

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/281561314>

Learning about the opponent in automated bilateral negotiation: a comprehensive survey of opponent modeling techniques

Article in *Autonomous Agents and Multi-Agent Systems* · September 2016

DOI: 10.1007/s10458-015-9309-1

CITATIONS

70

READS

290

4 authors:



Tim Baarslag

Centrum Wiskunde & Informatica

84 PUBLICATIONS 1,195 CITATIONS

[SEE PROFILE](#)



Mark Hendriks

9 PUBLICATIONS 407 CITATIONS

[SEE PROFILE](#)



Koen V. Hindriks

Vrije Universiteit Amsterdam

253 PUBLICATIONS 4,056 CITATIONS

[SEE PROFILE](#)



Catholijn M. Jonker

Delft University of Technology

574 PUBLICATIONS 7,753 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Personal Assistant healthy Lifestyle (PAL) [View project](#)



Autonomous negotiation agents for online peer to peer trading [View project](#)

Learning about the opponent in automated bilateral negotiation: a comprehensive survey of opponent modeling techniques

Tim Baarslag¹ · Mark J. C. Hendriks² ·
Koen V. Hindriks² · Catholijn M. Jonker²

© The Author(s) 2015. This article is published with open access at Springerlink.com

Abstract A negotiation between agents is typically an incomplete information game, where the agents initially do not know their opponent's preferences or strategy. This poses a challenge, as efficient and effective negotiation requires the bidding agent to take the other's wishes and future behavior into account when deciding on a proposal. Therefore, in order to reach better and earlier agreements, an agent can apply learning techniques to construct a *model* of the opponent. There is a mature body of research in negotiation that focuses on modeling the opponent, but there exists no recent survey of commonly used opponent modeling techniques. This work aims to advance and integrate knowledge of the field by providing a comprehensive survey of currently existing opponent models in a bilateral negotiation setting. We discuss all possible ways opponent modeling has been used to benefit agents so far, and we introduce a taxonomy of currently existing opponent models based on their underlying learning techniques. We also present techniques to measure the success of opponent models and provide guidelines for deciding on the appropriate performance measures for every opponent model type in our taxonomy.

Keywords Negotiation · Software agents · Opponent model · Learning techniques · Automated negotiation · Opponent modeling · Machine learning · Survey

✉ Tim Baarslag
T.Baarslag@soton.ac.uk

Mark J. C. Hendriks
M.J.C.Hendriks@tudelft.nl

Koen V. Hindriks
K.V.Hindriks@tudelft.nl

Catholijn M. Jonker
C.M.Jonker@tudelft.nl

¹ University of Southampton, Southampton, UK

² Delft University of Technology, Delft, The Netherlands

1 Introduction

Negotiation is a process in which parties interact to settle a mutual concern to improve their status quo. Negotiation is a core activity in human society, and is studied by various disciplines, including economics [147, 158], artificial intelligence [69, 91, 106, 107, 118, 182], game theory [19, 69, 91, 118, 120, 147, 172], and social psychology [170].

Traditionally, negotiation is a necessary, but time-consuming and expensive activity. Therefore, in the last two decades, there has been a growing interest in the *automation* of negotiation and e-negotiation systems [18, 71, 91, 99, 107], for example in the setting of e-commerce [20, 79, 105, 126]. This attention has been growing since the beginning of the 1980s with the work of early adopters such as Smith's Contract Net Protocol [186], Sycara's PERSUADER [189, 190], Robinson's OZ [164], and the work by Rosenschein [168] and Klein [102]. The interest is fueled by the promise of automated agents being able to negotiate on behalf of human negotiators and to find better outcomes than human negotiators [20, 55, 89, 121, 126, 149, 192].

The potential benefits of automation include reduced time and negotiation costs resulting from automation [33–35, 126], a potential increase in negotiation usage when the user can avoid social confrontation [27, 126], the ability to improve the negotiation skills of the user [80, 121, 125], and the possibility of finding more interesting deals by exploring more promising portions of the outcome space [80, 126].

One of the key challenges for a successful negotiation is that usually only limited information is available about the opponent [143]. Despite the fact that sharing private information can result in mutual gains, negotiators are unwilling to share information in situations with a competitive aspect to avoid exploitation by the other party [50, 75, 81, 158]. In an automated negotiation, this problem can be partially overcome by deriving information from the offers that the agents exchange with each other. Taking advantage of this information to learn aspects of the opponent is called *opponent modeling*.¹

Having a good opponent model is a key factor in improving the quality of the negotiation outcome and can further increase the benefits of automated negotiation, including the following: reaching win-win agreements [90, 123, 206]; minimizing negotiation cost by avoiding non-agreement [151, 153, 183, 184]; and finally, avoiding exploitation by adapting to the opponent's behavior during the negotiation [57, 85, 199]. Experiments have shown that by employing opponent models, automated agents can reach more efficient outcomes than human negotiators [22, 124, 149].

Besides improving the quality of the negotiation process, opponent models are essential for the transition of automated negotiation from theory to practice. It has been shown that non-adaptive agents are exploitable given a sufficiently large negotiation history as their behavior becomes predictable [24, 135]. The risk of exploitation can be minimized by creating adaptive agents that use opponent models to adapt their behavior.

Despite the advantages of creating an opponent model and two decades of research, there is no recent study that provides either an overview of the field, or a comparison of different opponent modeling techniques. Therefore, in order to stimulate the development of efficient future opponent models, and to outline a research agenda for the field of opponent modeling in negotiation, this survey provides an overview of existing opponent models and their underlying concepts. It discusses how to select the best model depending on the negotiation

¹ Despite the usage of the term "opponent", opponent models can be beneficial for both parties. For example, Lin et al. [123, 124] and Oshrat et al. [149] use opponent modeling techniques to maximize joint utility. The term *agent modeling* could also apply to our setting; however, it is in line with current practice to call it *opponent modeling*.

Table 1 Types of negotiation settings discussed in this work. Classification based on Lomuscio et al. [126]

Parameter	Value
Agent setting	Bilateral
Deadline	Private/public
Domain configuration	Single-issue/multi-object/multi-issue
Interaction between issues	Yes/no
Preference profiles	Private/partially disclosed
Sessions	Single/multiple
Strategy	Private/partially disclosed

setting, and identifies a number of problems that are still open. One of our major findings is that despite the variety in opponent modeling techniques, most current models rely on a small, common set of learning techniques. Furthermore, it turns out that there are only four types of opponent attributes that are learned by these techniques.

Apart from employing different techniques to build an opponent model, different benchmarks have been used to test the effectiveness of opponent models. This makes it particularly difficult to compare present techniques. An additional contribution of this work is to give an exhaustive overview of measures that are used throughout the literature. We distinguish two types of measures, and we recommend which measures to use to reliably quantify the quality of an opponent model.

The opponent modeling techniques discussed in this work are applicable to a large diversity of negotiation protocols. Protocols may differ in many aspects, including the domain configuration, the number of agents, state of issue(s), and availability of information. Table 1 provides an overview of the scope of this survey; distinguishing the main parameters of negotiation protocols as defined by Lomuscio et al. [126] and Fatima et al. [65].

Finally, note that the problems involved in automated negotiation are very different from human negotiation. In negotiation sessions between humans, only as little as ten bids may be exchanged, whereas negotiating agents may exchange thousands of bids in less than a minute. Humans may compensate for this lack of information exchange by explicitly communicating information about their preferences, both verbally and nonverbally. To delimit our scope, we do not discuss attributes that are relevant in human negotiations but are not yet used in automated negotiation, such as emotions [54, 100].

The remainder of our work is organized as follows. We start by providing an overview of related surveys in Sect. 2. Section 3 sets out the basic concepts of bilateral negotiation. Section 4 describes the fundamentals underlying the learning methods that have been applied to construct an opponent model. Different opponent models are created to learn different negotiation aspects; we introduce our taxonomy of the various concepts that are learned, and how they are learned, in Sect. 5. Section 6 provides recommendations on how to measure the quality of an opponent model. Finally, in Sect. 7 we cover the lessons learned, we examine the latest trends, and we provide directions for future work.

2 Related surveys

The field of automated negotiation has produced over 2000 papers in the last two decades. This work covers the period from the first opponent models introduced around 1997 (cf. [205])

to the latest models developed in 2014 (cf. [37,46,76,77,92]). During this period, several surveys have been conducted that are related to our work, including surveys by Beam and Segev [18], Papaioannou et al. [152], Masvroula et al. [134], Yang [203], and Chen and Pu [42]. Our work incorporates all techniques for bilateral negotiations covered in these surveys, as we consider various types of opponent models based on multiple different learning techniques, including Bayesian learning and artificial neural networks. In comparison to these surveys, we discuss a larger body of research and categorize the opponent models based on the aspect of the opponent they aim to model. Furthermore, we provide an overview of measures used to quantify the quality of opponent models, and provide guidelines on how to apply these metrics.

Beam and Segev surveyed the state of the art in automated negotiation in 1997 [18]. Their work describes machine learning techniques applied by intelligent negotiation agents, mainly discussing the potential of genetic algorithms to learn an effective negotiation strategy. Their survey naturally misses out on more recent developments, such as on-line opponent modeling techniques used in one-shot negotiations, as for example introduced by Buffett and Spencer [31,32] and Hindriks and Tykhonov [83]. More recently, Papaioannou et al. surveyed learning techniques based on neural networks to model the opponent's behavior in both bilateral and multilateral negotiations [152]. Masvroula et al. also surveyed learning methods to enhance the strategies of negotiation agents [134]. One of the strengths of their survey is that it provides a comprehensive overview of learning methods. The modeling techniques are divided based on the type of strategy in which they are applied. Finally, Chen and Pu survey preference elicitation methods for user modeling in decision support systems [42]. The goal of these systems is to capture the user's preferences in a setting in which the user is willing to share their preferences, or at least does not try to misrepresent them. While the goal of decision support systems differs from opponent modeling in automated negotiation, similar learning techniques—such as pattern matching—are used to estimate the user's or opponent's preferences.

