to multiple copies of the perceptron unit and the weight vector of the perceptron units, respectively. Next, the evaluation values are adjusted by feeding the issue weight values as the input into the perceptron units (perceptron type 2). In other words, the objective of these perceptron units is to efficiently learn both the issue weight values  $w_1^{OP}$  through  $w_n^{OP}$  and the evaluation values  $eval_1^{OP}(\omega_1)$  through  $eval_n^{OP}(\omega_n)$  for all possible values  $\omega_i \in \{v_{i1}, \dots, v_{im_i}\}$  of each issue  $i$  in a given negotiation domain, since for each issue  $i$ ,  $\omega_i \in \{v_{i1}, \dots, v_{im_i}\}$ , there are  $m_i$  possible evaluation values for issue  $i$ . This means that the total number of perceptron units (of type 2) for learning the evaluation values is calculated through Equation (4):

$$n_2 = \prod_{i=1}^n m_i \quad (4)$$

Moreover, in order to learn the negotiation weight values, only one perceptron unit (of type 1) would suffice, that is, in total,  $n_2 + 1$  perceptron units are needed to learn the preference profile of the opponent. Here, two types of perceptrons work next to one another in a manner in which at any given time the perceptron type 1 works together with one single perceptron from type 2.

Upon receipt of a new offer  $\langle \omega_1, \dots, \omega_n \rangle$  from the opponent, the perceptron unit calculates the utility of that offer, and the perceptron learning algorithm is used to learn the preference profile of the opponent. By inserting each training example into the perceptron, the algorithm tries to estimate the real preference profile of the opponent by minimizing the error between the produced and real output, by adjusting the weight vector values of the perceptron unit in an iterative manner.

The learning problem in Fig. 3(a) determines the weights  $w_1^{OP}$  through  $w_n^{OP}$ , in the sense that the training error of the learnt hypothesis (which is a vector of weight values) is minimized. The learning problem in Fig. 3(b) determines the evaluation values  $eval_i^{OP}(\omega_i)$  in the sense that the training error of the learnt hypothesis (which is a vector of evaluation values) is minimized.



**Fig. 4.** Moves for locating the global minimum in the plane of error values associated with different hypotheses (i.e., weight vectors) using (a) gradient descent, and (b) incremental gradient descent [32].

The hypothesis space of possible weights ( $w_1$  and  $w_2$  in a sample 2-issue domain) and the associated error values are presented in Fig. 4. To find a minimum point, first, the *incremental gradient descent* or *stochastic gradient descent* optimization algorithm begins with an initial point on the error surface, and next, it repeatedly updates the weight values in the weight vector so that the search is directed towards the steepest descent along the error surface. In this algorithm, the weight values are updated right after each single training example is encountered in an incremental manner. This process continues until the global (or local) minimum is achieved.

The error in modeling the opponent's preference profile is formulated as follows:

$$E(\vec{k}) = \sum_{d(\omega) \in D} E_d(\vec{k}) \quad (5)$$

where  $E_d(\vec{k})$  is the individual error for the training instance  $\vec{d}(\omega)$ , the value of which is calculated by Equation (6):

$$E_d(\vec{k}) = \frac{1}{2} (t_d - o_d)^2 \quad (6)$$

where  $\vec{k} = \langle \vec{w}^{OP}, \vec{eval}^{OP} \rangle$  is the opponent preference profile,  $\vec{w}^{OP} = \langle w_1^{OP}, \dots, w_n^{OP} \rangle$  is the issue weights vector,  $D = \{ \vec{d}(\omega) \mid \vec{d}(\omega) = \langle eval_1^{OP}(\omega_1), \dots, eval_n^{OP}(\omega_n) \rangle, \omega_i \in \{v_{i1}, \dots, v_{im_i}\}, i = 1, \dots, n \}$  is the set of training examples,  $t_d$  is the target output (real output value) for the training example  $\vec{d}(\omega)$ , and  $o_d$  is the output of the linear unit for the training example  $\vec{d}(\omega)$ . The vector  $\vec{eval}^{OP} = \langle eval_1^{OP}(v_{11}), \dots, eval_1^{OP}(v_{1m_1}), \dots, eval_n^{OP}(v_{n1}), \dots, eval_n^{OP}(v_{nm_n}) \rangle$  contains all the evaluation values for all the possible values for all issues. The set of training examples or  $D$  includes the evaluation values for *all* possible offers in the outcome space, its cardinality is calculated through Equation (4).

According to the function which delivers the utility value of a bid (Equation (1)), the utility value of a given training instance  $\vec{d}(\omega)$  is calculated through Equation (7):

$$o_d = \sum_{i=1}^n eval_i^{OP}(\omega_i) \cdot w_i^{OP} \quad (7)$$

By inserting Equation (7) into (6) and (6) into (5), Equation (8) is yielded:

$$E(\vec{k}) = \frac{1}{2} \sum_{d(\omega) \in D} \left( t_d - \sum_{i=1}^n eval_i^{OP}(\omega_i) \cdot w_i^{OP} \right)^2 \quad (8)$$

which represents the training error of all training examples. In Equation (8),  $\omega_i$  is the value of issue  $i$  in the training instance  $\vec{d}(\omega)$ .

The objective here is to minimize the error  $E$  by searching for the best preference profile which yields the minimum error between the estimated offer utility values, according to the bidding behavior of the opponent ( $t_d$ ) and the estimated offer utility values according to the current values of the preference profile of the opponent ( $\sum_{i=1}^n eval_i^{OP}(\omega_i) \cdot w_i^{OP}$ ) for all

the training examples in  $D$ . The direction of steepest descent along the error surface can be determined by computing the gradient or derivative of  $E$  with respect to each component of the opponent profile vector  $\vec{k}$ . This derivative vector, with respect to the vector  $\vec{k}$ , can be calculated through Equation (9):

$$\nabla E(\vec{k}) = \nabla E(\vec{w}^{OP}, \vec{eval}^{OP}) = \left[ \frac{\partial E}{\partial w^{OP}}, \frac{\partial E}{\partial eval^{OP}} \right] \quad (9)$$

Since the gradient calculated based on Equation (9) specifies the direction that yields the steepest increase in  $E$ , the negative of this value yields the direction of steepest decrease. The training rule for gradient descent search is expressed through Equation (10):

$$\Delta \vec{k} = -\eta \nabla E(\vec{k}) \quad \text{and} \quad \vec{k} = \vec{k} + \Delta \vec{k} \quad (10)$$

where the positive constant  $\eta$  is the learning rate which determines the step size in the gradient descent search. Writing Equation (10) in the componential form would yield:

$$\Delta k_i = -\eta \frac{\partial E}{\partial k_i} \quad \text{and} \quad k_i = k_i + \Delta k_i \quad (11)$$

where  $i$  is between 1 and  $\sum_{i=1}^n m_i + n$ ,  $m_i$  is the number of possible values for the issue  $i$ , and  $n$  is the number of negotiation issues, that is, the total number of the components in vector  $\vec{k}$  equals the total number of negotiation issues ( $n$ ) plus the total possible values of all issues ( $\sum_{i=1}^n m_i$ ). The vector  $\vec{k}$  in Equation (11) contains both the *issue weights vector* ( $\vec{w}^{OP}$ ) and the *evaluation values vector* ( $\vec{eval}^{OP}$ ). Provided that component  $k_i$  is a weight value, the following equations are yielded.

$$\Delta w_i^{OP} = -\eta \frac{\partial E}{\partial w_i^{OP}} \quad \text{and} \quad w_i^{OP} = w_i^{OP} + \Delta w_i^{OP} \quad (12)$$

In order to achieve the steepest descent each component  $w_i^{OP}$  of the weight vector  $\vec{w}^{OP}$  in proportion to  $\frac{\partial E}{\partial w_i^{OP}}$  should be altered. Applying Equation (8), the gradient of  $E$  with respect to  $w_i^{OP}$  in Equation (12) is calculated through Equation (13).

$$\begin{aligned} \frac{\partial E}{\partial w_i^{OP}} &= \frac{\partial}{\partial w_i^{OP}} \frac{1}{2} \sum_{d(\omega) \in D} (t_d - o_d)^2 \\ &= \frac{1}{2} \sum_{d(\omega) \in D} \frac{\partial}{\partial w_i} (t_d - o_d)^2 \\ &= \frac{1}{2} \sum_{d(\omega) \in D} 2(t_d - o_d) \frac{\partial}{\partial w_i^{OP}} (t_d - o_d) \\ &= \sum_{d(\omega) \in D} (t_d - o_d) \frac{\partial}{\partial w_i^{OP}} (t_d - o_d) \\ &= \sum_{d(\omega) \in D} (t_d - o_d) \frac{\partial}{\partial w_i^{OP}} \left( t_d - \sum_{i=0}^n eval_i^{OP}(\omega_i) \cdot w_i^{OP} \right) \\ &= \sum_{d(\omega) \in D} (t_d - o_d) (-eval_i^{OP}(\omega_i)) \end{aligned} \quad (13)$$

Now, by combining Equations (12) and (13), the training rule for updating the weight values is obtained as follows:

$$w_i^{OP} = w_i^{OP} + \eta \sum_{d(\omega) \in D} (t_d - o_d) \cdot eval_i^{OP}(\omega_i) \quad (14)$$

Once more, provided that component  $k_i$  is an evaluation value, Equation (11) will be modified into:

$$\Delta eval_i^{OP}(v_{ij}) = -\eta \frac{\partial E}{\partial eval_i^{OP}(v_{ij})} \quad \text{and} \quad eval_i^{OP}(v_{ij}) = eval_i^{OP}(v_{ij}) + \Delta eval_i^{OP}(v_{ij}) \quad (15)$$

where  $j$  is between 1 and  $m_i$  ( $m_i$  is the number of possible values for issue  $i$ ), and  $i$  is between 1 and  $n$  ( $n$  is the number of issues).

Similar to Equation (13), the training rule applied in learning the evaluation values is expressed as Equation (16).

$$\frac{\partial E}{\partial eval_i^{OP}(v_{ij})} = \frac{\partial}{\partial eval_i^{OP}(v_{ij})} \frac{1}{2} \sum_{d(\omega) \in D} (t_d - o_d)^2 \quad (16)$$

$$\begin{aligned}
&= \frac{1}{2} \sum_{d(\omega) \in D} \frac{\partial}{\partial eval_i^{OP}(v_{ij})} (t_d - o_d)^2 \\
&= \frac{1}{2} \sum_{d(\omega) \in D} 2(t_d - o_d) \frac{\partial}{\partial eval_i^{OP}(v_{ij})} (t_d - o_d) \\
&= \sum_{d(\omega) \in D} (t_d - o_d) \frac{\partial}{\partial eval_i^{OP}(v_{ij})} (t_d - o_d) \\
&= \sum_{d(\omega) \in D} (t_d - o_d) \frac{\partial}{\partial eval_i^{OP}(v_{ij})} \left( t_d - \sum_{i=0}^n eval_i^{OP}(\omega_i) \cdot w_i^{OP} \right)
\end{aligned}$$

If there exists  $d(\omega) \in D$ , that  $\omega_i = v_{ij}$ , then Equation (17) is yielded:

$$\frac{\partial E}{\partial eval_i^{OP}(v_{ij})} = \sum_{d(\omega) \in D} (t_d - o_d) (-w_i^{OP}) \quad (17)$$

If not, Equation (18) is yielded:

$$\frac{\partial E}{\partial eval_i^{OP}(v_{ij})} = 0 \quad (18)$$

Combining Equations (15), (17), and (18) would yield:

$$eval_i^{OP}(v_{ij}) = \begin{cases} eval_i^{OP}(v_{ij}) + \eta \sum_{d(\omega) \in D} (t_d - o_d) \cdot w_i^{OP} & \forall d(\omega) \in D \mid \omega_i = v_{ij} \\ eval_i^{OP}(v_{ij}) & otherwise \end{cases} \quad (19)$$

Equation (19) means that the evaluation value for the value  $v_{ij}$  (i.e.,  $eval_i^{OP}(v_{ij})$ ) remains unchanged as long as it is not met in a training example.

At this point, it could be deduced that in order to make a movement in the steepest descent direction in the error surface of different preference profiles, calculating the  $eval_i^{OP}(v_{ij})$  and  $w_i^{OP}$  values according to Equations (14) and (19), respectively, would suffice, hence the updating of the opponent preference profile.

When a sufficiently small learning rate  $\eta$  is used, the gradient descent search assures convergence into a vector with minimum error, regardless of whether the training examples are linearly separable or not. If  $\eta$  is too large, this search approach is at risk of exceeding the vector with minimum error in the error surface. Therefore, using an appropriate learning rate is of the essence [32].

This gradient descent-based opponent modeling algorithm could be subject to two major drawbacks, namely: (1) converging to a local minimum can sometimes be quite slow, and (2) if there are multiple local minima in the error surface, then there is no guarantee that the procedure will find the global minimum [32]. Besides this, in real world negotiation problems, all of the offers (training examples) are not always provided in advance. If all training examples were available in advance, reaching one of the global minimums would mean obtaining 100 percent accuracy (or zero error) in predicting the output values through the input values for the set of available training instances.

These difficulties are alleviated by the *incremental* or *stochastic* version of the *gradient descent*. As mentioned, non-incremental gradient descent first calculates  $\Delta w_i^{OP}$  and  $\Delta eval_i^{OP}(v_{ij})$  values separately for all training instances through the target values provided by the current evaluation values, vector  $eval^{OP}$ , and the weight vector,  $w^{OP}$ , and then applies them in calculating the  $w_i^{OP}$  and  $eval_i^{OP}(v_{ij})$  values, respectively (through Equations (14) and (19)). However, in incremental or stochastic gradient descent,  $eval_i^{OP}(v_{ij})$  and  $w_i^{OP}$  values are updated right after each offer is received, according to the following delta equations:

$$eval_i^{OP}(v_{ij}) = \begin{cases} eval_i^{OP}(v_{ij}) + \eta(t_d - o_d) \cdot w_i^{OP} & \text{if } \omega_i = v_{ij} \\ eval_i^{OP}(v_{ij}) & \text{otherwise} \end{cases} \quad (20)$$

$$w_i^{OP} = w_i^{OP} + \eta(t_d - o_d) \cdot eval_i^{OP}(\omega_i) \quad (21)$$

Equation (20) reveals that in the incremental gradient descent, not all the evaluation values are updated, but only those that have been met in the training instance  $d(\omega)$ , hence, Equation (20) can be rewritten as Equation (22).

$$eval_i^{OP}(\omega_i) = eval_i^{OP}(\omega_i) + \eta(t_d - o_d) \cdot w_i^{OP} \quad (22)$$

Now, in order to find the best preference profile, in the incremental gradient descent algorithm, after each training instance  $d(\omega)$  is met, Equations (21) and (22) are applied, and the evaluation values for the values that have been observed in the bid  $(\omega_1, \dots, \omega_n)$  and their weight values are updated.

Having introduced the mathematical basics of this algorithm, in Section 4.2 we present the algorithm based on the incremental gradient descent learning to extract the preference profile of the opponent.



#### 4.2. POPPONENT algorithm

The newly introduced *multi bipartite incremental gradient descent* or *stochastic gradient descent search* (Section 4.1.2) is applied here to learn the preference profile of the opponent in linear scenarios. To serve the purpose, it learns the issue priorities or weight values  $w_1^{OP}$  through  $w_n^{OP}$  and the evaluation values  $eval_i^{OP}(\omega_i)$  for all possible values  $\omega_i \in \{v_{i1}, \dots, v_{im_i}\}$  (where,  $m_i$  is the number of possible values for issue  $i$ ) and all negotiation issues  $i$  ( $i = 1, \dots, n$ ) in that negotiation domain.