A number of surveys have been conducted on the general topic of automated negotiation, for example by Jennings et al. [91], Kraus [107], Braun et al. [25], and Li et al. [118]. Jennings et al. argue that automated negotiation is a main concern for multi-agent system research [91]; and Kraus examines economic theory and game-theory techniques for reaching agreements in multi-agent environments [107]. Braun et al. review electronic negotiation systems and negotiation agents, concisely describing how learning techniques have been used to learn characteristics of the opponent [25]. Li et al. distinguish different types of negotiation, and briefly discuss opponent modeling [118]. Despite the wide scope of all of the surveys above, their discussion of opponent modeling is limited.

Negotiation is also studied as an extensive-form game within the game theory literature [118], a field of study founded on the work by Nash [141] and Raiffa [157]. In *cooperative* game theory, the aim is to jointly find a solution within the outcome space that satisfies particular axioms, an example being the Nash outcome (see also Sect. 6.1 on accuracy measures, p. 36) that satisfies the Nash axioms [141]. *Non-cooperative* game theory is concerned with identifying rational behavior using the concept of strategy equilibrium: the state in which for every agent it is not beneficial to change strategy assuming the other agents do not switch their tactic [148].

The game theory literature on the topic of negotiation is vast. For an overview we refer to Binmore and Vulkan [19]; Li and Giampapa [118]; and Chatterjee [40]. One prominent example of game theoretic negotiation research is by Rubinstein [172], who considers an alternating offers negotiation protocol without deadline in which two agents negotiate about the division of a pie. Another example is the work by Zlotkin and Rosenschein [208], which

investigates a monotonic concession strategy that results in a strategy equilibrium. As outlined in [64], agents do not typically perform opponent modeling in the game theoretic model, but instead determine their strategy through theoretical analysis, which is possible because of the assumption of perfect rationality. The assumption of common knowledge—an assumption typically made in cooperative game theory—can lead to difficulties in practice [40,41,64,118] as competitive agents aim to not share information to prevent exploitation [50,75,81,158]. Other practical issues include the computational intractability of full agent rationality [40,41,91,118] and the applicability of game theoretical results to specific negotiation settings only [19,91,118]. Despite these concerns, several authors have promoted the application of game theory results in the design of heuristic and learning negotiation strategies [91,118]. For instance, *evolutionary game theory* (EGT) is a framework to describe the dynamics and evolution of strategies under the pressure of natural selection [178]. In this approach, negotiating agents can learn the best strategy through repeated interactions with their opponents. This has just started to make its impact on research into the negotiation dynamics of multi-agent bargaining settings [10,43]. In Sect. 6.2, we discuss EGT as a way to quantify the robustness of a negotiation strategy to exploitability in an open negotiation environment.

An interesting area, although out of scope of this paper, is that of user modeling in general (see e.g., [136] for a survey by McTear on the topic), and in particular that of using machine learning of dialogue-management strategies by Schatzmann and colleagues [180]. McTear's work surveys artificial intelligence techniques applied to user modeling and is by now 20 years old (a newer one has not been published to date). Characteristics of users modeled by AI techniques include goals, plans, capabilities, attitudes, preferences, knowledge, and beliefs. The relevant parts with respect to our survey are the preference profiling and the distinction between learning models of individual users versus models for classes of users, and between models for one session and models maintained and updated over several sessions.

A survey on preference modeling is by Braziunas and Boutilier [26] and focuses on direct elicitation methods; i.e., by asking direct questions to the user and is therefore not in the scope of this paper. Schatzmann's survey [180] addresses systems and methods to learn a good dialogue strategy for which automatic user simulation tools are essential. The methods to learn these strategies can be relevant for argumentation-based negotiation systems.

Another related area of research is the topic of machine learning techniques in game playing; e.g., checkers, rock-paper-scissors, scrabble, go, and bridge. Fürnkranz argues that opponent modeling has not yet received much attention in the computer games community [67]—take for example chess, in which opponent modeling is not a critical component. However, it is essential in others, such as computer poker [171]. This is due to the fact that, as in negotiation, *maximizing* the reward against an effectively exploitable opponent is potentially more beneficial than exhibiting *optimal* play [171]. These surveys make several distinctions that we do, such as offline and online learning, and they employ many techniques that can also be used in negotiation, such as Bayesian learning and neural networks.

3 Preliminaries

Before we discuss opponent models, we first introduce the terminology used throughout the paper. The defining elements of a bilateral negotiation are depicted in Fig. 1. A bilateral automated negotiation concerns a negotiation between *two* agents, usually called *A* and *B* or *buyer* and *seller*. The party that is negotiated with is also called the *partner* or *opponent*.

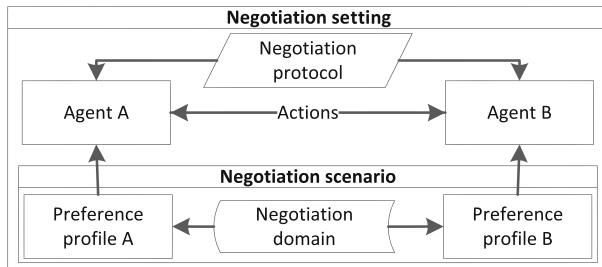


Fig. 1 Overview of the defining elements of an automated bilateral negotiation

The *negotiation setting* consists of the *negotiation protocol*, the *negotiating agents*, and the *negotiation scenario*. The negotiation protocol defines the rules of encounter to which the negotiating agents have to adhere. The negotiation scenario takes place in a *negotiation domain*, which specifies all possible outcomes (the so-called *outcome space*). The negotiation agents have a *preference profile*, which expresses the preference relations between the possible outcomes. Together, this defines the *negotiation scenario* that takes place between the agents. The negotiation scenario and protocol specify the possible *actions* an agent can perform, given the negotiation state.

3.1 Negotiation domain

The *negotiation domain*—or *outcome space*—is denoted by Ω and defines the set of possible negotiation outcomes. The *domain size* is the number of possible outcomes $|\Omega|$. A negotiation domain consists of one or more *issues*, which are the main resources or considerations that need to be resolved through negotiation; for example, the price or the color of a car that is for sale. Issues are also sometimes referred to as attributes, but we reserve the latter term for *opponent attributes*, which are properties that may be useful to model to gain an advantage in a negotiation.

To reach an agreement, the agents must settle on a specific alternative or *value* for each negotiated issue. That is, an *agreement* on n issues is an outcome that is accepted by both parties of the form $\omega = \langle \omega_1, \dots, \omega_n \rangle$, where ω_i denotes a value associated with the i th issue. We will focus mainly on settings with a finite set of discrete values per issue. A *partial agreement* is an agreement on a subset of the issues. We say that an outcome space defined by a single issue is a *single-issue negotiation*, and a *multi-issue negotiation* otherwise.

Negotiating agents can be designed either as general purpose negotiators, that is, *domain-independent* [122] and able to negotiate in many different settings, or suitable for only one specific domain (e.g., the *Colored Trail* domain [66, 68], or the *Diplomacy* game [52, 56, 108]). There are obvious advantages to having an agent designed for a specific domain: it enables the agent designer to construct more effective strategies that exploit domain-specific information. However, this is also one of the major weaknesses, as such agents need to be tailored to every new available domain and application; this is why many of the agents and learning mechanisms covered in this survey are domain-independent.

3.2 Negotiation protocol

A negotiation protocol fixes the rules of encounter [169], specifying which actions each agent can perform at any given moment. Put another way, it specifies the admissible *negotiation moves*. The literature discussed in this survey assumes that the protocol is shared knowledge,

and that the agents strictly adhere to it. Our focus here is on bilateral negotiation protocols. For other work in terms of one-to-many and many-to-many negotiations (for example to learn when to pursue more attractive outside options in a setting with multiple agents), we refer to [3, 119, 142, 147, 156, 183]. We do not aim to provide a complete overview of all protocols, instead we recommend Lomuscio et al. [126] for an overview of high-level parameters used to classify them, and Marsa-Maestre et al. [128] for guidelines on how to choose the most appropriate protocol to a particular negotiation problem.

An often used negotiation protocol in bilateral automated negotiation is the *alternating offers protocol*, which is widely studied and used in the literature, both in game-theoretic and heuristic settings (a non-exhaustive list includes [61, 107, 109, 147, 148]). This protocol dictates that the two negotiating agents propose *outcomes*, also called *bids* or *offers*, in turns. That is, the agents create a *bidding history*: one agent proposes an offer, after which the other agent proposes a counter-offer, and this process is repeated until the negotiation is finished, for example by time running out, or by one of the parties accepting.

In the alternating offers setting, when agent A receives an offer $x_{B \rightarrow A}$ from agent B , it has to decide at a later time whether to accept the offer, or to send a counter-offer $x_{A \rightarrow B}$. Given a bidding history between agents A and B , we can express the action performed by A with a *decision function* [62, 181]. The resulting action is used to extend the current bidding history between the two agents. If the agent does not accept the current offer, and the deadline has not been reached, it will prepare a counter-offer by using a negotiation strategy or *tactic* to generate new values for the negotiable issues (see Sect. 3.6).

Various alternative versions of the alternating offers protocol have been used in automated negotiation, extending the default protocol, and imposing additional constraints; for example, in a variant called the *monotonic concession protocol* [143, 169], agents are required to initially disclose information about their preference order associated with each issue and the offers proposed by each agent must be a sequence of concessions, i.e.: each consecutive offer has less utility for the agent than the previous one. Other examples are the three protocols discussed by Fatima et al. [65] that differ in the way the issues are negotiated: simultaneously in bundles, in parallel but independently, and sequentially. The first alternative is shown to lead to the highest quality outcomes. A final example is a protocol in which only *one* offer can be made. In such a situation, the negotiation can be seen as an instance of the ultimatum game, in which a player proposes a deal that the other player may only accept or refuse [185]. In [176], a similar bargaining model is explored as well; that is, models with one-sided incomplete information and one sided offers. It investigates the role of confrontation in negotiations and uses optimal stopping to decide whether or not to invoke conflict.

3.3 Preference profiles

Negotiating agents are assumed to have a *preference profile*, which is a *preference order* \geq that ranks the outcomes in the outcome space. Preferences are said to be *ordinal* when they are fully specified by a preference order. Together with the domain they make up the *negotiation scenario*.

In many cases, the domain and preferences stay fixed during a single negotiation encounter, but while the domain is common knowledge to the negotiating parties, the preferences of each player are private information. This means that the players do not have access to the preferences of the opponent. In this sense, the negotiators play a game of *incomplete information*. However, the players can attempt to learn as much as they can during the negotiation encounter.

An outcome ω' is said to be *weakly preferred* over an outcome ω if $\omega' \geq \omega$. If in addition $\omega \not\geq \omega'$, then ω' is *strictly preferred* over ω , denoted $\omega' > \omega$. An agent is said to be *indifferent* between two outcomes if $\omega' \geq \omega$ and $\omega \geq \omega'$. In that case, we also say that these outcomes are *equally valued* and we write $\omega' \sim \omega$. An *indifference curve* or *iso-curve* is a set of outcomes that are equally valued by an agent. In a *total preference order*, one outcome is always (weakly) preferred over the other outcome for any outcome pair, which means there are no undefined preference relations. Finally, an outcome ω is *Pareto optimal* if there exists no outcome ω' that is preferred by an agent without making another agent worse off [158]. For two players A and B with respective preference orders \geq_A and \geq_B , this means that there is no outcome ω' such that:

$$(\omega' >_A \omega \wedge \omega' \geq_B \omega) \vee (\omega' >_B \omega \wedge \omega' \geq_A \omega).$$

An outcome that is Pareto optimal is also said to be *Pareto efficient*. When an outcome is not Pareto efficient, there is potential, through re-negotiation, to reach a more preferred outcome for at least one of the agents without reducing the value for the other.

The outcome space can become quite large, which means it is usually not viable to explicitly state an agent's preference for every alternative. For this reason, there are more succinct preference representations for preferences [48, 53].

A well-known and compact way to represent preference orders is the formalism of conditional preference networks (CP-nets) [23]. CP-nets are graphical models, in which each node represents an negotiation issue and each edge denotes preferential dependency between issues. If there is an edge from issue i to issue j , the preferences for j depend on the specific value for issue i . To express conditional preferences, each issue is associated with a conditional preference table, which represents a total order of possible values for that issue, given its parents' values.

A preference profile may be specified as a list of ordering relations, but it is more common in the literature to express the agent's preferences by a *utility function*. A utility function assigns a utility value to every possible outcome, yielding a *cardinal* preference structure.

Cardinal preferences are 'richer' than ordinal preferences in the sense that ordinal preferences can only compare between different alternatives, while cardinal preferences allow for expressing the intensity of every preference [48]. Any cardinal preference induces an ordinal preference, as every utility function u defines an order $\omega' \geq \omega$ if and only if $u(\omega') \geq u(\omega)$.

Some learning techniques make additional assumptions about the structure of the utility function [98], the most common in negotiation being that the utility of a multi-issue outcome is calculated by means of a *linear additive function* that evaluates each issue separately [98, 158, 159]. Hence, the contribution of every issue to the utility is linear and does not depend on the values of other issues. The utility $u(\omega)$ of an outcome $\omega = \langle \omega_1, \dots, \omega_n \rangle \in \Omega$ can be computed as a weighted sum from evaluation functions $e_i(\omega_i)$ as follows:

$$u(\omega) = \sum_{i=1}^n w_i \cdot e_i(\omega_i), \quad (1)$$

where the w_i are normalized weights (i.e. $\sum w_i = 1$). Linear additive utility functions make explicit that different issues can be of different importance to a negotiating agent and can be used to efficiently calculate the utility of a bid at the cost of expressive power, as they cannot represent *interaction effects* (or dependencies) between issues.

A common alternative is to make use of non-linear utility functions to capture more complex relations between offers at the cost of additional computational complexity. Non-linear negotiation is an emerging area within automated negotiation that considers multiple

inter-dependent issues [88, 129]. Typically this leads to larger, richer outcome spaces in comparison to linear additive utility functions. A key factor in non-linear spaces is the ability of a negotiator to make a proper evaluation of a proposal, as the utility calculation of an offer might even prove NP-hard [52]. Examples of this type of work can be found in [87, 101, 127, 166].

For non-linear utility functions in particular, a number of preference representations have been formulated to avoid listing the exponentially many alternatives with their utility assessment [48]. The utility of a deal can be expressed as the sum of the utility values of all the *constraints* (i.e., regions in the outcome space) that are satisfied [87, 130]. These constraints may in turn exhibit additional structure, such as being represented by hyper-graphs [74]. One can also decompose the utility function into *subclusters* of individual issues, such that the utility of an agreement is equal to the sum of the sub-utilities of different clusters [166]. This is a special case of a utility structure called *k-additivity*, in which the utility assigned to a deal can be represented as the sum of basic utilities of subsets with cardinality $\leq k$ [49]. For example, for $k = 2$, the utility $u(\omega_1, \omega_2, \omega_3)$ might be expressed as the utility value of the individual issues $u_1(\omega_1) + u_2(\omega_2) + u_3(\omega_3)$ (as in the linear additive case), plus their 2-way interaction effects $u_4(\omega_1, \omega_2) + u_5(\omega_1, \omega_3) + u_6(\omega_2, \omega_3)$. This is in turn closely related to the OR and XOR languages for bidding in auctions [144], in which the utility is specified for a specific set of clusters, together with rules on how to combine them into utility functions on the whole outcome space.

Finally, the preference profile of an agent may also specify a *reservation value*. The reservation value is the minimal utility that the agent still deems an acceptable outcome. That is, the reservation value is equal to the utility of the best alternative to no agreement. A bid with a utility lower than the reservation value should not be offered or accepted by any rational agent. In a single-issue domain, the negotiation is often about the *price* P of a good [59, 62, 205, 206]. In that case, the agents usually take the roles of buyer and seller, and their reservation values are specified by their *reservation prices*; i.e., the highest price a buyer is willing to pay, and the lowest price at which a seller is willing to sell.

3.4 Time

Time in negotiation is limited, either because the issues under negotiation may expire, or one or more parties are pressing for an agreement [39]. Without time pressure, the negotiators have no incentive to accept an offer, and so the negotiation might go on forever. Also, with unlimited time an agent may simply try a large number of proposals to learn the opponent's preferences. The *deadline* of a negotiation refers to the time before which an agreement must be reached [158]. When the deadline is reached, the negotiators revert to their best alternative to no agreement.

The negotiator's nearness to a deadline is only one example of *time pressure* [38], which is defined as a negotiator's desire to end the negotiation quickly [154]. An alternative way to model time pressure is to supplement the negotiation scenario with a *discount factor*, which models the decline of the negotiated goods over time. Let δ in $[0, 1]$ be the discount factor and let t in $[0, 1]$ be the current normalized time. A way to compute the real-time discounted utility $u^\delta(\omega)$ from the undiscounted utility $u(\omega)$ is as follows:

$$u^\delta(\omega) = u(\omega) \cdot \delta^t. \quad (2)$$

If $\delta = 1$, the utility is not affected by time, and such a scenario is considered to be undiscounted, while if δ is very small, there is high pressure on the agents to reach an agreement.

Alternatively, time may be viewed as a discrete variable, in which the number of negotiation exchanges (or *rounds*) are counted. In that case, the deadline is specified as a maximum number of rounds n and discounting is applied in every round $k \leq n$ as $u^\delta(\omega) = u(\omega) \cdot \delta^k$. Note that, from a utility point of view, the presence of a discount factor δ is equivalent to the probability $1 - \delta$ that the opponent walks away from the negotiation in any given negotiation round.

Deadlines and discount factors can have a strong effect on the outcome of a negotiation and may also interact with each other. For example, it is shown in [177] that in a game-theoretic setting with fully rational play, time preferences in terms of deadlines may lead to a game of ‘sit and wait’ and may completely override other effects such as time discounting.

3.5 Outcome spaces

A useful way to visualize the preferences of both players simultaneously is by means of an *outcome space plot* (Fig. 2). The axes of the outcome space plot represent the utilities of player A and B , and every possible outcome $\omega \in \Omega$ maps to a point $(u_A(\omega), u_B(\omega))$. The line that connects all of the Pareto optimal agreements is the Pareto frontier.

Note that the visualization of the outcome space together with the Pareto frontier is only possible from an *external* point of view. In particular, the agents themselves are not aware of the opponent utility of bids in the outcome space and do not know the location of the Pareto frontier.

From Fig. 2 we can immediately observe certain characteristics of the negotiation scenario that are very important for the learning behavior of an agent. Examples include the domain size, the relative occurrence of Pareto optimal outcomes, and whether the bids are spread out over the domain.