This proposed algorithm is an incremental version of the perceptron-based learning method (Section 4.1) and applies two parameters of  $\eta$  and  $N$  as the input. Parameter  $\eta$  represents the learning rate which determines the step size in the gradient descent search. Parameter  $N$  represents the number of training repeats for each training instance. This algorithm includes two separate functions of initializer and updater.

The first function is invoked just once when the model is generated and the essential parameters of the proposed model, most importantly the preference profile of the opponent (that is, the issue weights  $w_i^{OP}$  and evaluations  $eval_i^{OP}(\omega_i)$  for issue values) is initialized. By trying different initial points, it is realized that 0.5, the midpoint in the hypothesis space, is the best point for initializing  $eval_i^{OP}(\omega_i)$  values for each  $\omega_i \in \{v_{i1}, \dots, v_{im_i}\}$ ,  $i = 1, \dots, n$  (Algorithm: line 13). Similarly, for the weight values  $w_i$ , equal weights  $\frac{1}{n}$  are chosen for all issues (Algorithm: line 14).

The second function receives an offer vector  $\vec{\omega}$  which specifies the issue values for all the negotiation issues of a new offer recently received from the opponent. That is, as soon as a new offer is received from the opponent, this function is invoked to update the model based on this newly received bid. It updates the estimated preference profile of the opponent by adjusting  $eval_i^{OP}(\omega_i)$  and  $w_i$  values (Algorithm: lines 18 and 21). Whenever a new offer is received from the opponent, the perceptron learning delta rules (Algorithm: lines 18 and 21) are repeated  $N$  times. The  $EstimatedUtility^{OP}(\vec{\omega})$  is a function which receives an offer as the input and returns the estimated utility value of that offer in the opponent's utility space as the output.

In this algorithm, instead of updating each  $eval_i^{OP}(\omega_i)$  and  $w_i^{OP}$  value after calculating all  $\Delta w_i$  and  $\Delta eval_i^{OP}$  values when all training examples are met, each  $eval_i^{OP}(\omega_i)$  and  $w_i^{OP}$  value is modified using training delta rules right after each single training instance is met in an incremental manner (Algorithm: lines 18, 21). Therefore, this algorithm can easily be applied in more realistic negotiation scenarios in which training examples (opponent offers) are gradually met one at a time.

The algorithm:

---

##### Algorithm: Perceptron-based opponent model

---

- 1: Where,
  - 2:  $\eta$  is the learning rate.
  - 3:  $N$  is the maximum number of training repeats for each training instance.
  - 4:  $\vec{\omega} = (\omega_1, \dots, \omega_n)$  is the offer vector containing the values of a new offer received from the opponent.
  - 5:  $n$  is the number of issues.
  - 6:  $m_i$  is the number of possible values for  $i$ -th issue.
  - 7:  $\omega_i$  denotes a possible value for the  $i$ -th issue.
  - 8:  $w_i^{OP}$  is the weight of the  $i$ -th issue.
  - 9:  $eval_i^{OP}$  is the evaluation function of the opponent for the  $i$ -th issue.
  - 10:  $\omega_i$  is the value of  $i$ -th issue for the offer  $\vec{\omega}$ .
  - 11:  $PerceptronUtility^{OP}(\vec{\omega})$  is the estimated utility of the offer  $\vec{\omega}$  in the opponent's utility space, which is obtained by feeding the offer into the perceptron unit and getting the output value.
  - 12:  $EstimatedUtility^{OP}(\vec{\omega})$  is a function which receives an offer  $\vec{\omega}$  as the input and returns the estimated utility of that bid in the opponent's utility space as the output. This function value is calculated using the estimated bidding behavior of the opponent.
- 

##### Initializer

---

- 13: Initialize all evaluation values  $eval_i^{OP}(\omega_i)$  to 0.5, for each  $\omega_i \in \{v_{i1}, \dots, v_{im_i}\}$  ( $i = 1, \dots, n$ ).
  - 14: Initialize each weight value  $w_i^{OP}$  to  $\frac{1}{n}$ .
- 

##### Updater

---

- 15: Repeat the following process  $N$  times
  - 16: For each evaluation value  $eval_i^{OP}(\omega_i)$ , Do
  - 17: Input the instance  $\vec{\omega}$  to the unit and compute the output  $PerceptronUtility^{OP}(\vec{\omega})$ .
  - 18:  $eval_i^{OP}(\omega_i) \leftarrow eval_i^{OP}(\omega_i) + \eta \cdot (EstimatedUtility^{OP}(\vec{\omega}) - PerceptronUtility^{OP}(\vec{\omega}))$ .
  - 19: For each issue weight value  $w_i$ , Do
  - 20: Input the instance  $\vec{\omega}$  to the unit and compute the output  $PerceptronUtility^{OP}(\vec{\omega})$ .
  - 21:  $w_i^{OP} \leftarrow w_i^{OP} + \eta \cdot (EstimatedUtility^{OP}(\vec{\omega}) - PerceptronUtility^{OP}(\vec{\omega})) \cdot eval_i^{OP}(\omega_i)$ .
-

This proposed model is a *supervised* algorithm, thus it needs output labels for training instances. The problem of preference modeling in bilateral multi issue negotiations through supervised learning methods can be separated in two sub-problems: 1) estimating the utility values of the opponent's offer history (the history of the offers received from the opponent through the negotiation session) and 2) extracting the estimated utility function (or the preference profile) of the opponent from the opponent's offer history. By solving the first problem, now the opponent's offer history contains all the estimated offer utilities for each offer in the opponent's offer history. The  $EstimatedUtility^{OP}(\vec{w})$  function which deals with the first sub-problem, can be estimated according to the perceived bidding behavior of the opponent.

In this article, four different values – three constant (0.6, 0.8, and 1) and one adaptive – are applied in order to estimate the utilities of the offers proposed by the opponent. Applying the constant values for estimating the utility value of the bids received from the opponent may seem simplistic at first, but actually it is not. As explained in Section 1, the opponent modeling problem in negotiation is a specific type of a learning problem, where all of the training examples are not provided in advance, but are provided throughout the negotiation session in an incremental manner. In fact, designing a preference model which is able to update in an incremental manner is one of the important challenges in modeling the user preferences in bilateral multi issue negotiations. More importantly, in such models the number of bids that the agents exchange before reaching deadline is limited [2], and this is another important challenge that an opponent model should overcome. Although estimating the utilities of the bids received by the opponent through a constant function may not be the best choice, it could at least be a reasonable assumption, whereas the rational agents try to reach the highest utility by following a concession-based bidding strategy [2]. For example, 1) assuming “most of the bids that the opponent sends throughout the negotiation session have utilities around a value such as 1” would not be an unreasonable assumption, and 2) in the *Scalable Bayesian Model* proposed by Hindriks et al. [2], they assume that the utility of the bids sent by the opponent follow the  $1 - 0.05t$  ( $0 \leq t \leq 1$ ) function which places the utility of the bids received from the opponent into the  $[0.95, 1]$  interval. Obviously, this assumption would not be very different from assuming that the opponent mostly sends the bids with the utility value 1.

For the fourth value, we use the adaptive method where the agent estimates the bids that the opponent will offer in future, based on the opponent's bid history [51–55]. At time  $t$ , the opponent will offer a bid with the utility  $target(t)$  the value of which is calculated based on Equations (23) and (24):

$$emax(t) = \mu(t) + (1 - \mu(t))d(t) \quad (23)$$

$$target(t) = 1 - (1 - emax(t))t^\alpha \quad (24)$$

where,  $emax(t)$  estimates the maximum utility of the bid that the opponent will offer in the future,  $\mu(t)$  is the average utility of the bids proposed by the opponent in the agent's own utility space, and  $d(t)$  determines the width of the bids received from the opponent and is calculated based on deviation.

Similar to the approach applied by CHUKAgent (see Section 2) [39], this adaptive model applies only the first 1000 (instead of all 5000) unique bids in a negotiation session for its estimation in order to prevent its accuracy decline (see Section 5.3.1).

The computational complexity of POPPONENT Algorithm is linear ( $O(n)$ ), and is analyzed in Appendix A.

## 5. Experiments

To evaluate the proposed POPPONENT model, two separate experimental settings are applied for assessing its accuracy and performance (in real world negotiation examples) compared to the available opponent models.

### 5.1. Experimental setting

As reviewed in Section 2, different settings and measures are applied by the researchers for evaluating their proposed models. The Genius framework, provided by ANAC, considers all the measures available in this field (see Appendix B) and presents the most extensive and comprehensive settings. The Genius framework facilitates development and evaluation of negotiation agents and their constituting components through a repository of the largest comparable set of models consistent with ANAC settings<sup>6</sup> [38]. Accordingly, a similar experimental setting is adopted here to assess the accuracy and performance of this POPPONENT model.

The first setting, introduced in Section 5.1.1, evaluates the accuracy of POPPONENT through the Pearson Correlation measure. The second setting, introduced in Section 5.1.2, evaluates this proposed model by measuring the real performance of the agents applying this model in real world experimental negotiation scenarios.

#### 5.1.1. Experiment I: evaluating the accuracy of POPPONENT

The experimental setting applied by Baarslag et al. for automated bilateral multi issue negotiations [38] is also applied here to evaluate the accuracy of this proposed POPPONENT model versus the state of the art opponent models.

<sup>6</sup> Baarslag et al. proposed a setting for evaluating and comparing the accuracy and performance of the set of state of the art opponent models currently available in the Genius repository [38].

According to this setting, 5 variations of POPPONENT (Section 4.2) are compared with a total of 15 opponent models including 8 Frequency-based opponent models, 5 Bayesian opponent models and 1 classic model (Opposite model<sup>7</sup>) together with a No Model. These models are tabulated in Table 1.

Each one of these 20 ( $= 8 + 5 + 1 + 1 + 5$ ) *opponent models* is equipped with a bidding strategy to compete with an opponent agent. The *bidding* behavior of an opponent could be influenced by the opponent model of the agent in two manners: first, as the opponent model helps the agent to make better offers, it may cause the negotiation to end earlier with a high quality agreement, hence a shorter agreement, and second, since applying an opponent model would change the bidding sequence of the agent, it may also cause the opponent to change its bidding behavior. For example, if the opponent applies some kind of an imitative bidding strategy [33], different bidding sequences for the agent will definitely result in different bidding sequences from the opponent. Therefore, to be able to compare the different opponent models, all the models being trained under the same circumstances should be confirmed to allow the non-adaptive opponents to be applied for training this proposed opponent model as well as the other models.

Here the *acceptance* capability from the negotiation strategy of the opponent is removed, that is, because an equal number of bids are to be exchanged through all the sessions. This ensures that all the opponent modeling techniques are actually being executed under the same conditions.

**5.1.1.1. Bidding behaviors of the agents** Once more, in accordance with the setting suggested by Baarslag et al. [38], we use four classes of different *opponents*, with different bidding behaviors applied here as follows:

1. *Conceding agents* apply a time dependent concession strategy [33] to make the bids during the negotiation session, that is, the target utility is calculated through Equation (25):

$$u_t = P_{max} \cdot (1 - t^{1/e}) \quad (25)$$

where, for concession rate  $e$ , four different values of 0.1, 0.2, 1, and 2 are applied with a fixed starting bid utility  $P_{max} = 1$ .

2. *Random agents* generate a bid with a utility above a fixed threshold  $m$  in a random manner. Four different values 0, 0.25, 0.5, and 0.75 are selected for the threshold.
3. *Conceding agents with an offset* are time-dependent agents which do not start the negotiation with their best bids. For this purpose, a linear concession rate ( $e = 1$ ), with different values of 0.7, 0.8, and 0.9 is applied in the starting bid utility  $P_{max}$ .
4. *Non-conceding agents*, which begin with a minimum target utility that increases to its maximum utility over time. The target utility is calculated through Equation (26):

$$u_t = P_{min} + (1 - P_{min}) \cdot t \quad (26)$$

where four different values of 0, 0.25, 0.5, and 0.75 are applied in the minimum acceptable utility  $P_{min}$  for the agent.

In accordance with these values, 4 types of conceding agents, 4 types of random agents, 3 types of conceding agents with an offset, and 4 types of non-conceding agents are obtained. Moreover, it is worth noting that the first class reveals a *predictable* bidding behavior, since it always begins with its best offer, always concedes, and makes a unique trace, while the other three classes are all considered to be *unpredictable*, since they make random offers or do not concede or begin with their best offers. From the four classes above, the second contains opponents with random or *non-deterministic* behavior, while the others comprise *deterministic* opponents. Consequently, the experiment involves a total of 15 opponents, among which 4 opponents (in the first class) are predictable, and the remaining 11 are unpredictable.

As mentioned in Section 1, all of the existing models apply a subset of assumptions on the bidding strategy or on the value distribution of the issues of the opponent's preference profile. Here, different opponents with different bidding strategies are applied in training the opponent model and to see how accurate each opponent model actually would be, when these assumptions would not hold true in the real world.

**5.1.1.2. Negotiation scenarios** According to Baarslag et al. [38], there exist the following three features of a negotiation scenario that significantly influence the ability of the opponent model in estimating the opponent's preferences in an accurate manner [38]:

1. The *Domain size* can be small, medium, or large based on its total number of possible offers and depending on the amount of parameters of the preference profile.
2. The *Bid distribution* relates to the average distance of outcomes of a scenario to the nearest Pareto optimal offers. In a high bid distribution domain, a high percentage of outcomes are distanced far from the Pareto frontier.<sup>8</sup>

<sup>7</sup> ANAC organizers did not consider the Perfect Model and Worst Model in the setting, since it is already clear what their Pearson Correlation values would be. According to their definitions, the Perfect Model and Worst Model always yield accuracy values of 1 and -1, respectively.

<sup>8</sup> *Pareto frontier*: The collection of all Pareto Bids in an outcome space [44,56–59].

**Table 2**  
Negotiation setting for Experiment I.

| Opponents  |   | Scenarios  |
|--|---|--|
| Bidding behavior   | Parameters  |  |
| <i>Conceding</i><br>(Deterministic and predictable)                  | Four opponents with $e = 0.1, 0.2, 1$ , and $2$ in Equation (25)                    |  |
| <i>Random</i><br>(Non-deterministic and unpredictable)               | Four opponents with thresholds of $0, 0.25, 0.5, 0.75$ , and $1$                    |  |
| <i>Conceding with an offset</i><br>(Deterministic and unpredictable) | Three opponents with $P_{\max} = 0.7, 0.8$ , and $0.9$ ( $e = 1$ ) in Equation (25) |  |
| <i>Non-conceding</i><br>(Deterministic and unpredictable)            | Four opponents with $P_{\min} = 0, 0.25, 0.5, 0.75$ , and $1$ in Equation (26)      |  |
|  |   | <i>Itex vs cypress</i><br>Size: Small<br>Opposition: L, M, and H <sup>a</sup><br>Bid Distribution: L, M, and H |
|  |   | <i>Employment</i><br>Size: Small<br>Opposition: L, M, and H<br>Bid Distribution: L, M, and H                   |
|  |   | <i>Car</i><br>Size: Medium<br>Opposition: L, M, and H<br>Bid Distribution: L, M, and H                         |
|  |   | <i>Supermarket</i><br>Size: Large<br>Opposition: L, M, and H<br>Bid Distribution: L, M, and H                  |
|  |   | <i>Travel</i><br>Size: Large<br>Opposition: L, M, and H<br>Bid Distribution: L, M, and H                       |

<sup>a</sup> Note: L, M, and H stand for Low, Medium, and High, respectively.

3. The *Opposition* defines the competitiveness of the domain and is determined by the minimum of distances of all points (i.e., the distance of Kalai–Smorodinsky point, a unique point on Pareto Frontier with equal utilities for both parties) in the outcome space to the point of perfect mutual satisfaction (i.e., maximum utility for both parties).