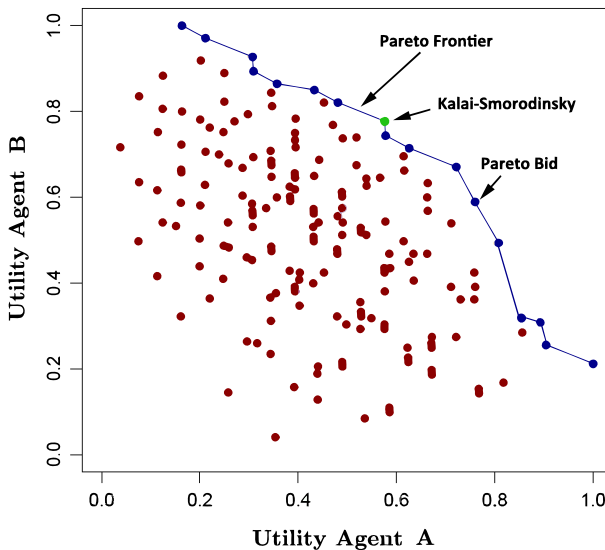


Fig. 2 A typical example of an outcome space between agents A and B . The *points* represent all outcomes that are possible in the negotiation scenario. The *line* is the Pareto frontier, which connects all of the Pareto efficient outcomes

3.6 Negotiation tactics

The *bidding strategy*, also called *negotiation tactic* or *concession strategy*, is usually a complex strategy component. Two types of negotiation tactics are very common: *time-dependent tactics* and *behavior-dependent tactics*. Each tactic uses a *decision function*, which maps the negotiation state to a *target utility*. Next, the agent can search for a bid with a utility close to the target utility and offer this bid to the opponent.

3.6.1 Time-dependent tactics

Functions which return an offer solely based on time are called *time-dependent tactics*. The standard time-dependent strategy calculates a target utility $u(t)$ at every turn, based on the current time t . Perhaps the most popular time-based decision function can be found in [59,61], which, depending on the current normalized time $t \in [0, 1]$, makes a bid with utility closest to

$$u(t) = P_{\min} + (P_{\max} - P_{\min}) \cdot (1 - F(t)), \quad (3)$$

where

$$F(t) = k + (1 - k) \cdot t^{1/e}.$$

The constants $P_{\min}, P_{\max} \in [0, 1]$ control the range of the proposed offers, and $k \in [0, 1]$ determines the value of the first proposal. For $0 < e < 1$, the agent concedes only at the end of the negotiation and is called a *Boulware* negotiation tactic. If $e \geq 1$, the function concedes quickly to the reservation value, and the agent is then called a *Conceder*. Figure 3 shows a plot of several time-dependent tactics for varying concession factors e .

The specification of these strategies given in [59,61] does not involve any opponent modeling; that is, given the target utility, a random bid is offered with a utility closest to it.

3.6.2 Baseline tactics

The *Hardliner* strategy (also known as *take-it-or-leave-it*, *sit-and-wait* [4] or *Hardball* [117]) can be viewed as an extreme type of time-dependent tactic. This strategy stubbornly makes a bid of maximum utility for itself and never concedes, at the risk of reaching no agreement.

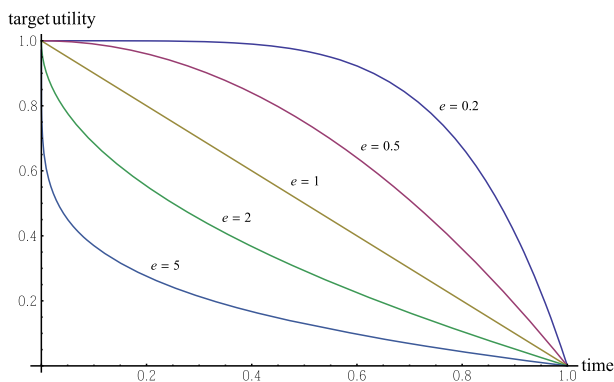


Fig. 3 Target utility through time of time-dependent tactics with concession factor $e \in \{0.2, 0.5, 1, 2, 5\}$

Random Walker (also known as the *Zero Intelligence* strategy [70]) generates random bids and thus provides the extreme case of a maximally unpredictable opponent. Because of its limited capabilities, it can also serve as a useful baseline strategy when testing the efficacy of other negotiation strategies.

3.6.3 Behavior-dependent tactics

Faratin et al. introduce a well-known set of *behavior-dependent tactics* or *imitative tactics* in [59]. The most well-known example of a behavior-dependent tactic is the *Tit for Tat* strategy, which tries to reproduce the opponent's behavior of the previous negotiation rounds by reciprocating the opponent's concessions. Thus, *Tit for Tat* is a strategy of cooperation based on reciprocity [5].

Tit for Tat has been applied and found successful in many other games, including the *Iterated Prisoner's Dilemma game* [6]. In total three tactics are defined: *Relative Tit for Tat*, *Random Absolute Tit for Tat*, and *Averaged Tit for Tat*. The *Relative Tit for Tat* agent mimics the opponent in a percentage-wise fashion by proportionally replicating the opponent's concession that was performed a number of steps ago.

The standard *Tit for Tat* strategies from [59] do not employ any learning methods, but this work has been subsequently extended by the *Nice Tit for Tat agent* [15] and the *Nice Mirroring Strategy* [81]. These strategies achieve more effective results by combining a simple *Tit for Tat* response mechanism with learning techniques to propose offers closer to the Pareto frontier.

4 Learning methods for opponent models

An extensive set of learning techniques have been applied in automated negotiation. Below we provide an introduction to the most commonly used underlying methods. Those that are already familiar with these techniques can skip to the next section.

The first two sections discuss *Bayesian Learning* (Sect. 4.1) and *Non-linear Regression* (Sect. 4.2). Both methods have mainly been applied as an online learning technique, because they do not require a training phase to produce a reasonable estimate, and because their estimates can be improved incrementally during the negotiation.

In contrast, the other two methods, *Kernel Density Estimation* (Sect. 4.3) and *Artificial Neural Networks* (Sect. 4.4), generally require a training phase, and are mainly applied when a record of the negotiation history is available. With these methods, it is computationally inexpensive to take advantage of the learned information during the negotiation.

4.1 Bayesian learning

Bayesian learning is the most prominent probabilistic approach in opponent modeling. Bayesian learning is based on Bayes' rule:

$$P(H | E) = \frac{P(E | H) \cdot P(H)}{P(E)}. \quad (4)$$

Bayes' rule is a tool for updating the probability that a hypothesis H holds based on observed evidence E . In the formula above, $P(H | E)$ is the *posterior probability* that the hypothesis H holds given evidence E , and $P(E | H)$ is called the *conditional probability* of an event E occurring given the hypothesis H . $P(H)$ denotes the *prior probability* for the hypothesis,

independent of any evidence, and similarly, $P(E)$ denotes the prior probability that evidence E is observed.

Bayesian learning is typically used to identify the most likely hypothesis H_i out of a set of hypotheses $\mathcal{H} = \{H_1, \dots, H_n\}$. In the negotiation literature typically a finite set of hypotheses is assumed, for example the *type* of the opponent. In that case the likelihood of the hypotheses given observed evidence E can be determined using the alternative formulation of the Bayes' rule:

$$P(H_i | E) = \frac{P(E | H_i) \cdot P(H_i)}{\sum_{j=1}^n P(E | H_j) \cdot P(H_j)}. \quad (5)$$

An agent can formulate a set of independent hypotheses about a property of the opponent and discover—using evidence—which hypothesis is most likely valid. The idea is that each time new evidence E is observed, we can use Equation (5) to update and compute an improved estimate of the posterior probability $P(H_i | E)$. After processing the evidence, an agent can conclude which hypothesis is most probable.

One disadvantage of using Bayesian learning is its computational complexity. Updating a single hypothesis H_i given a piece of evidence E_k may have a low computational complexity; however, there may be many such hypotheses H_i , and pieces of evidence E_k . For example, when modeling the opponent's preferences, this set of hypotheses can be custom made, or generated from the structure of the functions assumed to model the preferences. Even in a negotiation scenario with linear additive utility functions, modeling the preferences requires a set of preference profiles for each negotiable issue. This already leads to a number of hypotheses that is exponential in the number of issues. Another challenge lies in defining the right input for the learning method (e.g. finding a suitable representation of the opponent's preference profile); in general it is not straightforward to define a suitable class of hypotheses, and it may be hard to determine the conditional probabilities.

4.2 Non-linear regression

Non-linear regression is a broad field of research, and we only present the aspects needed for the application of this technique to opponent modeling. We provide a brief introduction based on [138]. For a more complete overview of the field of non-linear regression, we refer to [17].

Non-linear regression is used to derive a function which “best matches” a set of observational sample data. It is employed when we expect the data to display a certain functional relationship between input and output, from which we can then interpolate new data points. A typical negotiation application is to estimate the opponent's future behavior from the negotiation history assuming that the opponent's bidding strategy uses a known formula with unknown parameters.

A simple non-linear regression model consists of four elements: the *dependent* (or *response*) variable, the *independent* (or *predictor*) variables, the (non-linear) *formula*, and its *parameters*. To illustrate, suppose we have a set of observations as shown in Fig. 4, and we want to find the relationship between x and y in order to predict the value of y for new values of x . Suppose the relationship is believed to have the form $y'(x) = ax^2 + bx + c$, where a and b are parameters with unknown values. In this formula, y' is the dependent variable and x is the independent variable. Using non-linear regression, we can estimate the parameters a and b such that the error between the predicted y' values and the observed y values is minimized. The error is calculated using a loss function. In the negotiation literature typically the error is calculated as the sum of squared differences between the predicted and observed values.