Now, in order to evaluate the proposed model in a complete setting, all the possible combinations of these three features should be tested. Due to the considerable computational limitations of testing all domains available in the literature and the Genius, Baarslag et al. [38] have carefully selected 5 domains and have considered all levels of those three domain features. The same domains are adopted here for all the aspects affecting the performance of opponent models to be considered. For this purpose, 45 scenarios containing 3 opposition levels and 3 bid distributions levels in 5 domains are applied. The summary of this setting including 15 bidding strategies over 5 negotiation domains applied in interactions among the agents are tabulated in Table 2.

**5.1.1.3. Interactions** This proposed opponent model is trained to perform against all *deterministic* opponent agents in 45 aforementioned scenarios, a total of 495 ( $= 11 \times 45$ ) negotiation sessions. As for random agents, each negotiation session is repeated 5 times to eliminate the randomness in the results and to ensure that the results are reliable. Consequently, a total of 900 ( $= 5 \times 4 \times 45$ ) negotiation sessions are run against 4 types of *random* opponents, making a total of 1395 negotiation sessions to be executed. All negotiation sessions are executed in the round-based setting for an equal number of 5000 rounds, where each round contains exactly two bids: an *offer* sent by the agent and the associated *counter offer* received from the opponent, a total of 6975000 ( $= 1395 \times 5000$ ) training bids which can be fed into each one of the 20 ( $= 8 + 5 + 1 + 1 + 5$ ) opponent models for training purposes.

The interaction among agents in the first experiment, in which the BOA framework is applied (Section 1), is presented in Fig. 5. Agents in Side B include a bidding strategy, no opponent model, and do not follow an acceptance strategy. In each interaction, an opponent model (equipped with an arbitrary bidding strategy and without an acceptance strategy) embedded in an agent in Side A will be trained by competing against opponent agents in Side B in 45 scenarios. Both sides actually have an acceptance module, but that module never accepts. This is because it is accepted for each opponent model in order to be trained by the same amount of training instances or bids (i.e., 6975000), in order to preserve equal conditions.

In each one of the negotiation sessions, the agent in Side B applies one of 15 bidding strategies in Table 2, while the bidding strategy for the agent in Side A is not important; therefore it applies an arbitrary bidding strategy from Table 2. The strategies chosen in this setting are non-adaptive, because it is required that all the models apply the same bid sequences. If the bidding strategy in Side B were to be adaptive, it would change according to the bidding strategies in Side A. When the models apply different training instances or bids, they will not be compared in equal conditions. Since the bidding strategy in Side B (Opponent Side) is non-adaptive, it is not affected by the bidding strategy in Side A (the side which uses those 20 opponent models including POPPONENT variations). Therefore, it is actually not important which bidding strategy is applied

*Pareto bid*: A point is a Pareto Bid, if and only if it is not possible to make any Pareto improvement at that point.

*Pareto improvement*: The movement from one bid in the outcome space to another bid, which at least increases the utility of one agent, without decreasing the utility of the other agent.

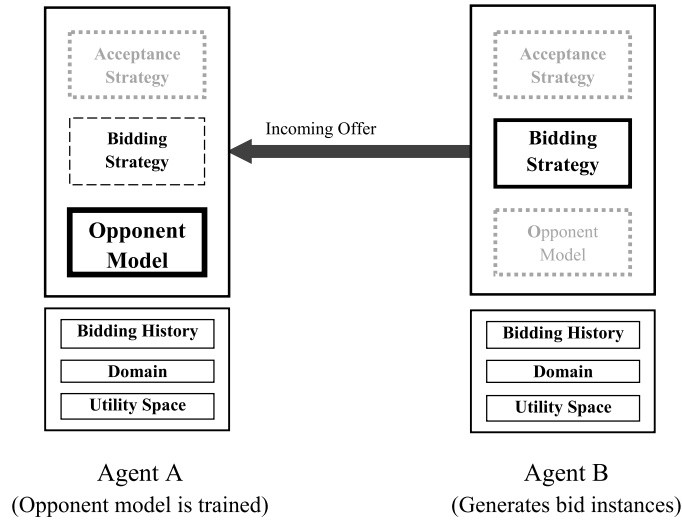


Fig. 5. Opponent models learning process during interactions in Experiment I.

on Side A, because it will not have any effect on the opponent model applied on Side A. This serves the objective of this experiment, which is to train the opponent models in Side A.

#### 5.1.2. Experiment II: evaluating performance of POPPONENT

To evaluate the real world performance of this proposed model in a realistic setting, a similar setting to that applied by the ANAC organizers [38] is designed here. To accomplish this, a BOA framework is applied (Section 1) to embed each opponent model into an agent framework, the bidding strategy of which (together with its associated acceptance strategy) is chosen from the state of the art agents. Next, the average performance of each constructed agent is assessed through different types of components in competition with each other.

**5.1.2.1. Agents** When the settings where each model would be trained are determined, the POPPONENT is compared with the following 6 state of the art *opponent models* together with the *Perfect Model*, *Worst Model*, and *No Model* (see Section 2.3) applied in the Automated Negotiating Agents Competitions (ANAC):

- Agent X Freq. Model
- Agent LG Freq. Model
- CUHK Freq. Model
- Smith Freq. Model
- HardHeaded Freq. Model
- InoxAgent Freq. Model
- Perfect Model
- Worst Model
- No Model

Regarding other components, the top *bidding strategies* together with their associated *acceptance strategies* from ANAC 2010, ANAC 2011, ANAC 2012, and ANAC 2013, and the four time-dependent bidding strategies together with a simple acceptance strategy are collected here. The agents equipped with these strategies are: AgentK [51], Yushu [60], Nozomi [2], HardHeaded [10], Gahbonio [61], IAMHaggler2011 [27], TheNegotiatorReloaded [40], BRAMAgent2 [40], and InoxAgent [62], and the four Time Dependent Conceding Agents from Experiment I (Table 2).

**5.1.2.2. Negotiation scenarios** Each agent competed 5 times, against all opponents on *seven scenarios*: Grocery [17], Thompson Employment [63], Travel [2], Small Energy, Supermarket (from ANAC 2012)<sup>9</sup> [40], Camera [62], and ItexVsCypress [2]. The size, opposition, and bid distribution values (Section 5.1.1.2) of these scenarios are tabulated in the Table 3, sorted by bid distribution.

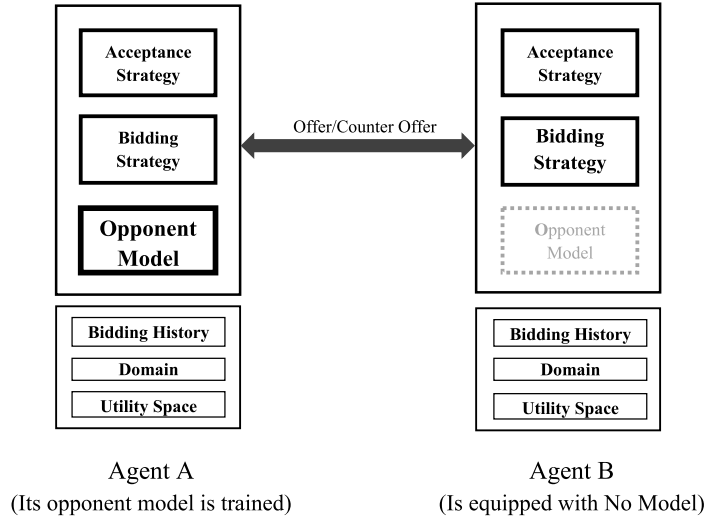
**5.1.2.3. Interactions** This newly proposed model of opponents (AP and P1) together with other 9 opponent models (6 state of the art, Perfect Model, Worst Model and No Model), is evaluated according to the model in Fig. 5 by competing in both

<sup>9</sup> It is worth noting that agents that participated in ANAC 2014 and ANAC 2015 respectively proposed models for nonlinear and multi-party scenarios which fall outside the scope of this paper.



**Table 3**  
Negotiation scenarios applied in Experiment II.

| Domain           | Size   | Issues | Opposition | Bid distribution |
|------------------|--------|--------|------------|------------------|
| Grocery          | 1600   | 5      | 0.806      | 0.191            |
| Energy small     | 15625  | 6      | 0.432      | 0.217            |
| Itex vs. cypress | 180    | 4      | 0.431      | 0.222            |
| Camera           | 3600   | 6      | 0.891      | 0.252            |
| Thompson         | 3125   | 5      | 0.267      | 0.325            |
| Supermarket      | 112896 | 6      | 0.347      | 0.347            |
| Travel           | 188160 | 7      | 0.23       | 0.416            |



**Fig. 6.** Opponent models' learning process during interactions in Experiment II.

sides of A and B (except that in Side A agents have all three BOA modules while agents in Side B have no opponent model). In both sides, the agents apply 13 bidding and acceptance strategies, consisting of 9 agents from ANAC competitions and 4 time-dependent agents. In Side A, there exist 7 opponent models combined with the same 13 bidding and acceptance strategies. Each agent competes 5 times over 7 negotiation scenarios for both preference profiles (i.e., each agent acts in both sides of a scenario); hence, for each model  $11830 (= 7 \times 5 \times 2 \times 13 \times 13)$  sessions are executed. In other words, a total number of  $82810 (= 11830 \times 7)$  sessions are executed in the second experiment to evaluate this newly proposed model together with the state of the art opponent models in a fair realistic competition. All of the sessions are executed in Genius 4.2 [36,37] using the round-based protocol for an equal number of 1000 rounds. The interaction among agents in the second experiment is expressed in Fig. 6, using the BOA framework (Section 1).

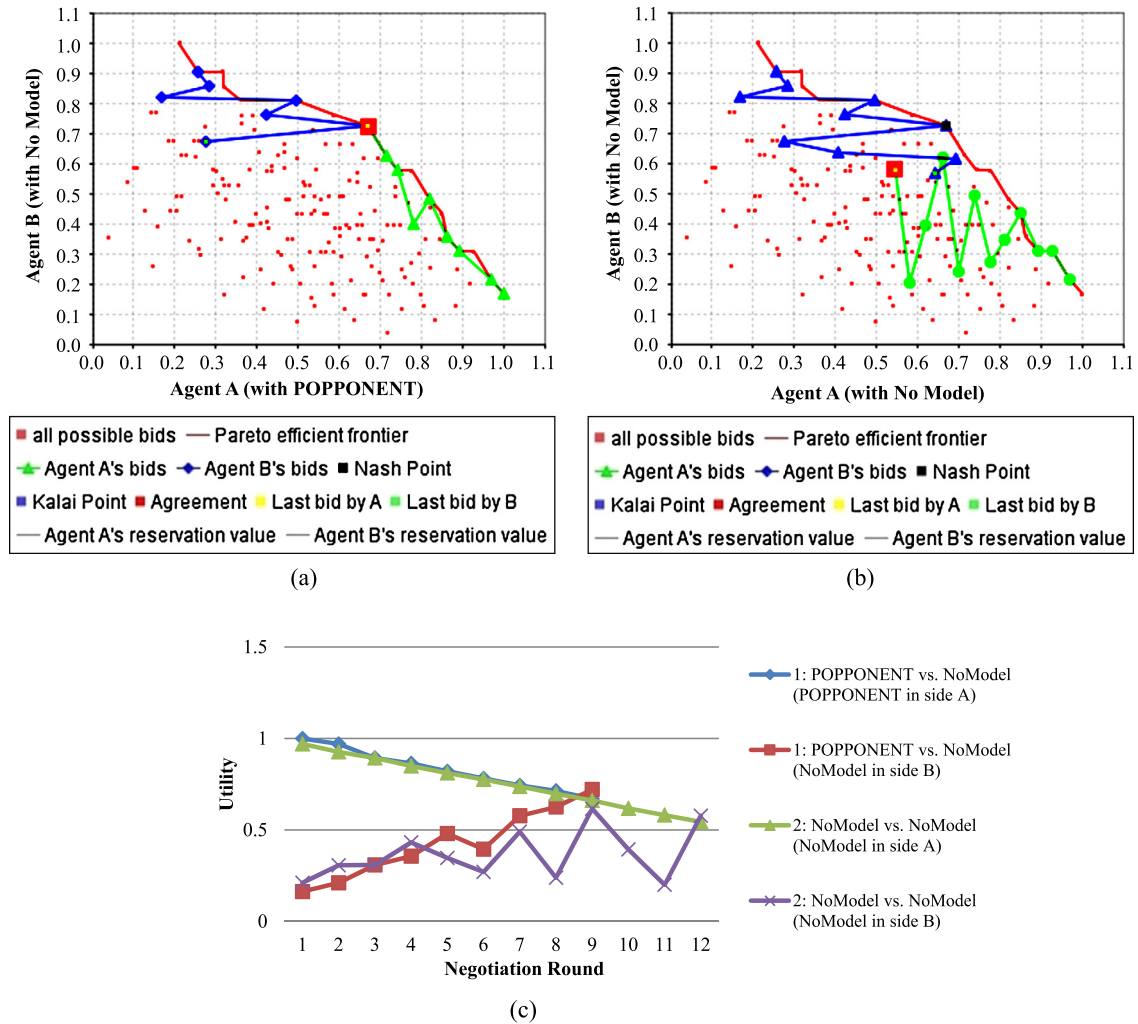
### 5.2. Examples

Before evaluating the overall results of experiment settings I and II, in this section an example of negotiations in course between agents including POPPONENT or No Model is presented. The performance of this proposed POPPONENT model applied by a Linear Time Dependent Agent (Section 5.1.1.1) against another Liner Time Dependent with No Model, Figs. 5 and 6, in 50 rounds (as the negotiation deadline) is presented in Fig. 7. In this figure, (a) especially shows how the agent applying POPPONENT performs in a negotiation session against another agent with no model. How the same agent with No Model performs against another agent of its kind is shown in Fig. 7(b). Finally, the utility of the bids that agent A sends to itself (side A) and to the opponent (side B), 1) when applying POPPONENT, and 2) when applying No Model are shown Fig. 7(c). When the agent A applies POPPONENT, a rapid agreement is achieved with a higher utility for both sides.

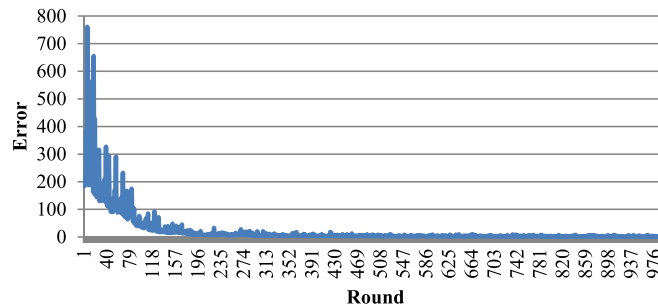
In Fig. 8, we show how the MBGD search method seeks to minimize Standard Error (Equation (5)) by converging towards zero, while this proposed POPPONENT model is being trained against a simple Time-Dependent Agent in a total of 5000 rounds. As the first 1000 rounds indicate, in this proposed model the algorithm moves towards the minimum point in the error surface, Fig. 4.

### 5.3. Experimental results

The detailed experimental results comparing the accuracy and performance of POPPONENT against state of the art models are presented in Sections 5.3.1 and 5.3.2. The major findings are summarized in Section 5.3.3.



**Fig. 7.** Performance gained against a time dependent agent by an agent equipped with (a) POPPONENT, (b) No Model, and (c) Their bid utilities vs. negotiation round.