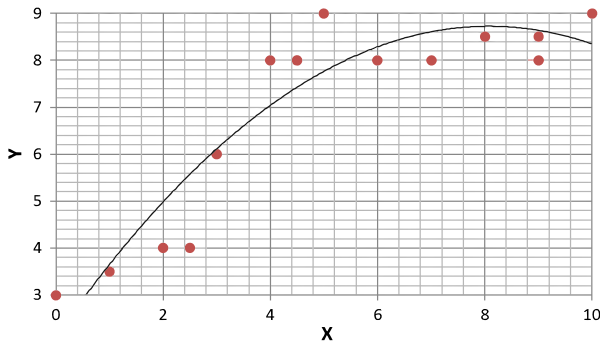


Fig. 4 Example of a non-linear regression based on a polynomial of the second degree. The best fit is shown as the *black line*

Alternative loss functions may for example calculate the absolute difference, or treat positive and negative errors differently.

The parameters for the quadratic formula discussed in this example can be solved using a closed-form expression. Non-linear regression is typically used when this is not possible, for example when there are a large number of parameters that have a non-linear relation with the solution. The calculation of the parameters is based on an initial guess of the parameters, after which an iterative hill-climbing algorithm is applied to refine the guess until the error becomes negligible. Commonly used algorithms are the Marquardt Method and the simplex algorithm. An introduction to both these methods is provided by Motulsky and Ransnas [138]. The main problem with hill-climbing algorithms is that they can return a local optimum instead of the global optimum. Furthermore, in extreme cases the algorithm may even not converge at all. This can be resolved by using multiple initial estimates and selecting the best fit after a specified amount of iterations.

4.3 Kernel density estimation

Kernel density estimation (KDE) is a mathematical technique used to estimate the probability distribution of a population given a set of population samples [50]. Figure 5 illustrates the estimated probability density function constructed from six observations.

The first step of KDE consists of converting each sample into a so-called *kernel function*. A kernel function is a probability distribution which quantifies the uncertainty of the observation. Common choices for kernels are the standard normal distribution or the uniform probability distribution. The second step is to accumulate all kernels to estimate the probability distribution of the population.

While KDE makes no assumptions about the values of the samples, or in which order the samples are obtained, the kernel function typically requires a parameter called *bandwidth*, which determines the width of each kernel. When a large number of samples is available over the complete range of the variable of interest, then a small bandwidth can lead to an accurate estimate. With few samples, a large bandwidth can help generalize the limited available information. The choice of bandwidth needs to strike a balance between under-fitting and over-fitting the resulting distributions. As there is no choice that works optimally in all cases, heuristics have been developed for estimating the bandwidth. Jones et al. provide an overview of commonly used bandwidth estimators [93]. The heuristics are based on

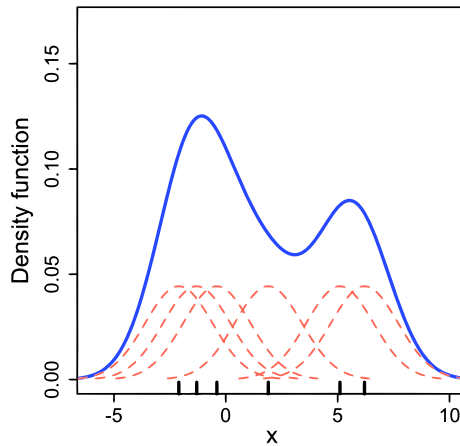


Fig. 5 Example of the use of KDE. Observations are marked with a *line* on the *x* axis. The *dashed lines* mark the standard normal kernels that sum up to the probability density estimation indicated by the *solid line*

statistical characteristics of the sample set, such as the sample variance and sample count. The estimation quality of KDE can be further improved by varying the bandwidth for each kernel, for example based on the amount of samples found in a window centered at each observation. Using an adaptive bandwidth is called adaptive (or variable) KDE and can further decrease the estimation error at the cost of additional workload.

KDE is a computationally attractive learning method. The computationally intensive parts (automatic bandwidth selection and the construction of a kernel density estimate) can be done offline, after which the lookup can be performed during the negotiation.

4.4 Artificial neural networks

Artificial neural networks are networks of simple computational units that together can solve complex problems. Below we provide a short introduction to artificial neural networks (ANN's) based on Kröse et al. [110]. Our overview is necessarily incomplete due to broadness of the field; therefore, for a more complete overview we refer to Haykin [78] and for a survey of the applications of neural networks in automated negotiation to Papaioannou et al. [152].

An ANN is a computational model with the ability to learn the relationship between the input and output by minimizing the error between the output signal of the ANN and the expected output. Since all that is required is a mechanism to calculate the error, ANN's can be applied when the relation between input and output is unknown. ANN's have been used for several purposes, including classification, remembering, and structuring of data.

A neural network consists of computational units called neurons, which are connected by weighted edges. Figure 6 visualizes a simple neural network consisting of six neurons. A single neuron can have several incoming and outgoing edges. When a neuron has received all inputs, it combines them according to a combination rule, for example the sum of the inputs. Next, it tests whether it is triggered by this input by using an activation function; e.g., whether a threshold has been exceeded or not. If the neuron is triggered, it propagates the combined signal over the output lines, else it sends a predefined signal.

The set of neurons function in an environment that provides the input signals and processes the output signals of the ANN. The environment calculates the error of the output, which the

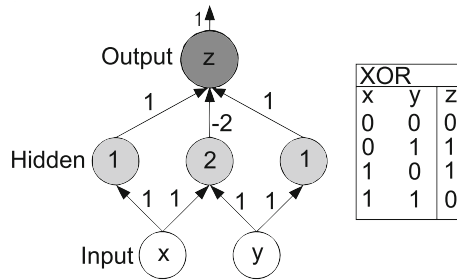


Fig. 6 Example of an ANN symbolizing the logical XOR. The input neurons expect a value of 0 or 1. The edges show the weights. The activation functions are threshold functions, which values are depicted inside the nodes. The combination rule is the sum of the inputs. If the combined input is larger than or equal to the threshold the combined input is propagated. Otherwise, the value 0 is propagated on the output line

neural network uses to better learn the relation between the input and output by adjusting the weights on the edges between the neurons.

Neurons can be ordered in successive layers based on their depth. In Fig. 6 each layer has a unique color. The first layer is called the input layer, the last one is the output layer. Both the input and output neuron generally have no activation function. The layers in between are called hidden layers as they are not directly connected to the environment.

To illustrate how a simple neural network works, assume that the input $x = 0$ and $y = 1$ are fed to the network in Fig. 6. In that case, the leftmost light gray neuron receives input 0, which results in the output 0, as the neuron is not triggered. The rightmost light gray neuron however, is triggered since it receives input 1 and therefore propagates the output 1. The middle light gray neuron receives the input 0 and 1, which are integrated using the combination rule. The combined signal is insufficient to trigger the neuron, resulting in a 0 as output. Since the rightmost light gray neuron is the only neuron that produced a non-zero output, the final output is 1.

The amount of neurons and their topology determines the complexity of the input-output relationship that the ANN can learn. Overall, the more neurons and layers, the more flexible the ANN. However, the more complex the ANN, the more complex the learning algorithm and consequentially the higher the computational cost of learning.

An ANN is typically used when there is a large amount of sample data available, and when it is difficult to capture the relationship between input and output in a functional description; e.g., when negotiating against humans.

5 Learning about the opponent

A bilateral negotiation may be viewed as a two-player game of incomplete information where both players aim to achieve the best outcome for themselves. In general, an opponent model is an abstracted description of a player (and/or its behavior) during the game [193]. In negotiation, opponent modeling often revolves around three questions:

- **Preference estimation** What does the opponent want?
- **Strategy prediction** What will the opponent do, and when?
- **Opponent classification** What type of player is the opponent, and how should we act accordingly?

These questions are often highly related. For example, some form of preference estimation is needed in order to understand how the opponent has acted according to its own utility. Then,

by adequately interpret the opponent's actions, we may deduce its strategy, which in turn can help predict what the agent will do in the future.

Constructing an opponent model may alternatively be viewed as a *classification problem* where the *type* of the opponent needs to be determined from a range of possibilities [179]; one example being the work by Lin et al. [124]. Here the *type* of an opponent refers to all opponent attributes that may be modeled to gain an advantage in the game. Taking this perspective is particularly useful when a limited number of opponent types are known in advance, which at the same time is its main limitation.

Note that our definition excludes work in which a pool of agents are tuned or evolved to optimize their performance when playing against each other, without having an explicit opponent modeling component themselves. For readers interested in this type of approach we refer to Liang and Yuan [120], Oliver [145], Sánchez-Anguix et al. [175], and Tu et al. [191].

Opponent modeling can be performed online or offline, depending on the availability of historical data. Offline models are created before the negotiation starts, using previously obtained data from earlier negotiations. Online models are constructed from knowledge that is collected during a single negotiation session. A major challenge in online opponent modeling is that the model needs to be constructed from a limited amount of negotiation exchanges, and a real-time deadline may pose the additional challenge of having to construct the model as fast as possible.

Opponent modeling can be performed at many different levels of granularity. The most elementary of preference models may only yield a set of offers likely to be accepted by the opponent, for instance by modeling the reservation value. A more detailed preference model is able to estimate the acceptance *probability* for every outcome (e.g. using a probabilistic representation of the reservation value). An even richer model can involve the opponent's *preference order*, allowing us to rank the outcomes. We can achieve the richest preference representations with a *cardinal model* of preferences, yielding an estimate of the opponent's full preference profile. The preferred form of granularity depends not only on the complexity of the negotiation scenario, but also on the level of information required by the agent. For instance, if the agent is required to locate Pareto optimal outcomes in a multi-issue domain, it will require at least an ordinal preference model.