**Fig. 8.** Standard error vs. negotiation round, for training POPPONENT against a time dependent opponent.

### 5.3.1. Experiment results I: accuracy of POPPONENT

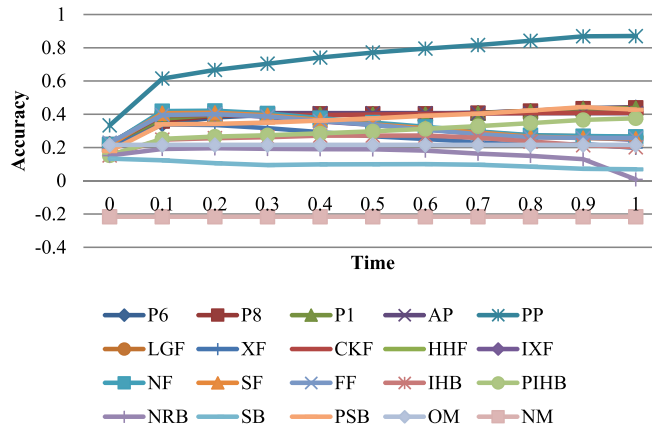
Based on the setting designed for Experiment I (Section 5.1.1), the accuracy of POPPONENT is compared with state of the art opponent models using the *Pearson Correlation* between the estimated and real bid utilities (Section B.1). For each model, the Pearson Correlation is in 11 points in time with equal distances (i.e., 10 time slots). The accuracies of all 20 models

**Table 4**

The list of the opponent models.

| Abbreviation | Opponent model                         | Participated in               | Rank |
|--------------|--|-------------------------------|------|
| P6           | Perceptron Based Model (Constant 0.6)  | –                             | –    |
| P8           | Perceptron Based Model (Constant 0.8)  | –                             | –    |
| P1           | Perceptron Based Model (Constant 1)    | –                             | –    |
| AP           | Adaptive Perceptron Based Model        | –                             | –    |
| PP           | Perfect Perceptron Based Model         | –                             | –    |
| LGF          | AgentLG Frequency Model                | ANAC 2012 (Q, F) <sup>a</sup> | 2F   |
| XF           | AgentX Frequency Model                 | ANAC 2012 (Q)                 | 13Q  |
| CKF          | CUHKAgent Frequency Model              | ANAC 2012 (Q, F)              | 1F   |
| HHF          | HardHeaded Frequency Model             | ANAC 2011 (Q, F)              | 1F   |
| IXF          | InoxAgent Frequency Model              | ANAC 2013 (Q, F)              | 6Q   |
| NF           | Nash Frequency Model                   | –                             | –    |
| SF           | Smith Frequency Model                  | ANAC 2010 (Q, F)              | 7F   |
| FF           | The Fawkes Frequency Model             | ANAC 2013 (Q, F)              | 1F   |
| IHB          | IAMHaggler Bayesian Model              | ANAC 2010 (Q, F)              | 4F   |
| PIHB         | Perfect IAMHaggler Bayesian Model      | –                             | –    |
| NRB          | The Negotiator Reloaded Bayesian Model | ANAC 2012 (Q, F)              | 3F   |
| SB           | Scalable Bayesian Model                | –                             | –    |
| PSB          | Perfect Scalable Bayesian              | –                             | –    |
| OM           | Opposite Model                         | –                             | –    |
| NM           | No Model                               | –                             | –    |
| PM           | Perfect Model                          | –                             | –    |
| WM           | Worst Model                            | –                             | –    |

<sup>a</sup> Note: F and Q respectively stand for the final and qualifying rounds (on ANAC) the agent employing the model participated in.

**Fig. 9.** Average accuracy obtained through all models against all opponent agents.

including 5 variations of POPPONENT (Section 4.2) together with one classic and 13 state of the art models (Section 5.1.1) and No Model<sup>10</sup> are assessed here. The abbreviations of these models are listed in Table 4.

The average accuracy and respective standard deviations of all the opponent models against all opponent agents (predictable and unpredictable opponents) are expressed in Fig. 9, where at the perfect information state (PP), this proposed model outperforms the state of the art models by a large average accuracy. As observed, the other three variations of POPPONENT (i.e., P6, P8, and P1 except PP) outperform all state of the art models with respect to the average accuracy over all opponents. Here, unlike most state of the art models, the accuracy of this model in its four variations increases over time in a monotonic manner. The accuracies of POPPONENT variations and CKF (which have the highest and the closest accuracy compared with POPPONENT variations) are compared in Fig. 10. This figure shows that all POPPONENT variations (except AP) exceed CKF (and all the other models) in terms of average accuracy.

As observed in Fig. 9, most of the state of the art models lose their accuracy over time, since they handle the later bids in an incorrect manner. This is unique to the Bayesian models, since these models make some assumptions on the bidding behavior of the opponent, which become invalid as time passes. Most of the frequency models (and value models) are subject to this phenomenon as well, since they make some assumptions on the opponent's preference profile, which

<sup>10</sup> As also mentioned in footnote 6, there is no need to execute Perfect Model and Worst Model, since their accuracies were already clear given their definition.

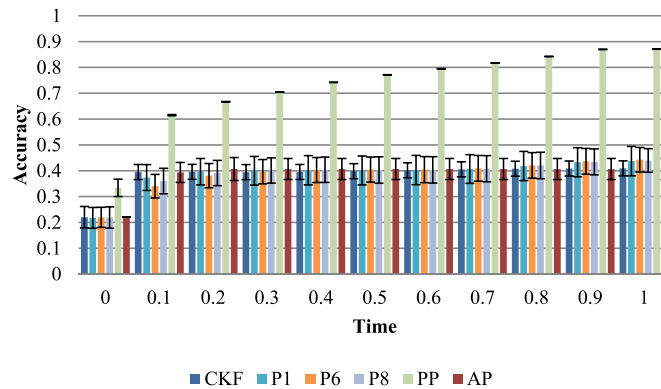


Fig. 10. Average accuracy against all opponent agents obtained through POPPONENT variations and the top state of the art opponent model.

do not necessarily hold true. However, this is not true for the CUHKAgent frequency model, since it only uses 100 initial unique bids received from the opponent in order to update it. The accuracy of this model does not change after a few initial rounds [38].

It is obvious that the POPPONENT in its perfect information state (PP) is not affected by the later bids, since it does not make any assumption on the opponent at all. However, in the other three constant-value variations of POPPONENT, one assumption is made on the opponent's bidding behavior, which is why the accuracy of this proposed model in these three variations is low compared to the PP state. However, even in these three states, the accuracy of POPPONENT increases gradually. This could be attributed to the fact that compared to the other models, this model works based on a lower number of assumptions on the opponent or on the value distribution of the issues of the opponent's preference profile.

At first, it may not look surprising if PP reaches a great overall accuracy, because it uses Perfect Information to estimate the utility values of the bids received from the opponent; in fact it is, for several reasons. First, as observed in Fig. 9, it is evident that the accuracy that the two other Perfect Information models of PSB and PIHB achieve is nothing in comparison with that of the PP. In fact, PP doubles the accuracy values of PSB and PIHB. Second, adding perfect information to these models (PP, PSB, PIHB) would mean that an unsupervised learning problem is converted into a supervised learning problem. Obviously, supervised learning problems are still one of the most challenging problems in data mining and machine learning. Finally, as mentioned in Section 1, in automated negotiations, the agents usually exchange only a limited number of bids before they reach the deadline. However, as observed in this figure, this challenge does not prevent PP from achieving high accuracies. By the time only 10 percent of the negotiation time has passed, the accuracy of PP exceeds 60 percent.

By comparing the accuracy of AP with CKF in Fig. 10, it is revealed that AP outperforms CKF. It is noticed that the accuracy of AP when it is trained using all the bids it receives in a negotiation session reaches its maximum at  $t = 0.2$  (round 1000) and then it begins to decrease. Similar to the approach applied by CKF in preventing a decline in its accuracy, 1000 initial bids are used here to train AP (instead of all 5000 bids).

The accuracy of the models in Experiment I in terms of different features of the domains (Size, Opposition, and Distribution) is presented in Fig. 11, where the following properties are induced:

**Property 1.** POPPONENT model works better in medium to large negotiation domains. □

**Property 2.** Accuracy of POPPONENT model improves with an increase in the distribution level of the negotiation scenario. □

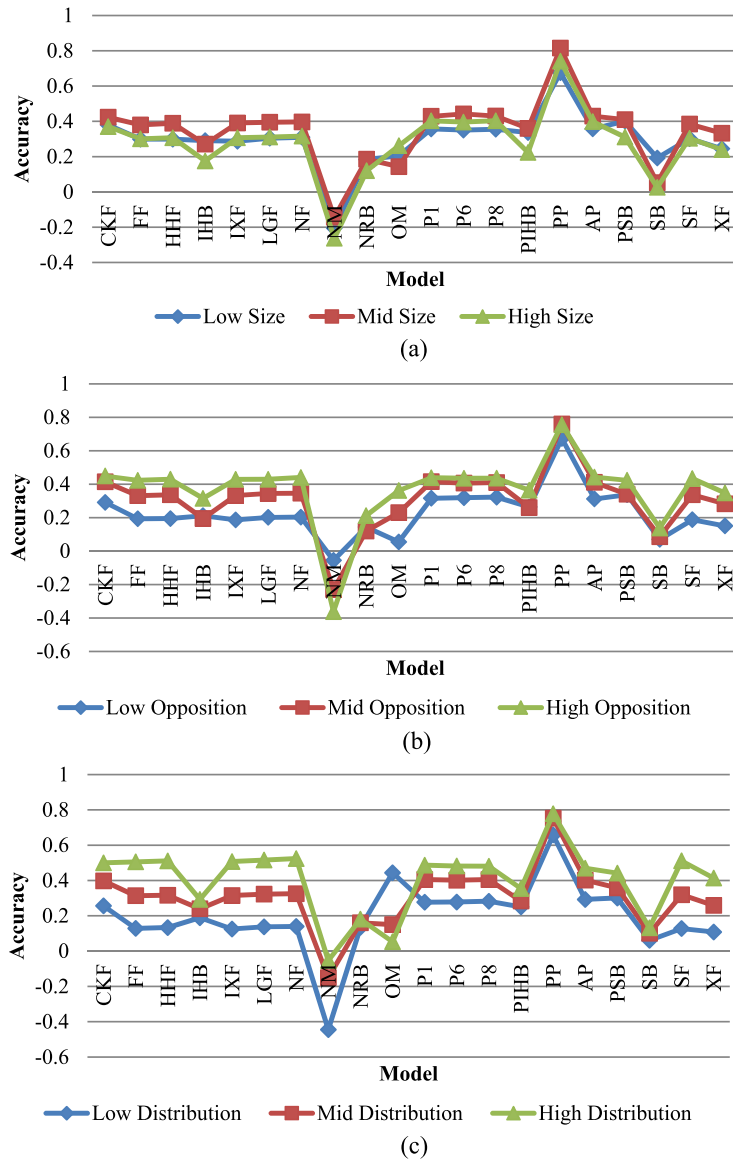
**Property 3.** Most models (including the POPPONENT variations) generally perform better in scenarios with a medium size, high level of opposition, and high level of distribution. □

### 5.3.2. Experiment results II: performance of POPPONENT

The performance of each model is measured and revealed in Experiment II according to the setting explained in Section 5.1.2. Six performance measures of Avg. Utility, Avg. Time of Agreement, Avg. Pareto Distance of Agreement, Avg. Kalai Distance of Agreement, Avg. Nash Distance of Agreement and Avg. Percentage of Pareto Bids are applied for this purpose (see details in Table B.1).

The values of the aforementioned performance measures and the final accuracy (accuracy at the final round of the negotiation) of this proposed POPPONENT model, together with the state of the art models in Experiment II are tabulated in Tables C.1 through C.7 (Appendix C); all the results are presented in Fig. 12.

The content of Tables C.1 through C.7 and Fig. 12 indicate that POPPONENT is the most accurate model compared with other models available. This is consistent with the results of the first experiment, where it is found that the accuracy of



**Fig. 11.** Summary of the results of average accuracy of all models over all opponents in terms of different levels of (a) domain size, (b) domain opposition, and (c) domain distribution.

POPPONENT exceeds state of the art models. By considering the content of [Tables C.1–C.7](#), it is found that POPPONENT (AP) and IXF models are the only two models which can outperform all other models in at least one domain with respect to all performance/accuracy measures; the proposed model's properties consist of:

**Property 4.** In terms of the Pearson Correlation measure, on average, POPPONENT is the most accurate opponent model in linear bilateral multi-issue negotiations against all types of opponents. □

**Property 5.** Among the models in Experiment II, POPPONENT is the most efficient model in at least one domain, with respect to all the measures. □

### 5.3.3. Key findings

The results of Experiment I (Section 5.3.1) are tabulated in [Tables 5 and 6](#). These results show the accuracies of all POPPONENT variations, the nearest opponent models (in accuracy) to the POPPONENT accuracy, and the opponent models applied by the winners of ANAC 2012 and 2013. As observed in these tables, this proposed POPPONENT model in the perfect information state (PP column) reaches higher accuracy levels than the state of the art models, as well as the models



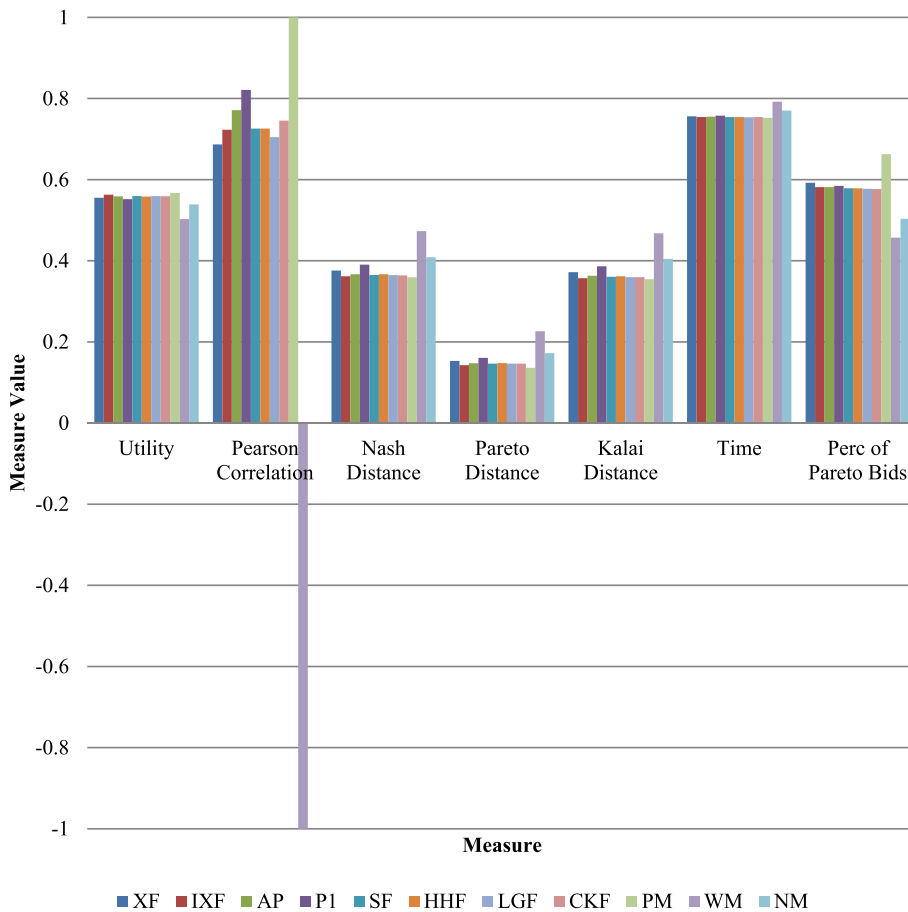


Fig. 12. Summary of the results of Experiment II.