Note that in most cases, comparing different approaches is impossible due to the variety of quality measures, evaluation techniques and testbeds in use; we will have more to say on how to evaluate the different approaches in Sect. 6.

Even though there are large differences between the models, a common set of high level motivations behind their construction can be identified. We found the following motivations for why opponent models have been used in automated negotiation:

1. **Minimize negotiation cost** [7–9, 11, 50, 72, 73, 90, 103, 113, 137, 143, 146, 149, 151, 153, 155, 160, 165, 166, 183, 184, 188, 205–207] In general, it costs time and resources to negotiate. As a consequence, (early) agreements are often preferred over not reaching an agreement. As such, an opponent model of the opponent's strategy or preference profile aids towards minimizing negotiation costs, by determining the bids that are likely to be accepted by the opponent. An agent may even decide that the estimated negotiation costs are too high to warrant a potential agreement, and prematurely end the negotiation.
2. **Adapt to the opponent** [1, 15, 28, 29, 44–46, 57, 72, 73, 75, 81, 82, 85, 92, 133, 139, 140, 150, 155, 162, 173, 197, 199, 204] With the assistance of an opponent model, an agent can adapt to the opponent in multiple ways. One way is to estimate the opponent's reservation value in an attempt to deduce the best possible outcome that the opponent will settle for.

Another method is to use an estimate of the opponent's deadline to elicit concessions from the opponent by stalling the negotiation, provided, of course, that the agent itself has a later deadline. Finally, an opponent model can be used to estimate the opponent's concessions to accurately reciprocate them.

3. **Reach win-win agreements** [11, 12, 15, 21, 30, 50, 51, 81, 83, 90, 94, 103, 113, 123, 124, 137, 143, 146, 149, 155, 160, 165, 166, 174, 183, 184, 188, 194, 195, 198, 205–207] In a cooperative environment, agents aim for a fair result, for example because there might be opportunity for future negotiations. Cooperation, however, does not necessarily imply that the parties share explicit information about their preferences or strategy, as agents may still strive for a result that is beneficial for themselves and acceptable for their opponent. An agent can estimate the opponent's preference profile to maximize joint utility.

We found that existing work on opponent models can fulfill any of the goals above by learning a combination of *four* opponent attributes, which we have listed in Table 2. The remainder of this section discusses, for each attribute, the applicable opponent modeling techniques, following the order of Table 2.

5.1 Learning the acceptance strategy

All negotiation agent implementations need to deal with the question of when to accept. The decision is made by the *acceptance strategy* of a negotiating agent, which is a boolean function indicating whether the agent should accept the opponent's offer. Upon acceptance of an offer, the negotiation ends in agreement, otherwise it continues. More complex acceptance strategies may be probabilistic or include the possibility of breaking off the negotiation without an agreement—if that is supported by the protocol.

A common default is for the agent to accept a proposal when the value of the offered contract is higher than the offer it is ready to send out at that moment in time. The bidding strategy then effectively dictates the acceptance strategy, making this a significant case in which it suffices to learn the opponent's bidding strategy (see Sect. 5.4). Examples include the time-dependent negotiation strategies defined in [167] (e.g. the *Boulware* and *Conceder* tactics). The same principle is used in the equilibrium strategies of [61] and the *Trade-off agent* [60]. Other agents use much more sophisticated methods to accept; for example, acceptance strategies based on extrapolation of all received offers [97], dynamic time-based acceptance [2, 51], and optimal stopping [13, 104, 116, 176, 201].

Learning an opponent's acceptance strategy is potentially of great value as it can help to find the best possible deal for an agent which at the same time is satisfactory for the opponent. Two general approaches have been used to estimate the acceptance strategy, depending on the negotiation domain:

- *Estimating the reservation value* (Sect. 5.1.1) In a negotiation about a single quantitative issue, where the opponent's have opposing preferences that are publicly known—such as the price of a service—, knowledge of the opponent's reservation value is sufficient to determine all acceptable bids. An opponent model can learn the opponent's reservation value by extrapolating the opponent's concessions.
- *Estimating the acceptance strategy* (Sect. 5.1.2) An alternative approach applied to multi-issue negotiations is to estimate the probability that a particular offer is accepted, based on the similarity with bids that the opponent previously offered and/or accepted.

Table 2 All learning techniques and methods that help to learn four different opponent attributes

Opponent attributes	Procedure	Learning technique
5.1 Acceptance strategy	Bidding strategy estimation	Bayesian learning [72,92,162,183,184,204–207]
		Non-linear regression [1,73,85,204]
	Interpolation of acceptance likelihood	Bayesian learning [113]
		Kernel density estimation [149] Neural networks [57] Polynomial interpolation [173]
5.2 Deadline	Bidding strategy estimation	Bayesian learning [72,92,184,204] Non-linear regression [72,85,184,204]
5.3 Preference profile	Estimation of issue preference order	Bayesian learning [143] Heuristics [21,37,94] Kernel density estimation [50,58] Simplified genetic algorithm [90]
	Classification	Bayesian learning [15,31,32,51,83,123,124,155,198]
	Data mining aggregate preferences	Bayesian network [174] Graph theory [165,166] Random variable estimation [103,188]
5.4 Bidding strategy	Regression analysis	Heuristics [7–9,30,75–77,163,194,195]
	Time series forecasting	Bayesian networks [139] Genetic algorithms [151] Non-linear regression [1,28,73,85,153,161,204] Polynomial interpolation [153]
		Derivatives [29,137] Markov chains [140] Neural networks [35,36,115,133,146,151,153,160] Signal processing [44–46,132,150,197,199]

5.1.1 Learning the acceptance strategy by estimating the reservation value

Current methods for estimating the reservation value stem from the idea that an agent will cease to concede near its reservation value, and that this behavior occurs when the negotiation deadline approaches. These methods make assumptions about the availability of domain knowledge, or assume that the opponent uses a particular strategy.

The oldest and most popular approach is by Zeng and Sycara [205,206], who propose a Bayesian learning method to estimate the reservation value, using data from previous negotiations. One single quantitative issue is negotiated, for which it is assumed the agents have opposing preferences. Before the negotiation, a set of hypotheses $\mathcal{H} = \{H_1, \dots, H_n\}$ about the opponent's reservation value is generated. Each hypothesis H_i is of the form $rv = v_i$,

where v_i is one of the possible values for the opponent's reservation value rv . The hypotheses, the values v_i , and their a priori likelihood, are all determined based on domain knowledge derived from previous negotiations. By applying Bayesian learning during the negotiation, the probabilities of the hypotheses are updated based on observed behavior and the available domain knowledge. Intuitively, the idea is that an offer at the beginning of the negotiation is likely to be far from the reservation value. The reservation value is estimated by using the weighted sum of the hypotheses according to their likelihood. This method is widely applied; for example in work by Ren and Anumba [162] and Zhang et al. [207].

Closely related to the work by Zeng and Sycara is the work by Sim et al., who apply the same procedure when the opponent is constrained to use a particular time-dependent tactic, but with a private deadline [72, 92, 183, 184]. The opponent's decision function is assumed to be of a particular form in which the reservation value and deadline are related, in the sense that one can be derived from the other.

A different approach is taken by Hou [85], who presents a method to estimate the opponent's tactic in a negotiation about a single quantitative issue with private deadlines. It is assumed that the opponent employs a tactic dependent on either time, behavior, or resources. Non-linear regression is used to estimate which of the three types of strategies is used, and to estimate the values of the parameters associated with the tactic [85], including the reservation value (cf. Sect. 5.4.1). A similar approach is followed by Agrawal and Chari, who model the opponent's decision function as an exponential function [1]. When the deadline is public knowledge, Haberland's method [73] can be used to estimate the opponent's reservation value assuming the opponent uses a time-dependent tactic.

To improve reliability of the estimates, Yu et al. [204] combine non-linear regression with Bayesian learning to estimate the opponent's reservation value, as well as the deadline. In their model, the opponent is assumed to use a time-dependent tactic with unknown parameters. Each round the parameters are estimated using non-linear regression. Next, that round's estimate is used to create a more reliable set of hypotheses about the opponent's reservation value and deadline by using Bayesian learning.

All these methods estimate the opponent's reservation value in a single-issue negotiation using Bayesian learning (which is more computationally involved), or non-linear regression (which is faster, but requires knowledge about the structure of the opponent's strategy). To our knowledge, artificial neural networks and kernel density estimation have not been used for this purpose. Furthermore, all these methods assume that given the reservation value all acceptable bids are known due to the known ordering of the possible values. An interesting open problem is how to apply these techniques to situations where such an ordering is not straightforward.

5.1.2 Learning the acceptance strategy by estimating the acceptance probability

The acceptance strategy can be learned by keeping track of what offers were accepted in previous negotiations and by recording the offers the opponent sends out. From this information, an agent can estimate the probability that a bid will be accepted in a particular negotiation state. As it is unlikely that such an estimate can be derived for all possible bids, regression methods can be applied to determine the acceptance probability for the entire outcome space.

It is easiest to apply this method in repeated single-issue negotiations, as Saha and Sen do in [173]. In this scenario, a seller may only propose a price once, which a buyer then accepts or rejects. An increasingly better estimate of the buyer's acceptance strategy allows the seller to maximize its profit over time. To derive the set of samples, the buyer first samples the outcome space to find bids that are either always rejected, or always accepted. After that,

a number of in-between values are sampled, until the acceptance probability of a sufficient amount of bids has been determined. In order to estimate the acceptance probability of *all* possible offers, polynomial interpolation is applied, using Chebyshev polynomials [131]. Given the probability distribution of acceptance of each offer, the seller can determine the optimal price to maximize profit.