Table 5

POPPONENT's accuracy compared with that of the state of the art and ANAC winners' opponent model against different opponent agents.

| Opponent category                  | Accuracy of POPPONENT |               |               |         |               | State of the art opponent model |               | ANAC winner |         |
|------------------------------------|-----------------------|---------------|---------------|---------|---------------|---------------------------------|---------------|-------------|---------|
|                                    | P6                    | P8            | P1            | AP      | PP            | Model                           | Accuracy      | 2012 CKF    | 2013 FF |
| All opponents                      | 0.3872                | 0.3895        | <b>0.3899</b> | 0.3885  | <b>0.7299</b> | CKF                             | 0.3848        | 0.3848      | 0.3160  |
| Predictable opponents              | 0.7961                | <b>0.8036</b> | <b>0.8077</b> | 0.7325  | 0.7645        | CKF                             | 0.70543       | 0.7054      | 0.4618  |
| Unpredictable opponents            | 0.2384                | 0.2389        | 0.2379        | 0.2634  | <b>0.7173</b> | PSB                             | <b>0.3317</b> | 0.2682      | 0.2630  |
| Random opponents                   | 0.3858                | 0.3665        | 0.3533        | 0.3767  | <b>0.8457</b> | LGF                             | <b>0.5406</b> | 0.4413      | 0.5135  |
| Conceding opponents with an offset | 0.7663                | <b>0.7899</b> | <b>0.8010</b> | 0.7644  | 0.7444        | CKF                             | 0.5991        | 0.5992      | 0.1645  |
| Non-conceding opponents            | −0.3048               | −0.3020       | −0.2997       | −0.2256 | <b>0.5686</b> | OM                              | <b>0.2158</b> | −0.1530     | 0.0864  |

from top ANAC agents. Besides this, as Table 5 shows, this proposed model in the P1 variation (the fourth column) is ranked second (after the PP variation) among all the state of the art models. This includes the models applied by top ANAC 2012 and 2013 agents against all opponents, predictable or conceding opponents, and conceding-opponents-with-an-offset (with accuracies of 0.3899, 0.8077, and 0.8010, respectively). However, the state of the art models of PSB, LGF, and OM with accuracies of 0.3317, 0.5406, and 0.2158 are ranked second against unpredictable opponents, random opponents, and non-conceding opponents, respectively. The opponent model applied by Agent Fawkes (FF) is weak in accuracy, meaning that as the winner of ANAC 2013, this agent must have been applying a very efficient bidding strategy or acceptance strategy or both, to compensate for the very poor accuracy of its opponent model.

That this proposed POPPONENT model in PP variation has the highest accuracy in terms of different levels of scenario features is observed in Table 6. Moreover, when the scenario size is medium and high, this proposed model in P6 and P8 variations, respectively, has ranked second among all the models from the state of the art. Another intriguing observation in Table 6 is the fact that for all levels of scenario features, CUHKAgent Freq. Model has an excellent accuracy compared with

**Table 6**

POPPONENT's accuracy compared with that of the state of the art and ANAC winners' opponent model regarding different scenario features.

| Scenario feature | Feature level | Accuracy of POPPONENT |               |               |               |               | State of the art opponent model |               | ANAC winner |         |
|------------------|---------------|-----------------------|---------------|---------------|---------------|---------------|---------------------------------|---------------|-------------|---------|
|                  |               | P1                    | P6            | P8            | AP            | PP            | Model                           | Accuracy      | 2012 CKF    | 2013 FF |
| Size             | Low           | 0.3573                | 0.3507        | 0.3552        | 0.3575        | <b>0.6757</b> | <b>PSB</b>                      | <b>0.4011</b> | 0.3798      | 0.2990  |
|                  | Mid           | <b>0.4286</b>         | <b>0.4423</b> | <b>0.4301</b> | <b>0.4299</b> | <b>0.8158</b> | <b>CKF</b>                      | 0.4236        | 0.4236      | 0.3801  |
|                  | High          | <b>0.4031</b>         | <b>0.3960</b> | <b>0.4034</b> | <b>0.3988</b> | <b>0.7412</b> | <b>CKF</b>                      | 0.3705        | 0.3705      | 0.3010  |
| Distribution     | Low           | 0.2772                | 0.2781        | 0.2825        | 0.2930        | <b>0.6583</b> | <b>OM</b>                       | <b>0.4442</b> | 0.2567      | 0.1288  |
|                  | Mid           | <b>0.4055</b>         | <b>0.4015</b> | <b>0.4052</b> | <b>0.4025</b> | <b>0.7531</b> | <b>CKF</b>                      | 0.3979        | 0.3979      | 0.3137  |
|                  | High          | 0.4869                | 0.4818        | 0.4808        | 0.4700        | <b>0.7784</b> | <b>NF</b>                       | <b>0.5243</b> | 0.4999      | 0.5056  |
| Opposition       | Low           | 0.3157                | 0.3192        | 0.3224        | 0.3121        | <b>0.6734</b> | <b>PSB</b>                      | <b>0.3376</b> | 0.2912      | 0.1935  |
|                  | Mid           | <b>0.4151</b>         | 0.4067        | 0.4099        | 0.4110        | <b>0.7606</b> | <b>CKF</b>                      | 0.4141        | 0.4141      | 0.3315  |
|                  | High          | 0.4388                | 0.4356        | 0.4361        | 0.4424        | <b>0.7557</b> | <b>CKF</b>                      | <b>0.4492</b> | 0.4492      | 0.4231  |

**Table 7**

Summary of Experiment I results: the most accurate realistic model.

| Against                            | Top model           | Scenario feature | Feature level | Top model           |
|------------------------------------|---------------------|------------------|---------------|---------------------|
| All opponents                      | P1 (and P8, AP, P6) | Size             | Low           | CKF                 |
| Predictable opponents              | P8 (and P1, P6, AP) |                  | Mid           | P6 (and P8, AP, P1) |
|                                    |                     |                  | High          | P8 (and P1, AP, P6) |
| Unpredictable opponents            | NF                  | Distribution     | Low           | OM                  |
| Random opponents                   | LGF                 |                  | Mid           | P1 (and P8, AP, P6) |
|                                    |                     |                  | High          | NF                  |
| Conceding opponents with an offset | P1 (and P8, P6, AP) | Opposition       | Low           | CKF                 |
| Non-conceding opponents            | OM                  |                  | Mid           | P1                  |
|                                    |                     |                  | High          | CKF                 |

**Table 8**

Top performing real model in each domain for all performance/accuracy measures in Experiment II.

| Domain        | Utility              | Pearson correlation        | Nash distance        | Pareto distance     | Kalai distance       | Time                       | Perc. of Pareto bids       |
|---------------|----------------------|----------------------------|----------------------|---------------------|----------------------|----------------------------|----------------------------|
| Supermarket   | IXF (and LGF)        | <b>P1</b> (and <b>AP</b> ) | IXF (and XF)         | XF (and IXF)        | IXF (and XF)         | XF (and <b>AP</b> )        | XF (and <b>P1</b> )        |
| Travel        | IXF (and <b>AP</b> ) | <b>P1</b> (and <b>AP</b> ) | IXF (and HHF)        | IXF (and HHF)       | IXF (and CKF)        | <b>AP</b> (and XF)         | XF (and LGF)               |
| Thompson      | IXF (and SF)         | <b>P1</b> (and <b>AP</b> ) | CKF (and LGF)        | IXF (and SF)        | CKF (and LGF)        | CKF (and LGF)              | <b>P1</b> (and <b>AP</b> ) |
| Grocery       | <b>AP</b> (and IXF)  | IXF (and <b>AP</b> )       | CKF (and IXF)        | <b>AP</b> (and CKF) | CKF (and IXF)        | SF (and LGF)               | LGF (and SF)               |
| Energy        | HHF (and SF)         | <b>P1</b> (and CKF)        | IXF (and <b>AP</b> ) | HHF (and IXF)       | IXF (and <b>AP</b> ) | IXF (and HHF)              | XF (and <b>P1</b> )        |
| Camera        | LGF (and SF)         | <b>P1</b> (and HHF)        | CKF (and <b>AP</b> ) | LGF (and SF)        | CKF (and SF)         | LGF (and HHF)              | LGF (and IXF)              |
| ItexVsCypress | IXF (and LGF)        | <b>P1</b> (and <b>AP</b> ) | <b>AP</b> (and LGF)  | IXF (and LGF)       | <b>AP</b> (and LGF)  | <b>AP</b> (and <b>P1</b> ) | IXF (and XF)               |
| Average       | IXF (and LGF)        | <b>P1</b> (and <b>AP</b> ) | IXF (and CKF)        | IXF (and LGF)       | IXF (and CKF)        | LGF (and SF)               | XF (and P1)                |

the other state of the art models. Since the agent applied here is the winner of ANAC 2012, it could be deduced that part of this success must have been due to the high accuracy of the opponent models' module of this agent.

As tabulated in Table 7, in average the POPPONENT (P1 variation) is the most accurate model vs. all opponents. Moreover, here, it is observed that for five levels of scenario features out of a total of nine, POPPONENT exhibited an excellent accuracy in comparison with all other top models, thus recommending the application of POPPONENT as the most accurate model in scenarios with medium or large size, medium distribution, and low or medium level of opposition.

Both Experiments I and II reveal that POPPONENT is undoubtedly the most accurate model among all its counterparts. Besides this, as expressed in Tables C.1–C.7, POPPONENT achieves the highest performance in the Grocery domain for the average utility measure (through AP), the highest performance in the Travel and ItexVsCypress domains for the average time of agreements measure (through AP), the highest performance in the ItexVsCypress for the Nash and Kalai distance measures (through AP), the highest performance in the Grocery domain for the Pareto distance measure (through AP), the best performance in the Supermarket, Travel, Thompson, Energy, Camera, and ItexVsCypress, for the Pearson Correlation measures (through P1), and the best performing model in the Thompson domain for the percentage of Pareto bids measure (through P1). These results are tabulated in Table 8.

According to Tables C.1–C.7 and Table 8, it is observed that P1 has not performed well in terms of the measures, except for a rather intriguing performance value in the Pearson Correlation and a very high performance value in the Percentage of Pareto Bids. Although it is the most accurate model (high Pearson Correlation value) and is successful in making Pareto Bids (high Percentage of Pareto Bids value), it is not successful in obtaining favorable outcomes. This phenomenon can be observed for XF as well. However, AP (with lower Pearson Correlation and Percentage of Pareto Bids values) has been more successful in getting favorable utilities; thus, it could be concluded that making more Pareto Bids does not necessarily make an opponent

model successful. Rather, it seems that there is an optimal point in the *Percentage of Pareto Bids* measure, which guides the agent to achieve the highest performance.

#### 5.3.4. Summary

In brief, by considering PP, POPPONENT is the most accurate model against predictable, unpredictable, random, conceding opponents with an offset, non-conceding, and all types of opponents, and also in terms of different levels of scenario features (size, opposition, distribution). Excluding PP, POPPONENT variations are the top performing models against all opponents, predictable opponents, and conceding opponents with an offset. POPPONENT variations rank first in the scenarios with medium opposition, medium distribution, and medium to large size. The results of Experiment II reveal that for all the measures, POPPONENT reaches the highest performance in at least one of the domains.

## 6. Conclusions

Negotiation is the science and art of resolving disputes and reaching consensus between human parties. Automated bilateral multi issue negotiations are a special kind of traditional negotiations where two intelligent automated negotiating agents undertake the task of making an agreement on multiple issues on behalf of their human clients. Each automated negotiating agent strategy can be of the following subcomponents [1]: (1) Opponent Modeling, (2) Bidding Strategy, and (3) Acceptance Strategy. Thus, the negotiation strategy could be thought of as a larger component, which embodies the aforementioned subcomponents, and manages an efficient collaboration and interaction between them to accomplish the whole negotiation task. Opponent modeling techniques, as one of the constituent components of a negotiation strategy, are highly contributive in the success of a negotiating agent, in terms of both obtaining better individual utilities and achieving higher social welfare values. Therefore they are widely studied in the literature of Automated Negotiations.

In this article, a new technique is proposed based on perceptron units, called POPPONENT, in order to model the preferences of the opponent in bilateral multi issue negotiations with linear utility functions. In fact, POPPONENT is a very successful implementation of a generalized version of the Standard Gradient Descent search algorithm (GD), referred to as the Multi Bipartite Gradient Descent search (MBGD) in a practical AI problem. The performance of the POPPONENT model is compared with the state of the art models from the Genius repository [36,37]. The POPPONENT is evaluated in 5 separate states which include the *perfect information* state (PP), the *constant value* states (with constant values of 0.6 (P6), 0.8 (P8), 1 (P1)), and the *Adaptive* state (AP). In the perfect information state, it is revealed that this outperforms the state of the art models in average accuracy by a large margin, when the accuracy is measured over all opponents including predictable (conceding opponents), unpredictable, random, conceding opponents with an offset, and non-conceding. It is found in the experiments that the accuracy of POPPONENT (in all the three constant value states) exceeds the accuracy of the most accurate state of the art model against all opponents in average and most other scenarios. Evaluating the performance of POPPONENT through Genius indicates that it overcomes the most accurate state of the art opponent models. The results indicate that POPPONENT overcomes all the state of the art models in at least one domain for all the performance/accuracy measures.

In Experiment I, three constant values (together with perfect information) are applied to estimate the utility of a received bid in the opponent's utility space. In Experiment II, an adaptive method is applied to estimate the bidding behavior of the opponent.

It is believed that there is still a possibility for more improvements in accuracy and performance through adaptive methods. In the future, we plan to work further on the estimation of the opponent bids utilities, to design and implement highly efficient adaptive methods in order to estimate bid utilities through the history of bids which are received from the opponent in an incremental manner, and to evaluate the accuracy of this model while employing such adaptive methods. Another interesting future study direction would be the evaluation of this proposed model through different parameter values (i.e. learning rate, number of repeats, etc.). It is unknown how the accuracy and performance of this model is affected by the changes made to its parameters, and whether the model accuracy and performance could be further improved with proper parameter values. POPPONENT is a successful implementation of this new search method, with MBGD in the area of preference modeling automated negotiations. We are interested in the applicability of this new method to other practical areas of Artificial Intelligence.

## Acknowledgements

The authors would like to warmly thank Dr. Tim Baarslag for providing the necessary scenario files for the experiments. The authors would also like to thank him for his compassion, and his continued support through the whole process of this research.

## Appendix A. Complexity of the POPPONENT algorithm

Complexity analysis of this proposed POPPONENT algorithm is run easily. According to Equations (21) and (22), upon receiving a new bid from the opponent, POPPONENT applies delta rules to update the current estimated preference profile of the opponent. Let  $n$  represent the number of negotiation issues, and  $N$  represent the number of training repeats for each

**Table B.1**

Summary of the measures used in the experiments.

| Experiment I              |  |
|---------------------------|--|
| Pearson correlation       | $d_p(u_{OP}, u'_{OP}) = \frac{\sum_{\vec{w} \in \Omega} (u_{OP}(\vec{w}) - \overline{u_{OP}})(u'_{OP}(\vec{w}) - \overline{u'_{OP}})}{\sqrt{\sum_{\vec{w} \in \Omega} (u_{OP}(\vec{w}) - \overline{u_{OP}})^2 \sum_{\vec{w} \in \Omega} (u'_{OP}(\vec{w}) - \overline{u'_{OP}})^2}}$ |
| Experiment II             |  |
| Avg. Utility              | $AvgUtility_A = \frac{\sum_{i=1}^N utility_A^i}{N}$  |
| Avg. Nash Distance        | $AvgNashDist_{A,B} = \frac{\sum_{i=1}^N NashDist_{A,B}^i}{N}$<br>$NashDist_{A,B}^i = \sqrt{(utility_A^i - NashUtility_A^i)^2 + (utility_B^i - NashUtility_B^i)^2}$   |
| Avg. Kalai Distance       | $AvgKalaiDist_{A,B} = \frac{\sum_{i=1}^N KalaiDist_{A,B}^i}{N}$<br>$KalaiDist_{A,B}^i = \sqrt{(utility_A^i - KalaiUtility_A^i)^2 + (utility_B^i - KalaiUtility_B^i)^2}$  |
| Avg. Pareto Distance      | $AvgParetoDist_{A,B} = \frac{\sum_{i=1}^N ParetoDist_{A,B}^i}{N}$<br>$ParetoDist_{A,B}^i = \sqrt{(utility_A^i - ParetoUtility_A^i)^2 + (utility_B^i - ParetoUtility_B^i)^2}$   |
| Avg. Time                 | $AvgTime = \frac{\sum_{i=1}^N TimeOfAgreement^i}{N}$   |
| Avg. Perc. of Pareto Bids | $AvgPercOfParetoBids = \frac{\sum_{i=1}^N (\frac{TotalParetoBids^i}{TotalBidsOffered^i})}{N}$  |

training example; next, for each training instance, the delta rule in Equation (22) (Algorithm: line 18) repeats  $n$  times, the delta rule in Equation (21) (Algorithm: line 21) repeats  $n$  times, and the whole process is repeated  $N$  times for each training instance (Algorithm: line 15), hence, the total number of executions for the delta rule:

$$EN_{Delta Rule} = N(n + n) = 2Nn \quad (A.1)$$

Accordingly, the computational complexity of the algorithm only depends on the number of negotiation issues, therefore the computational complexity of the algorithm is linear ( $O(n)$ ).

## Appendix B. Measures

This section explains different measures provided by ANAC in the Genius framework [38], applied here to assess the accuracy and performance of POPPONENT in both of the aforementioned experiments. These measures are tabulated in Table B.1 and explained in Sections B.1 and B.2.

### B.1. The measures applied in Experiment I

In the setting adopted in Experiment I (Section 5.1.1), the accuracy of this proposed model is evaluated together with other opponent models through a *Pearson Correlation* between the estimated and real bid utilities [38]:

$$d_p(u_{OP}, u'_{OP}) = \frac{\sum_{\vec{w} \in \Omega} (u_{OP}(\vec{w}) - \overline{u_{OP}})(u'_{OP}(\vec{w}) - \overline{u'_{OP}})}{\sqrt{\sum_{\vec{w} \in \Omega} (u_{OP}(\vec{w}) - \overline{u_{OP}})^2 \sum_{\vec{w} \in \Omega} (u'_{OP}(\vec{w}) - \overline{u'_{OP}})^2}} \quad (B.1)$$

where,  $u_{OP}$  is the real utility space or preference profile of the opponent,  $u'_{OP}$  is the estimated utility space or preference profile of the opponent,  $u_{OP}(\vec{w})$  is the real utility of the negotiation outcome  $\vec{w}$  in the opponent's utility space, and  $u'_{OP}(\vec{w})$  is the estimated utility of the negotiation outcome  $\vec{w}$  in the opponent's utility space.

Since this measure evaluates the extent to which a model can accurately predict the ranking of the bids in outcome space and not their exact utilities, it best fits our purpose. Awareness regarding the rankings of the bids in the outcome space is sufficient to make bids on the Pareto Frontier<sup>11</sup> (or at least as close to Pareto Frontier as possible). It is worth noting that in order to make conceding moves in further approaching the Pareto Frontier, knowing the exact utility values of bids is non-essential. Instead, we only need to know the best bid among a number of bids with equal utilities for the

<sup>11</sup> See footnote 7.

agent (i.e., Iso-curve bids). If the best bid is known, a move can happen in a negotiation round, which would get us as close to the Pareto Frontier as possible.

## B.2. The measures applied in Experiment II

In Experiment II (Section 5.1.2), the performance of this proposed model is evaluated by measuring the real performance of agents applying opponent models in real world experimental negotiation scenarios. For this purpose, the six performance measures [9], which are explained below are applied here:

*Average utility* [12,17,64,65] of the agents in this designed setting is computed through Equation (B.1):

$$AvgUtility_A = \frac{\sum_{i=1}^N utility_A^i}{N} \quad (B.2)$$

where  $N$  is the total number of sessions,  $AvgUtility_A$  is the average utility of agent  $A$  in those  $N$  sessions, and  $utility_A^i$  is the utility of agent  $A$  in the  $i$ th session.

*Average Nash distance of agreements* [65,66] specifies the average distance to the Nash point<sup>12</sup>:

$$AvgNashDist_{A,B} = \frac{\sum_{i=1}^N NashDist_{A,B}^i}{N} \quad (B.3)$$

thus

$$NashDist_{A,B}^i = \sqrt{(utility_A^i - NashUtility_A^i)^2 + (utility_B^i - NashUtility_B^i)^2}$$

where  $N$  is the total number of sessions,  $AvgNashDist_{A,B}$  is the average Nash distance of agreements for agents  $A$  and  $B$  in those  $N$  negotiation sessions,  $NashDist_{A,B}^i$  is the Nash distance of the agreement in the  $i$ th session,  $NashUtility_A^i$  is the utility of the Nash point for agent  $A$  in the  $i$ th session, and  $NashUtility_B^i$  is the utility of the Nash point for agent  $B$  in the  $i$ th session.

The *Average Kalai distance of agreements* [65,66] specifies the average distance to the Kalai point<sup>13</sup>:

$$AvgKalaiDist_{A,B} = \frac{\sum_{i=1}^N KalaiDist_{A,B}^i}{N} \quad (B.4)$$

so that,

$$KalaiDist_{A,B}^i = \sqrt{(utility_A^i - KalaiUtility_A^i)^2 + (utility_B^i - KalaiUtility_B^i)^2}$$

where  $N$  is the total number of sessions,  $AvgKalaiDist_{A,B}$  is the average Kalai distance of agreements for agents  $A$  and  $B$  in those  $N$  negotiation sessions,  $KalaiDist_{A,B}^i$  is the Kalai distance of the agreement in the  $i$ th session,  $KalaiUtility_A^i$  is the utility of the Kalai point for agent  $A$  in the  $i$ th session, and  $KalaiUtility_B^i$  is the utility of the Kalai point for agent  $B$  in the  $i$ th session.

*Average Pareto distance of agreements* [17,65,66] specifies the average minimal distance of agreements from the Pareto Frontier:

$$AvgParetoDist_{A,B} = \frac{\sum_{i=1}^N ParetoDist_{A,B}^i}{N} \quad (B.5)$$

here,

$$ParetoDist_{A,B}^i = \sqrt{(utility_A^i - ParetoUtility_A^i)^2 + (utility_B^i - ParetoUtility_B^i)^2}$$

where  $N$  is the total number of sessions,  $AvgParetoDist_{A,B}$  is the average Pareto distance of agreements for agents  $A$  and  $B$  in the  $N$  negotiation sessions,  $ParetoDist_{A,B}^i$  is the Pareto distance of the agreement in the  $i$ th session,  $ParetoUtility_A^i$  is the utility of the Pareto point for agent  $A$  in the  $i$ th session, and  $ParetoUtility_B^i$  is the utility of the Pareto point for agent  $B$  in the  $i$ th session.

The *Average time of agreement* [1,65] specifies the average amount of time to reach an agreement:

$$AvgTime = \frac{\sum_{i=1}^N TimeOfAgreement^i}{N} \quad (B.6)$$

<sup>12</sup> Nash: a unique point on Pareto Frontier at which the product of the utilities of both parties is maximized [43,56,59].

<sup>13</sup> Kalai–Smorodinsky: the point intersecting Pareto Frontier and the line linking Conflict point (0, 0) to Utopia Point (1, 1) [57,58].



where  $N$  is the total number of sessions,  $AvgTime$  is the average time of agreements in those  $N$  sessions, and  $TimeOfAgreement^i$  is amount of time elapsed before reaching an agreement in the  $i$ -th session.

*Average Percentage of Pareto Bids* [36,62], specifies the average percentage of bids offered in a negotiation, which reside on the Pareto Frontier as Equation (B.7):

$$AvgPercOfParetoBids = \frac{\sum_{i=1}^N \left( \frac{TotalParetoBids^i}{TotalBidsOffered^i} \right)}{N} \quad (B.7)$$

where  $N$  is the total number of sessions,  $TotalParetoBids^i$  and  $TotalBidsOffered^i$  are the total number of *Pareto bids* and the total number of bids that the agent has offered in the  $i$ th session.

The Nash Distance, Kalai Distance, and Pareto Distance measures determine the social efficiency of the model; the lower these numbers, the more socially efficient the model in terms of these performance measures.

## Appendix C. Tabulated Experiment results II

**Table C.1**

POPPONENT (a) *average utility*, (b) *standard deviation* compared with that of the state-of-the art opponent models in different domains.

| Domain        | LGF             | CKF      | SF              | XF       | HHF            | IXF             | AP                 | P1       |
|---------------|-----------------|----------|-----------------|----------|----------------|-----------------|--------------------|----------|
| Supermarket   | <b>0.573509</b> | 0.568959 | 0.569648        | 0.586    | 0.565695       | <b>0.580867</b> | 0.571050296        | 0.569139 |
| Travel        | 0.647501        | 0.649589 | 0.649605        | 0.638958 | 0.651974       | <b>0.656795</b> | <b>0.650498172</b> | 0.63812  |
| Thompson      | 0.538719        | 0.538328 | <b>0.540929</b> | 0.528138 | 0.537564       | <b>0.541321</b> | 0.535137574        | 0.515908 |
| Grocery       | 0.696039        | 0.697289 | 0.697068        | 0.691276 | 0.695758       | <b>0.698418</b> | <b>0.698718439</b> | 0.689838 |
| Energy        | 0.297334        | 0.299239 | <b>0.30033</b>  | 0.294104 | <b>0.30118</b> | 0.300602        | 0.297127466        | 0.295067 |
| Camera        | <b>0.647862</b> | 0.647205 | <b>0.647522</b> | 0.647475 | 0.645714       | 0.644866        | 0.646040119        | 0.642942 |
| ItexVsCypress | <b>0.514222</b> | 0.513911 | 0.509994        | 0.502188 | 0.509697       | <b>0.516857</b> | 0.511943972        | 0.512119 |
| Average       | <b>0.559312</b> | 0.559217 | 0.5593          | 0.555448 | 0.558226       | <b>0.562818</b> | 0.558645148        | 0.551876 |

(a)

| Domain        | LGF         | CKF         | SF       | XF       | HHF      | IXF      | AP          | P1       |
|---------------|-------------|-------------|----------|----------|----------|----------|-------------|----------|
| Supermarket   | 0.021368767 | 0.025083718 | 0.025362 | 0.023389 | 0.026293 | 0.026438 | 0.028204762 | 0.027757 |
| Travel        | 0.011287649 | 0.015570161 | 0.01921  | 0.022077 | 0.016814 | 0.020338 | 0.019192385 | 0.019124 |
| Thompson      | 0.021494188 | 0.018236218 | 0.023743 | 0.023343 | 0.020075 | 0.019065 | 0.023571066 | 0.023923 |
| Grocery       | 0.003272382 | 0.002298787 | 0.003168 | 0.00411  | 0.002325 | 0.002778 | 0.002256075 | 0.006159 |
| Energy        | 0.015129671 | 0.01443495  | 0.014447 | 0.016717 | 0.012761 | 0.013459 | 0.017596407 | 0.014809 |
| Camera        | 0.005067326 | 0.005288428 | 0.004266 | 0.007161 | 0.00569  | 0.005116 | 0.006288319 | 0.00633  |
| ItexVsCypress | 0.021513613 | 0.016542951 | 0.019317 | 0.020531 | 0.017027 | 0.020397 | 0.017874846 | 0.020225 |
| Average       | 0.014161942 | 0.013922174 | 0.015645 | 0.016761 | 0.014427 | 0.01537  | 0.016426266 | 0.016904 |

(b)

**Table C.2**

POPPONENT (a) *average time of agreement*, (b) *standard deviation* compared with that of the state of the art opponent models in different domains.

| Domain        | LGF             | CKF             | SF              | XF              | HHF             | IXF             | AP                 | P1              |
|---------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|--------------------|-----------------|
| Supermarket   | 0.808724        | 0.810161        | 0.809534        | <b>0.804034</b> | 0.80866         | 0.806934        | <b>0.806424936</b> | 0.807854        |
| Travel        | 0.76054         | 0.759906        | 0.759773        | <b>0.758998</b> | 0.75941         | 0.759364        | <b>0.757925506</b> | 0.760031        |
| Thompson      | <b>0.835528</b> | <b>0.835166</b> | 0.83646         | 0.838632        | 0.838402        | 0.837165        | 0.836101768        | 0.839077        |
| Grocery       | <b>0.608107</b> | 0.609099        | <b>0.607777</b> | 0.618218        | 0.611178        | 0.612435        | 0.614978513        | 0.619999        |
| Energy        | 0.773351        | 0.774335        | <b>0.773201</b> | 0.778995        | <b>0.77264</b>  | <b>0.772588</b> | 0.780173672        | 0.781178        |
| Camera        | <b>0.637347</b> | 0.639523        | 0.640586        | 0.640841        | <b>0.639089</b> | 0.641449        | 0.641822083        | 0.646237        |
| ItexVsCypress | 0.850859        | 0.851292        | 0.851457        | 0.852534        | 0.851107        | 0.850214        | <b>0.849201095</b> | <b>0.849606</b> |
| Average       | <b>0.753494</b> | 0.754212        | <b>0.754113</b> | 0.756036        | 0.754355        | 0.754307        | 0.755232511        | 0.757712        |

(a)

| Domain        | LGF      | CKF      | SF       | XF       | HHF      | IXF      | AP       | P1       |
|---------------|----------|----------|----------|----------|----------|----------|----------|----------|
| Supermarket   | 0.007324 | 0.007726 | 0.007778 | 0.006191 | 0.006907 | 0.007042 | 0.006146 | 0.005747 |
| Travel        | 0.005982 | 0.006718 | 0.006245 | 0.007309 | 0.006322 | 0.006693 | 0.00703  | 0.005444 |
| Thompson      | 0.006336 | 0.005719 | 0.005978 | 0.006809 | 0.006068 | 0.006218 | 0.00574  | 0.006233 |
| Grocery       | 0.006204 | 0.00662  | 0.005722 | 0.007825 | 0.006449 | 0.006173 | 0.005165 | 0.006089 |
| Energy        | 0.010744 | 0.009376 | 0.010404 | 0.010118 | 0.009639 | 0.011847 | 0.012291 | 0.00984  |
| Camera        | 0.009788 | 0.008842 | 0.007387 | 0.009765 | 0.00978  | 0.008193 | 0.010582 | 0.009451 |
| ItexVsCypress | 0.005668 | 0.004243 | 0.004235 | 0.004982 | 0.004877 | 0.003727 | 0.004849 | 0.004239 |
| Average       | 0.007435 | 0.007035 | 0.006822 | 0.007571 | 0.007149 | 0.007128 | 0.0074   | 0.00672  |

(b)

**Table C.3**POPPONENT (a) *average Nash Distance*, (b) *standard deviation* compared with that of the state of the art opponent models in different domains.

| Domain        | LGF             | CKF             | SF       | XF              | HHF             | IXF             | AP                 | P1       |
|---------------|-----------------|-----------------|----------|-----------------|-----------------|-----------------|--------------------|----------|
| Supermarket   | 0.4428          | 0.449104        | 0.447391 | <b>0.436957</b> | 0.454414        | <b>0.435619</b> | 0.446968028        | 0.468108 |
| Travel        | 0.39598         | <b>0.389799</b> | 0.390145 | 0.409763        | <b>0.388411</b> | <b>0.381927</b> | 0.396482824        | 0.420687 |
| Thompson      | <b>0.457366</b> | <b>0.456009</b> | 0.457919 | 0.480158        | 0.461135        | 0.457773        | 0.465870384        | 0.515648 |
| Grocery       | 0.195603        | <b>0.194597</b> | 0.195551 | 0.209179        | 0.19901         | <b>0.194792</b> | 0.196830137        | 0.215273 |
| Energy        | 0.427181        | 0.424818        | 0.424903 | 0.431643        | 0.41975         | <b>0.419263</b> | <b>0.424612468</b> | 0.439893 |
| Camera        | 0.195573        | <b>0.193879</b> | 0.194174 | 0.202829        | 0.201405        | 0.203392        | <b>0.200030498</b> | 0.211667 |
| ItexVsCypress | <b>0.437558</b> | 0.439873        | 0.443611 | 0.462262        | 0.442742        | 0.439208        | <b>0.437223755</b> | 0.461518 |
| Average       | 0.36458         | <b>0.364011</b> | 0.364814 | 0.376113        | 0.366695        | <b>0.361711</b> | 0.366859728        | 0.390399 |