Interpolation of the acceptance likelihood does not directly carry over to a multi-issue negotiation setting, because the multi-issue preference space lacks the structure of the single-issue case with opposite preferences. The key approach in overcoming this challenge is by Oshrat et al. [149] and relies on a database of negotiations against a set of human negotiators with known preference profiles. During the negotiation, it is assumed that the opponent's preference profile is known, or that the Bayes classifier introduced in [123, 124] can be applied to reliably learn the opponent's profile. The database traces then determine what bids have been proposed or accepted by the opponent, which are pooled together under the assumption that if an agent makes an offer, it is also willing to accept it. The authors then use kernel density estimation to estimate the acceptance probability for all the other bids.

Lau et al. apply a similar method based on Bayesian learning with the addition that the effect of time pressure and possible changes of the opponent's negotiation strategy are taken into account [113]. The underlying idea is that a bid which is unacceptable for an opponent at the beginning of the negotiation might be acceptable at the end. The effect of time pressure is modeled by giving recent bids in a negotiation a higher weight. In addition, more recent negotiation traces receive a higher weight to account for possible opponent strategy changes.

Finally, Fang et al. [57] present a lesser known technique for multi-issue negotiation, which assumes that every presented bid is also acceptable for the opponent. The set of acceptable offers from earlier negotiations are used to train a simple neural network that can then test whether any particular bid is acceptable or not.

5.2 Learning the deadline

The *deadline* of a negotiation refers to the time before which an agreement must be reached to achieve an outcome better than the best alternative to a negotiated agreement [158]. Each agent can have its own private deadline, or the deadline can be shared among the agents. The deadline may be specified as a maximum number of rounds [187], or alternatively as a real-time target. Note that when the negotiation happens in real time, the time required to reach an agreement depends on the deliberation time of the agents (i.e., the amount of computation required to evaluate an offer and produce a counter offer).

When the opponent's deadline is unknown, it is of great value to learn more about it, as an agent is likely to concede strongly near the deadline to avoid non-agreement [63]. Because of this strong connection between the two, most of the procedures discussed in Sect. 5.1.1 can also be used to estimate the opponent's deadline. Hou [85] for example, estimates the deadline following the same procedure for estimating the reservation value. Yu et al. [204] apply a similar method with the additional constraint that the opponent uses a time-dependent tactic. Finally, Sim et al. directly calculate an estimate for the deadline from the estimated reservation value [72, 92, 184].

As is the case for the reservation value, these methods assume a single-issue negotiation and make strong assumptions about the opponent's strategy type. How to weaken these assumptions and estimate the deadline in multi-issue negotiations is still an open research topic.

5.3 Learning the preference profile

The *preference profile* of an agent represents the private valuation of the outcomes. To avoid exploitation, agents tend to keep their preference information private [50,206]; however, when agents have limited knowledge of the other's preferences, they may fail to reach a Pareto optimal outcome as they cannot take the opponent's desires into account [83].

In order to improve the efficiency of the negotiation and the quality of the outcome, agents can construct a model of the opponent's preferences [50,83,206]. Over time, a large number of such opponent models have been introduced, based on different learning techniques and underlying assumptions [12]. Learning the opponent's preference profile can be of great value, as it provides enough information to allow an agent to propose outcomes that are Pareto optimal and thereby increase the chance of acceptance [60,81].

Four approaches have been used to estimate the opponent's preference information. The first approach, which is discussed in Sect. 5.3.1, assumes that the opponent uses a linear additive utility function. Many of the other three approaches are applicable in negotiation settings with non-linear utility functions as well.

- *Estimation of issue preference order* (Sect. 5.3.1). The agent can estimate the importance of the issues by assuming that, as the opponent concedes, the issues it values the least are conceded first.
- *Classifying the opponent's negotiation trace* (Sect. 5.3.2). During the negotiation, the agent classifies the opponent's negotiation trace, using a finite set of groups of which the preferences are known.
- *Data mining aggregate preferences* (Sect. 5.3.3). The agent is assumed to have available a large database containing aggregate customer data. The problem of opponent modeling then essentially reduces to a data mining problem.
- *Applying logical reasoning and heuristics to derive outcome order* (Sect. 5.3.4). The agent deduces preference relations of the opponent from the opponent's negotiation trace, using common sense reasoning and heuristics.

5.3.1 Learning the preference profile by estimating the issue preference order

The *issue preference order* of an agent is the way the agent ranks the negotiated issues according to its preferences; that is, it is an ordinal preference model over the set of *issues*, rather than the full set of *outcomes*. Learning the opponent ranking of issues can be already sufficient to improve the utility of an agreement [60]. If the opponent is assumed to use a linear additive utility function (as defined in Sect. 3.3), then this reduces the problem of learning the continuous preference of all possible outcomes to learning the ranks of n issues, effectively limiting the size of the search space to $n!$ discrete possibilities. Needless to say, such an assumption might not be realistic depending on the definition of the issues and the complexity of the negotiation scenario. However, especially in situations where the number of interactions between the negotiating parties is limited, an agent can fall back on learning the opponent's issue preference order.

The learning techniques discussed in this section estimate the importance of the issues by analyzing the opponent's concessions, assuming that an opponent concedes more strongly on issues that are valued less. They all follow the same pattern: initially, each issue is assigned an initial weight. Next, each round the difference in value for each issue between the current and previous bid is mapped to an issue weight by applying a similarity measure. Finally, the estimated weights are used to update an incremental estimate.

There are multiple approaches to map the similarity between two bids to a set of issue weights. The most established one is given by Coehoorn and Jennings, who propose to use kernel density estimation [50] (a similar approach is discussed by Farag et al. [58]). The main assumption here is that the opponent uses a time-dependent, concession-based bidding strategy. Based on this assumption, the distance between sequential offers is mapped to a kernel, which represents the estimated issue weight, and the probability of the weight given the distance. The mappings are derived from previous negotiations and stored in a database. Each round, a new estimation of the issue weights is calculated by combining the current estimate with all previous estimates, using kernel density estimation (cf. Sect. 4.3). This results in a probability distribution for each issue weight.

The second most popular approach is by Jonker et al. [21,94], who first classify the issues as either predictable or unpredictable [82], and then estimate the weights of the predictable issues by using domain-dependent heuristics. When the opponent's offer is received, each issue value is first converted to an evaluation scale based on a predefined mapping per issue. For example, the value "5000" for the issue *price* and value "red" for the issue *color* are both mapped to the evaluation "good". Next, the evaluation distance between two sequential bids is calculated for each issue, and mapped to weights based on a predefined mapping. As a final step, all issue weights are normalized to obtain an estimate of the weights for every predictable issue. Carboneau and Vahidov [37] propose a similar method to estimate on which issue the opponent will likely concede. Instead of using a common evaluation scale, the difference in issue value between two sequential offers is normalized by dividing by the range of the issue.

Niemann and Lang [143] introduce an alternative method that relies on Bayesian learning. This approach assumes that the agents publicly announce their preference directions and the acceptable ranges for every issue. They solely negotiate on issues they disagree about, and thus, only the issue weights of the n issues with opposing directions need to be estimated. Note that this is rather strong assumption, effectively assuming all win-win outcomes can be achieved, even before starting the negotiation. Initially, all issue weights have the same likelihood, and the updating process proceeds as follows: given two consecutive offers of an opponent, a normalized concession ratio c_i per issue is computed that is supposed to be inversely related to the weight associated with the issue; i.e., the more an issue is conceded on, the less important it is: $w_i = 1 - c_i$. Using these estimates, the hypotheses about the weight distribution are updated using Bayesian learning.

As an alternative to these approaches, the opponent's issue weights can also be estimated using knowledge about the opponent's bidding strategy [90]. In this approach, the ranges of the negotiated issues are subdivided in a set of fuzzy membership functions, i.e.: probability distributions over a subrange of a variable. For example, a single issue can have four membership functions that are centered around 0.25, 0.50, 0.75, and 1. Next, a set of hypotheses about the issue weights is generated, which are mappings from each issue to a membership function. The set of possible hypotheses is initialized as the Cartesian product of all possible assignments, after which hypotheses with significantly low weights are removed. During the negotiation, the likelihood of the hypotheses are updated by using a predefined estimate of the lower and upper bound of the opponent's target utility for a particular negotiation round. In every round, the most likely hypothesis is used as the current estimate of the issue weights.

The quality of the models discussed above inherently depends on the quality of the mapping that is derived from domain knowledge, or the reliability of the estimate of the opponent's bidding strategy. These models are thus not suited for agents that negotiate on beforehand unknown domains against unknown opponents.

5.3.2 Learning the preference profile by classifying the negotiation trace

Learning the opponent's outcome preferences using classification consists of two steps: identifying the opponent groups and their preferences; and applying a classification algorithm to categorize the opponent. Bayesian learning is the most common way to classify the opponent, by determining which opponent type is most likely given its negotiation actions.

There are two main Bayesian classification methods. The first is given by Lin et al. [123, 124], who propose a method in which the set of possible groups are given. The preference profile of an agent is a mapping from an offer to a Luce number, which is the utility of the offer divided by sum of utilities of all possible offers. Bayesian learning is used to determine which preference profile t^i is the best match given a finite set $\{t^1, \dots, t^k\}$ of possible preference profiles. The preference profile t^i with the highest probability in a round is used as the currently estimated profile.