(a)

| Domain        | LGF      | CKF      | SF       | XF       | HHF      | IXF      | AP       | P1       |
|---------------|----------|----------|----------|----------|----------|----------|----------|----------|
| Supermarket   | 0.025064 | 0.03004  | 0.030687 | 0.028139 | 0.031151 | 0.032586 | 0.03396  | 0.033023 |
| Travel        | 0.015679 | 0.02139  | 0.024804 | 0.028405 | 0.023234 | 0.026597 | 0.024961 | 0.023043 |
| Thompson      | 0.027038 | 0.02437  | 0.028723 | 0.029527 | 0.024818 | 0.023894 | 0.029372 | 0.03046  |
| Grocery       | 0.004842 | 0.003124 | 0.004646 | 0.006004 | 0.003458 | 0.004193 | 0.003613 | 0.009106 |
| Energy        | 0.030315 | 0.028886 | 0.028073 | 0.036176 | 0.025967 | 0.026234 | 0.035326 | 0.030404 |
| Camera        | 0.006846 | 0.00725  | 0.005527 | 0.009397 | 0.00819  | 0.007034 | 0.008469 | 0.008525 |
| ItexVsCypress | 0.02908  | 0.022034 | 0.025028 | 0.026597 | 0.021975 | 0.0253   | 0.02296  | 0.024737 |
| Average       | 0.019838 | 0.019585 | 0.02107  | 0.023464 | 0.019828 | 0.020834 | 0.022666 | 0.022757 |

(b)

**Table C.4**POPPONENT (a) *average Pareto Distance*, (b) *standard deviation* compared with that of the state of the art opponent models in different domains.

| Domain        | LGF             | CKF             | SF              | XF             | HHF             | IXF             | AP                | P1       |
|---------------|-----------------|-----------------|-----------------|----------------|-----------------|-----------------|-------------------|----------|
| Supermarket   | 0.187234        | 0.196962        | 0.195259        | <b>0.17587</b> | 0.202949        | <b>0.182263</b> | 0.19170905        | 0.199027 |
| Travel        | 0.166809        | 0.161085        | 0.160808        | 0.175792       | <b>0.159009</b> | <b>0.15325</b>  | 0.163144761       | 0.17902  |
| Thompson      | 0.190255        | 0.191763        | <b>0.189758</b> | 0.204219       | 0.190913        | <b>0.189265</b> | 0.196355287       | 0.228592 |
| Grocery       | 0.049098        | <b>0.047864</b> | 0.048642        | 0.05976        | 0.051005        | 0.048415        | <b>0.04782886</b> | 0.063034 |
| Energy        | 0.24504         | 0.23955         | 0.238579        | 0.2482         | <b>0.234917</b> | <b>0.23516</b>  | 0.237892409       | 0.246791 |
| Camera        | <b>0.018599</b> | 0.019341        | <b>0.018732</b> | 0.020988       | 0.02142         | 0.023509        | 0.021636859       | 0.028243 |
| ItexVsCypress | <b>0.168117</b> | 0.169787        | 0.174484        | 0.186501       | 0.174247        | <b>0.166601</b> | 0.172447519       | 0.18098  |
| Average       | <b>0.14645</b>  | 0.146622        | 0.146609        | 0.153047       | 0.14778         | <b>0.142637</b> | 0.147287821       | 0.160812 |

(a)

| Domain        | LGF      | CKF      | SF       | XF       | HHF      | IXF      | AP       | P1       |
|---------------|----------|----------|----------|----------|----------|----------|----------|----------|
| Supermarket   | 0.021219 | 0.026404 | 0.026157 | 0.024601 | 0.028204 | 0.027674 | 0.031429 | 0.030529 |
| Travel        | 0.011541 | 0.016171 | 0.019836 | 0.020106 | 0.016996 | 0.021053 | 0.019662 | 0.01764  |
| Thompson      | 0.024243 | 0.019144 | 0.027377 | 0.025687 | 0.019748 | 0.01879  | 0.024168 | 0.024088 |
| Grocery       | 0.00205  | 0.000395 | 0.00189  | 0.002453 | 0.000474 | 0.000786 | 0.00043  | 0.006147 |
| Energy        | 0.025625 | 0.025089 | 0.024971 | 0.032085 | 0.021664 | 0.020435 | 0.028861 | 0.022982 |
| Camera        | 0.00042  | 0.001511 | 0.000267 | 0.003072 | 0.000998 | 0.001244 | 0.001062 | 0.002088 |
| ItexVsCypress | 0.023175 | 0.016976 | 0.020516 | 0.021104 | 0.017771 | 0.021537 | 0.018315 | 0.020995 |
| Average       | 0.015468 | 0.015099 | 0.017288 | 0.018444 | 0.015122 | 0.015931 | 0.017704 | 0.017781 |

(b)

**Table C.5**POPPONENT (a) *average Kalai Distance*, (b) *standard deviation* compared with that of the state of the art opponent models in different domains.

| Domain        | LGF             | CKF             | SF       | XF              | HHF      | IXF             | AP                 | P1       |
|---------------|-----------------|-----------------|----------|-----------------|----------|-----------------|--------------------|----------|
| Supermarket   | <b>0.415358</b> | 0.422711        | 0.422281 | <b>0.413941</b> | 0.428822 | <b>0.41037</b>  | 0.424757548        | 0.446463 |
| Travel        | 0.39598         | <b>0.389799</b> | 0.390145 | 0.409763        | 0.388411 | <b>0.381927</b> | 0.396482824        | 0.420687 |
| Thompson      | <b>0.457366</b> | <b>0.456009</b> | 0.457919 | 0.480158        | 0.461135 | 0.457773        | 0.465870384        | 0.515648 |
| Grocery       | 0.195603        | <b>0.194597</b> | 0.195551 | 0.209179        | 0.19901  | <b>0.194792</b> | 0.196830137        | 0.215273 |
| Energy        | 0.426142        | 0.423733        | 0.423705 | 0.43042         | 0.418411 | <b>0.418065</b> | <b>0.42309111</b>  | 0.438433 |
| Camera        | 0.189814        | <b>0.189249</b> | 0.18979  | 0.1974          | 0.194916 | 0.196988        | 0.196060175        | 0.205757 |
| ItexVsCypress | <b>0.437558</b> | 0.439873        | 0.443611 | 0.462262        | 0.442742 | 0.439208        | <b>0.437223755</b> | 0.461518 |
| Average       | 0.359689        | <b>0.359425</b> | 0.360429 | 0.371875        | 0.361921 | <b>0.357018</b> | 0.362902276        | 0.446463 |

(a)

**Table C.5** (continued)

| Domain        | LGF      | CKF      | SF       | XF       | HHF      | IXF      | AP       | P1       |
|---------------|----------|----------|----------|----------|----------|----------|----------|----------|
| Supermarket   | 0.025549 | 0.029285 | 0.030156 | 0.027401 | 0.030556 | 0.032266 | 0.033389 | 0.032131 |
| Travel        | 0.015679 | 0.02139  | 0.024804 | 0.028405 | 0.023234 | 0.026597 | 0.024961 | 0.023043 |
| Thompson      | 0.027038 | 0.02437  | 0.028723 | 0.029527 | 0.024818 | 0.023894 | 0.029372 | 0.03046  |
| Grocery       | 0.004842 | 0.003124 | 0.004646 | 0.006004 | 0.003458 | 0.004193 | 0.003613 | 0.009106 |
| Energy        | 0.030146 | 0.028889 | 0.028021 | 0.036101 | 0.025865 | 0.026084 | 0.035141 | 0.030398 |
| Camera        | 0.00536  | 0.00594  | 0.00425  | 0.007497 | 0.006767 | 0.006377 | 0.006808 | 0.007048 |
| ItexVsCypress | 0.02908  | 0.022034 | 0.025028 | 0.026597 | 0.021975 | 0.0253   | 0.02296  | 0.024737 |
| Average       | 0.019671 | 0.01929  | 0.020804 | 0.023076 | 0.019525 | 0.020673 | 0.022321 | 0.032131 |

(b)

**Table C.6**

POPPONENT (a) average Pearson Correlation of Bids, (b) standard deviation compared with that of the state of the art opponent models in different domains.

| Domain        | LGF      | CKF             | SF       | XF       | HHF             | IXF             | AP                 | P1              |
|---------------|----------|-----------------|----------|----------|-----------------|-----------------|--------------------|-----------------|
| Supermarket   | 0.708262 | 0.758502        | 0.741386 | 0.72082  | 0.732425        | 0.730347        | <b>0.821774577</b> | <b>0.853468</b> |
| Travel        | 0.586256 | 0.608388        | 0.623808 | 0.573111 | 0.621801        | 0.613423        | <b>0.684165935</b> | <b>0.699424</b> |
| Thompson      | 0.57681  | 0.677093        | 0.604107 | 0.63678  | 0.607754        | 0.605026        | <b>0.680735977</b> | <b>0.765818</b> |
| Grocery       | 0.799005 | 0.824615        | 0.825272 | 0.796731 | 0.837442        | <b>0.849078</b> | <b>0.847187627</b> | 0.842055        |
| Energy        | 0.691501 | <b>0.735425</b> | 0.710331 | 0.67537  | 0.700408        | 0.695129        | 0.680266747        | <b>0.852173</b> |
| Camera        | 0.801799 | 0.805355        | 0.807166 | 0.644701 | <b>0.822057</b> | 0.816216        | 0.821461418        | <b>0.842427</b> |
| ItexVsCypress | 0.7685   | 0.807852        | 0.768802 | 0.760716 | 0.760233        | 0.75148         | <b>0.862496743</b> | <b>0.892469</b> |
| Average       | 0.70459  | 0.745319        | 0.725839 | 0.68689  | 0.726017        | 0.722957        | <b>0.771155575</b> | <b>0.821119</b> |

(a)

| Domain        | LGF      | CKF      | SF       | XF       | HHF      | IXF      | AP          | P1       |
|---------------|----------|----------|----------|----------|----------|----------|-------------|----------|
| Supermarket   | 0.003249 | 0.002386 | 0.003046 | 0.013425 | 0.008649 | 0.011086 | 0.012384432 | 0.005236 |
| Travel        | 0.003141 | 0.002953 | 0.003104 | 0.014834 | 0.009044 | 0.011857 | 0.00896404  | 0.005079 |
| Thompson      | 0.00241  | 0.002172 | 0.002695 | 0.01101  | 0.008288 | 0.009024 | 0.020941602 | 0.004233 |
| Grocery       | 0.001468 | 0.001604 | 0.001485 | 0.0048   | 0.003652 | 0.005377 | 0.002967473 | 0.003301 |
| Energy        | 0.00796  | 0.003655 | 0.00666  | 0.015097 | 0.009944 | 0.014162 | 0.030966949 | 0.004915 |
| Camera        | 0.002729 | 0.002056 | 0.001914 | 0.009481 | 0.008239 | 0.009759 | 0.005869184 | 0.003511 |
| ItexVsCypress | 0.002219 | 0.001554 | 0.001773 | 0.008429 | 0.007647 | 0.012303 | 0.012836845 | 0.003718 |
| Average       | 0.003311 | 0.00234  | 0.002954 | 0.011011 | 0.007923 | 0.01051  | 0.013561504 | 0.004285 |

(b)

**Table C.7**

POPPONENT (a) average Percentage of Pareto Bids, (b) standard deviation compared with that of the state of the art opponent models in different domains.

| Domain        | LGF             | CKF      | SF              | XF              | HHF      | IXF             | AP              | P1              |
|---------------|-----------------|----------|-----------------|-----------------|----------|-----------------|-----------------|-----------------|
| Supermarket   | 0.394825        | 0.395804 | 0.398458        | <b>0.462466</b> | 0.400153 | 0.406819        | 0.415159        | <b>0.417763</b> |
| Travel        | <b>0.373331</b> | 0.372682 | 0.372852        | <b>0.376019</b> | 0.371945 | 0.371299        | 0.370899        | 0.372523        |
| Thompson      | 0.644955        | 0.650202 | 0.649068        | 0.647309        | 0.646974 | 0.648322        | <b>0.661895</b> | <b>0.664335</b> |
| Grocery       | <b>0.698418</b> | 0.696347 | <b>0.696748</b> | 0.689248        | 0.687082 | 0.689335        | 0.695065        | 0.695124        |
| Energy        | 0.378452        | 0.375572 | 0.386453        | <b>0.440211</b> | 0.394558 | 0.396635        | 0.390569        | <b>0.402541</b> |
| Camera        | <b>0.738829</b> | 0.733052 | 0.733789        | 0.7123          | 0.735578 | <b>0.738423</b> | 0.72382         | 0.726447        |
| ItexVsCypress | 0.812633        | 0.813282 | 0.813051        | <b>0.818836</b> | 0.813848 | <b>0.819694</b> | 0.81304         | 0.814807        |
| Average       | 0.577349        | 0.576706 | 0.578631        | <b>0.592341</b> | 0.578591 | 0.581504        | 0.581492        | <b>0.584791</b> |

(a)

| Domain        | LGF      | CKF      | SF       | XF       | HHF      | IXF      | AP       | P1       |
|---------------|----------|----------|----------|----------|----------|----------|----------|----------|
| Supermarket   | 0.021432 | 0.023384 | 0.021473 | 0.027782 | 0.023061 | 0.024289 | 0.021618 | 0.024082 |
| Travel        | 0.017361 | 0.022211 | 0.019032 | 0.023011 | 0.021339 | 0.020696 | 0.022112 | 0.022496 |
| Thompson      | 0.014005 | 0.011695 | 0.011886 | 0.013829 | 0.012629 | 0.012791 | 0.013503 | 0.013269 |
| Grocery       | 0.008898 | 0.010165 | 0.008992 | 0.010919 | 0.010622 | 0.01138  | 0.007556 | 0.007954 |
| Energy        | 0.026258 | 0.021565 | 0.024716 | 0.027132 | 0.022433 | 0.021497 | 0.020732 | 0.023611 |
| Camera        | 0.010874 | 0.011328 | 0.010319 | 0.011261 | 0.012254 | 0.012613 | 0.015186 | 0.012913 |
| ItexVsCypress | 0.007182 | 0.005427 | 0.005512 | 0.007092 | 0.007257 | 0.004341 | 0.006929 | 0.005512 |
| Average       | 0.015144 | 0.015111 | 0.014561 | 0.017289 | 0.015656 | 0.015373 | 0.015376 | 0.015691 |

(b)

## References

- [1] T. Baarslag, K. Hindriks, M. Hendriks, A. Dirkzwager, C. Jonker, Decoupling negotiating agents to explore the space of negotiation strategies, in: *Proceedings of the 5th International Workshop on Agent-Based Complex Automated Negotiations, ACAN*, 2012.
- [2] T. Baarslag, K. Hindriks, C. Jonker, S. Kraus, R. Lin, The first automated negotiating agents competition (ANAC 2010), in: T. Ito, M. Zhang, V. Robu, S. Fatima, T. Matsuo (Eds.), *New Trends in Agent-Based Complex Automated Negotiations*, vol. 383, Springer, Berlin, Heidelberg, 2012, pp. 113–135.
- [3] K. Fujita, T. Ito, T. Baarslag, K. Hindriks, C. Jonker, S. Kraus, et al., The second automated negotiating agents competition (ANAC2011), in: T. Ito, M. Zhang, V. Robu, T. Matsuo (Eds.), *Complex Automated Negotiations: Theories, Models, and Software Competitions*, vol. 435, Springer, Berlin, Heidelberg, 2013, pp. 183–197.
- [4] T. Baarslag, K. Hindriks, C. Jonker, Acceptance conditions in automated negotiation, in: T. Ito, M. Zhang, V. Robu, T. Matsuo (Eds.), *Complex Automated Negotiations: Theories, Models, and Software Competitions*, vol. 435, Springer, Berlin, Heidelberg, 2013, pp. 95–111.
- [5] T. Baarslag, K. Hindriks, C. Jonker, Effective acceptance conditions in real-time automated negotiation, *Decis. Support Syst.* 60 (2014) 68–77.
- [6] T. Baarslag, K.V. Hindriks, Accepting optimally in automated negotiation with incomplete information, presented at the in: *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems*, St. Paul, MN, USA, 2013.
- [7] T. Baarslag, M.C. Hendriks, K. Hindriks, C. Jonker, Learning about the opponent in automated bilateral negotiation: a comprehensive survey of opponent modeling techniques, *Auton. Agents Multi-Agent Syst.* (2015) 1–50, 2015/09/07.
- [8] S.Z. Li, A.K. Jain, *Encyclopedia of Biometrics: I–Z*, vol. 2, Springer, 2009.
- [9] T. Baarslag, M. Hendriks, K. Hindriks, C. Jonker, Measuring the performance of online opponent models in automated bilateral negotiation, in: M. Thielscher, D. Zhang (Eds.), *AI 2012: Advances in Artificial Intelligence*, vol. 7691, Springer, Berlin, Heidelberg, 2012, pp. 1–14.
- [10] T. Krimpen, D. Looije, S. Hajizadeh, HardHeaded, in: T. Ito, M. Zhang, V. Robu, T. Matsuo (Eds.), *Complex Automated Negotiations: Theories, Models, and Software Competitions*, vol. 435, Springer, Berlin, Heidelberg, 2013, pp. 223–227.
- [11] N. Galen Last, Agent Smith: opponent model estimation in bilateral multi-issue negotiation, in: T. Ito, M. Zhang, V. Robu, S. Fatima, T. Matsuo (Eds.), *New Trends in Agent-Based Complex Automated Negotiations*, vol. 383, Springer, Berlin, Heidelberg, 2012, pp. 167–174.
- [12] K. Hindriks, D. Tykhonov, Opponent modelling in automated multi-issue negotiation using Bayesian learning, presented at the in: *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, Estoril, Portugal, 2008.
- [13] A. Frieder, G. Miller, Value model agent: a novel preference profiler for negotiation with agents, in: T. Ito, M. Zhang, V. Robu, T. Matsuo (Eds.), *Complex Automated Negotiations: Theories, Models, and Software Competitions*, vol. 435, Springer, Berlin, Heidelberg, 2013, pp. 199–203.
- [14] H. Jazayeriy, M. Azmi-Murad, N. Sulaiman, N. Izura Udizir, The learning of an opponent's approximate preferences in bilateral automated negotiation, *J. Theor. Appl. Electron. Commer. Res.* 6 (2011) 65–84.
- [15] J. Zhang, F. Ren, M. Zhang, Prediction of the opponent's preference in bilateral multi-issue negotiation through Bayesian learning, in: *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems*, Paris, France, 2014, pp. 331–338.
- [16] C. Williams, V. Robu, E. Gerding, N. Jennings, IAMhaggler: a negotiation agent for complex environments, in: T. Ito, M. Zhang, V. Robu, S. Fatima, T. Matsuo (Eds.), *New Trends in Agent-Based Complex Automated Negotiations*, vol. 383, Springer, Berlin, Heidelberg, 2012, pp. 151–158.
- [17] T. Baarslag, K. Fujita, E.H. Gerding, K. Hindriks, T. Ito, N.R. Jennings, et al., Evaluating practical negotiating agents: results and analysis of the 2011 international competition, *Artif. Intell.* 198 (2013) 73–103.
- [18] K. Sycara, D. Zeng, Benefits of learning in negotiation, in: *Proceedings of the 14th National Conference on Artificial Intelligence and 9th Innovative Applications of Artificial Intelligence Conference, AAAI-97/IAAI-97*, Menlo Park, CA, USA, 1997, pp. 36–42.
- [19] J. Gwak, 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*, Hong Kong, 2011, pp. 163–168.
- [20] C. Yu, F. Ren, M. Zhang, An adaptive bilateral negotiation model based on Bayesian learning, in: T. Ito, M. Zhang, V. Robu, T. Matsuo (Eds.), *Complex Automated Negotiations: Theories, Models, and Software Competitions*, vol. 435, Springer, Berlin, Heidelberg, 2013, pp. 75–93.
- [21] P. Faratin, C. Sierra, N.R. Jennings, Using similarity criteria to make issue trade-offs in automated negotiations, *Artif. Intell.* 142 (2002) 205–237.
- [22] C. Niemann, F. Lang, Assess your opponent: a Bayesian process for preference observation in multi-attribute negotiations, in: T. Ito, M. Zhang, V. Robu, S. Fatima, T. Matsuo (Eds.), *Advances in Agent-Based Complex Automated Negotiations*, vol. 233, Springer, Berlin, Heidelberg, 2009, pp. 119–137.
- [23] R. Lin, S. Kraus, J. Wilkenfeld, J. Barry, Negotiating with bounded rational agents in environments with incomplete information using an automated agent, *Artif. Intell.* 172 (2008) 823–851.
- [24] T. Baarslag, K. Hindriks, C. Jonker, A Tit for Tat negotiation strategy for real-time bilateral negotiations, in: T. Ito, M. Zhang, V. Robu, T. Matsuo (Eds.), *Complex Automated Negotiations: Theories, Models, and Software Competitions*, vol. 435, Springer, Berlin, Heidelberg, 2013, pp. 229–233.
- [25] R. Lin, S. Kraus, J. Wilkenfeld, J. Barry, An automated agent for bilateral negotiation with bounded rational agents with incomplete information, *Front. Artif. Intell. Appl.* 141 (2006) 270–274.
- [26] L. Šerban, G. Silaghi, C. Litan, AgentFSEGA: time constrained reasoning model for bilateral multi-issue negotiations, in: T. Ito, M. Zhang, V. Robu, S. Fatima, T. Matsuo (Eds.), *New Trends in Agent-Based Complex Automated Negotiations*, vol. 383, Springer, Berlin, Heidelberg, 2012, pp. 159–165.
- [27] C. Williams, V. Robu, E. Gerding, N. Jennings, IAMhaggler2011: a Gaussian process regression based negotiation agent, in: T. Ito, M. Zhang, V. Robu, T. Matsuo (Eds.), *Complex Automated Negotiations: Theories, Models, and Software Competitions*, vol. 435, Springer, Berlin, Heidelberg, 2013, pp. 209–212.
- [28] R. Carboneau, G.E. Kersten, R. Vahidov, Predicting opponent's moves in electronic negotiations using neural networks, *Expert Syst. Appl.* 34 (2008) 1266–1273.
- [29] C.R. Williams, V. Robu, E.H. Gerding, N.R. Jennings, Using Gaussian processes to optimise concession in complex negotiations against unknown opponents, presented at the in: *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, Barcelona, Catalonia, Spain, 2011.
- [30] R.Y.K. Lau, Y. Li, D. Song, R.C.W. Kwok, Knowledge discovery for adaptive negotiation agents in e-marketplaces, *Decis. Support Syst.* 45 (2008) 310–323.
- [31] R. Aydoğan, P. Yolum, Learning opponent's preferences for effective negotiation: an approach based on concept learning, *Auton. Agents Multi-Agent Syst.* 24 (2012) 104–140.
- [32] J.R. Anderson, R.S. Michalski, R.S. Michalski, J.G. Carbonell, T.M. Mitchell, *Machine Learning: An Artificial Intelligence Approach*, vol. 2, Morgan Kaufmann, 1986.
- [33] P. Faratin, C. Sierra, N.R. Jennings, Negotiation decision functions for autonomous agents, *Robot. Auton. Syst.* 24 (1998) 159–182.
- [34] T.M. Mitchell, *Machine Learning*, vol. 45, McGraw Hill, Ridge, IL, 1997.
- [35] S.S. Fatima, M. Wooldridge, N. Jennings, Optimal negotiation strategies for agents with incomplete information, in: J.-J. Meyer, M. Tambe (Eds.), *Intelligent Agents VIII*, vol. 2333, Springer, Berlin, Heidelberg, 2002, pp. 377–392.
- [36] R. Lin, S. Kraus, T. Baarslag, D. Tykhonov, K. Hindriks, C.M. Jonker, Genius: an integrated environment for supporting the design of generic automated negotiators, *Comput. Intell.* 30 (2012) 48–70.
- [37] K. Hindriks, C.M. Jonker, S. Kraus, R. Lin, D. Tykhonov, Genius: negotiation environment for heterogeneous agents, presented at the in: *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, Budapest, Hungary, 2009.
- [38] T. Baarslag, M. Hendriks, K. Hindriks, C. Jonker, Predicting the performance of opponent models in automated negotiation, in: *Proceedings of the 7th International Conference on Advanced Information Technologies*, St. Paul, MN, USA, 2013, pp. 59–66.

- [39] J. Hao, H.-f. Leung, CUHKAgent: an adaptive negotiation strategy for bilateral negotiations over multiple items, in: I. Marsa-Maestre, M.A. Lopez-Carmona, T. Ito, M. Zhang, Q. Bai, K. Fujita (Eds.), *Novel Insights in Agent-Based Complex Automated Negotiation*, vol. 535, Springer, Japan, 2014, pp. 171–179.
- [40] C. Williams, V. Robu, E. Gerding, N. Jennings, An overview of the results and insights from the third automated negotiating agents competition (ANAC2012), in: I. Marsa-Maestre, M.A. Lopez-Carmona, T. Ito, M. Zhang, Q. Bai, K. Fujita (Eds.), *Novel Insights in Agent-Based Complex Automated Negotiation*, vol. 535, Springer, Japan, 2014, pp. 151–162.
- [41] F. Zafari, F. Nassiri-Mofakham, BraveCat: iterative deepening distance-based opponent modeling and hybrid bidding in nonlinear ultra large bilateral multi issue negotiation domains, in: N. Fukuta, T. Ito, M. Zhang, K. Fujita, V. Robu (Eds.), *Recent Advances in Agent-Based Complex Automated Negotiation*, vol. 638, Springer International Publishing, Switzerland, 2016, pp. 285–293.
- [42] A. Rubinstein, Perfect equilibrium in a bargaining model, *Econometrica: J. Econom. Soc.* 50 (1982) 97–109.
- [43] G. Weiss, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press, 1999.
- [44] H. Raiffa, *The Art and Science of Negotiation*, Harvard University Press, 1982.
- [45] F. Nassiri-Mofakham, M.A. Nematbakhsh, N. Ghasem-Aghaee, A. Baraani-Dastjerdi, A heuristic personality-based bilateral multi-issue bargaining model in electronic commerce, *Int. J. Hum.-Comput. Stud.* 67 (2009) 1–35.
- [46] F. Nassiri-Mofakham, N. Ghasem-Aghaee, M.A. Nematbakhsh, A. Baraani-Dastjerdi, A personality-based simulation of bargaining in e-commerce, *Simul. Gaming* 39 (2008) 83–100.
- [47] F. Nassiri-Mofakham, M.A. Nematbakhsh, A. Baraani-Dastjerdi, N. Ghasem-Aghaee, R. Kowalczyk, Bidding strategy for agents in multi-attribute combinatorial double auction, *Expert Syst. Appl.* 42 (2015) 3268–3295.
- [48] E. Alpaydin, *Introduction to Machine Learning*, MIT Press, 2004.
- [49] D.T. Larose, *Discovering Knowledge in Data: An Introduction to Data Mining*, John Wiley & Sons, 2014.
- [50] I.H. Witten, E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, 2005.
- [51] S. Kawaguchi, K. Fujita, T. Ito, AgentK: compromising strategy based on estimated maximum utility for automated negotiating agents, in: T. Ito, M. Zhang, V. Robu, S. Fatima, T. Matsuo (Eds.), *New Trends in Agent-Based Complex Automated Negotiations*, vol. 383, Springer, Berlin, Heidelberg, 2012, pp. 137–144.
- [52] S. Kawaguchi, K. Fujita, T. Ito, AgentK2: compromising strategy based on estimated maximum utility for automated negotiating agents, in: T. Ito, M. Zhang, V. Robu, T. Matsuo (Eds.), *Complex Automated Negotiations: Theories, Models, and Software Competitions*, vol. 435, Springer, Berlin, Heidelberg, 2013, pp. 235–241.
- [53] S. Kawaguchi, K. Fujita, T. Ito, Compromising strategy based on estimated maximum utility for automated negotiation agents competition (ANAC-10), in: K. Mehrotra, C. Mohan, J. Oh, P. Varshney, M. Ali (Eds.), *Modern Approaches in Applied Intelligence*, vol. 6704, Springer, Berlin, Heidelberg, 2011, pp. 501–510.
- [54] M. Ikrashi, K. Fujita, Compromising strategy using weighted counting in multi-times negotiations, in: 2014 IIAI 3rd International Conference on Advanced Applied Informatics, IIAIAI, 2014, pp. 453–458.
- [55] K. Fujita, Efficient strategy adaptation for complex multi-times bilateral negotiations, in: 2014 IEEE 7th International Conference on Service-Oriented Computing and Applications, SOCA, 2014, pp. 207–214.
- [56] M. Fasli, *Agent Technology for E-Commerce*, Wiley Press, Chichester, 2007.
- [57] H. Gimpel, J. Mäkiö, Towards multi-attribute double auctions for financial markets, *EM* 16 (2006) 130–139.
- [58] E. Kalai, M. Smorodinsky, Other solutions to Nash's bargaining problem, *Econometrica: J. Econom. Soc.* (1975) 513–518.
- [59] J.F. Nash Jr., The bargaining problem, *Econometrica: J. Econom. Soc.* 18 (1950) 155–162.
- [60] B. An, V. Lesser, Yushu: a heuristic-based agent for automated negotiating competition, in: T. Ito, M. Zhang, V. Robu, S. Fatima, T. Matsuo (Eds.), *New Trends in Agent-Based Complex Automated Negotiations*, vol. 383, Springer, Berlin, Heidelberg, 2012, pp. 145–149.
- [61] M. Ben Adar, N. Sofy, A. Elimelech, Gahboninho: strategy for balancing pressure and compromise in automated negotiation, in: T. Ito, M. Zhang, V. Robu, T. Matsuo (Eds.), *Complex Automated Negotiations: Theories, Models, and Software Competitions*, vol. 435, Springer, Berlin, Heidelberg, 2013, pp. 205–208.
- [62] K. Gal, L. Ilany, The fourth automated negotiation competition, in: K. Fujita, T. Ito, M. Zhang, V. Robu (Eds.), *Next Frontier in Agent-Based Complex Automated Negotiation*, vol. 596, Springer, Japan, 2015, pp. 129–136.
- [63] L. Thompson, *The Heart and Mind of the Negotiator*, 3rd ed., Prentice Hall, Upper Saddle River, New Jersey, 2000.
- [64] T. Klos, K. Somefun, H. La Poutré, Automated interactive sales processes, *IEEE Intell. Syst.* 26 (2011) 54–61.
- [65] F. Zafari, F. Nassiri-Mofakham, A.Z. Hamadani, DOPPONENT: a socially efficient preference model of opponent in bilateral multi issue negotiations, *J. Comput. Secur.* 1 (4) (2015) 283–292.
- [66] K. Hindriks, C.M. Jonker, D. Tykhonov, Negotiation dynamics: analysis, concession tactics, and outcomes, presented at the in: *Proceedings of the 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, Silicon Valley, San Francisco, CA, USA, 2007.