The second important Bayesian approach is by Hindriks and Tykhonov [83], who also enumerate all possible hypotheses, but they do so for the more complex setting of multi-issue negotiation. The exact parameters of the opponent's utility function are unknown, but it is assumed to be linear additive. The set of all possible preference profiles is generated by combining two sets of hypotheses. The first set corresponds to all possible hypotheses about the ranks of the issues. The second set is used for estimating the evaluation functions for each issue (as in Eq. (1), p. 8). Each issue evaluation function is approximated by a set of basic triangular (i.e., continuously single-peaked) functions that consist of (combinations of) linear functions. Figure 7a shows that even two basic functions can already reasonably approximate a more complex evaluation function. The complete set of hypotheses is the Cartesian product of both sets.

The next step is to update the likelihood of the hypotheses. To do so, the opponent is assumed to concede over time to its reservation value. As it is unknown what kind of bidding strategy the opponent is actually using, a probability distribution over a set of concession strategies is maintained, as illustrated in Fig. 7b. Each time a bid is presented, the estimated utility is calculated for each preference order hypothesis, and compared to the predicted utility by the strategy hypotheses. The difference in utility is used to update the likelihood of the hypotheses about the opponent's preference profiles and strategy.

Due to poor scalability of the model, this approach only works for small negotiation domains. Therefore, Hindriks and Tykhonov also propose a scalable variant in which the

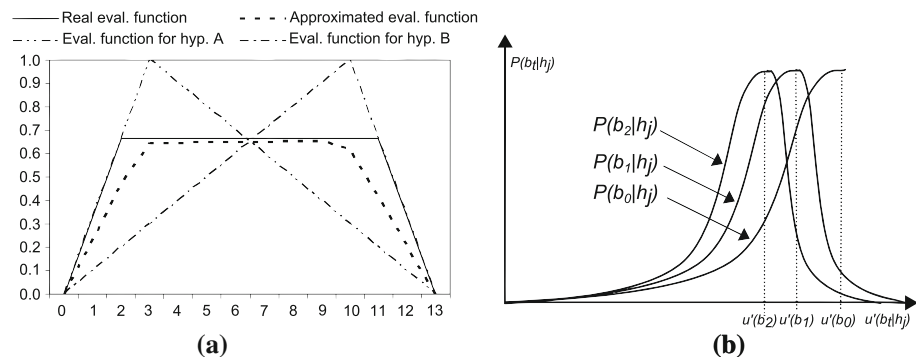


Fig. 7 Estimating the opponent's decision function using Bayesian learning [83]. **a** Combination of two basic functions. **b** Hypotheses of the opponent's decision function

additional assumption is made that the issue weights and evaluation functions can be learned separately. This particular variant is used by multiple negotiation agents [15, 51, 198] in the Automated Negotiating Agent Competition [10, 16, 200, 202].

Building upon the approach by Hindriks and Tykhonov, Rahman et al. [155] learn the opponent's preference profile using data from previous negotiations. The main difference between the two methods is that the availability of historical data is exploited to estimate the issue weights. This reduces the hypothesis space, further improving the scalability of the method.

Finally, Buffett and Spencer [31, 32] discuss a contrasting method for multi-object negotiation (a special case of multi-issue negotiation about the inclusion of items in a set), in which the possible groups are automatically enumerated. The classification method relies on the assumptions that the interaction effects between objects are minimal and that both agents use a pure concession strategy. Given the set of hypotheses about the possible preference profiles, Bayesian learning is used to determine which one is most likely to be valid. The update mechanism derives individual preference relations and uses these to update the hypotheses; e.g., when the opponent has presented a bid that contains only objects X and Y , and later on presents the offer X, Z ; then it can be assumed that Y is valued over Z .

5.3.3 Learning the preference profile by data mining aggregate preferences

Learning the opponent's outcome preference order from aggregate negotiation data is fundamentally different from classification, as it deals with a single group of agents. The agents in this group are assumed to have similar preferences, but may employ different strategies. In this setting, the challenge is to derive the opponent's preference profile from a large database of negotiation traces from similar—but not identical—opponents.

The work by Robu et al. [165, 166] has had the most impact in this area. They introduce the concept of utility graphs for complex, non-linear negotiation settings, in which the opponent's evaluation of a bundle is assumed to be the sum of the evaluation of its clusters. For example, if a bundle contains three items X, Y , and Z , then some clusters can be indicated to have interdependency between the items (for example $\{X, Y\}, \{X, Z\}$). Each cluster is assumed to have a certain evaluation for the buyer. A positive evaluation of a cluster with more than one item means that the items augment each other (a left and a right shoe), whereas a negative evaluation means that redundant items are included (two right shoes).

The learning method models the interaction effects between the items using a graph where every node is an item, and an edge between two nodes is drawn when they belong to a common cluster. Each time the opponent presents an offer, the links of the clusters corresponding to the offer are strengthened whereas the others are weakened. Selecting the best counter-offer then reduces to finding the bundle with the utility. The graph is assumed to be initialized using a reasonable approximation of the buyer's preferences. In [165], Robu and Poutré introduce a method to derive such an approximation using collaborative filtering to estimate the structure of the utility graph from a database of anonymous negotiation data.

Klos et al. [103] study the same setting in an e-commerce scenario in which interaction effects are common. A buyer and seller agent negotiate about the price of a bundle of items, which are described by a boolean vector that specifies which items are included in the offer. The buyer is a selfish agent, whereas the seller tries to take the buyer's preferences into account by recommending bundles with high social welfare and thereby minimize negotiation cost and maximize consumer satisfaction. The utility of a bundle for the buyer is defined as its valuation minus the cost; the utility for the seller is the price paid by the customer minus the production cost of the items in the bundle. The gains from trade for a bundle b is defined

as the sum of the utility of both parties, which is equal to customer's valuation minus the seller's valuation. Given this formalization, the goal of the method is to estimate the buyer's preference profile to find a Pareto improvement upon the currently negotiated bundle.

There are two main methods to learn the opponent's preferences: a simple aggregation method [103] and a method that estimates the parameters of a conjectured utility function [103, 188]. When a negotiation about a bundle ends—which happens upon acceptance or when an alternative bundle is offered—the price the customer is willing to pay for the bundle is known. The aggregation method determines the relative valuation of a bundle by comparing the price paid for similar bundles. The problem with this approach is that there are 2^n possible bundles given n items. Therefore, [103, 188] introduce the assumption that the buyer uses a particular utility function of which the parameters are treated as random variables. In addition, the learning method preprocesses the data to minimize the influence of the negotiation strategy.

Finally, Saha and Sen discuss a similar concept for a multi-issue negotiation, but for the case where the agents can use *arguments* to decide upon the negotiation context and to reach dynamic agreements [174]. The authors discuss the idea of an opponent model, but leave its implementation for future work. The idea is that all the relevant attributes of a negotiation can be modeled using a Bayesian network. The agent has an initial estimate of the Bayesian network (derived, for example, from a database of previous negotiations), which could be updated and refined during the negotiation based on the opponent's arguments.

5.3.4 Learning the preference profile by applying logical reasoning and heuristics

Modeling the opponent's preferences is even more challenging when no previous negotiations against the same or similar opponents have been conducted. In that case, the opponent's preferences need to be learned from a limited amount of data, within a limited amount of time. To do so, certain assumptions and heuristics are required to interpret the opponent's behavior. In this section we discuss several opponent models, starting with the models that make the weakest assumptions about the opponent and its preferences.

The candidate elimination algorithm only assumes that the opponent's preferences do not change during the negotiation. The algorithm is an inductive learning algorithm that can be adapted to learn the preferred bids of the opponent during a negotiation [8, 9]. In this approach, the opponent's preferences are represented as a set of acceptable offers. When the opponent sends out an offer, this is interpreted as a *positive* training instance. When a counter-offer is rejected by the opponent, this counts as a *negative* example, and general hypotheses are specialized not to cover this example anymore. As a concrete example, consider the following situation: an agent negotiates with the opponent over three different issues at the same time, and receives the offer (x_1, x_2, x_3) . Suppose the agent responds by making the counter-offer (x_1, x_2, x'_3) , proposing a different value for the third issue. If this offer is rejected, it reveals a lot of information on the opponent's preferences. Before this exchange of offers, the agent could do no better than to have the general hypothesis that any offer is acceptable for the opponent. However, the rejection of its last offer counts as a negative example, and the agent can conclude that x_3 is an important value for the opponent, and can specialize the general hypothesis to exclude x'_3 . Note that while the opponent model makes few assumptions, it is likely that only a part of the relationships between the outcomes is found.

Related to this concept, Restificar and Haddawy [163] estimate the opponent's preference profile in a negotiation over a single issue in which the agents are assumed to have conflicting preference directions. This work assumes that the negotiators use a particular type of bidding strategy that is based on the concept of an *offer/counter-offer gamble*. Such a gamble is a

decision whether to accept the opponent's offer, or to make a counter-offer and thereby risk that the negotiation will (ultimately) result in non-agreement. By making this assumption, the agent can interpret the opponent's moves to derive preference relations, for example how much the seller prefers making a counter-offer over accepting a particular bid. Similar to the approach discussed above, the authors assume that it is impossible to learn all relations and therefore, an artificial neural network is trained using the derived relations. The agent can use the network to estimate whether the opponent will accept an offer, or will take the risk of proposing an alternative offer that will potentially result in disagreement.

An important method with many possible applications is introduced in [60], where Faratin et al. propose to measure the similarity between the opponent's most recent bid and a set of bids under consideration. The idea is that the bid most similar to the opponent's previous bid has the highest chance to be accepted. The method is not descriptive in the sense that it does not define a model of the opponent's preferences as such; nevertheless, we briefly discuss the work as it borders on the scope of this survey. Faratin et al. show that applying this approach in a negotiation agent can result in mutually beneficial outcomes with a relatively higher gain. The heuristic is implemented in [111], and Lau et al. use a similar approach combined with a genetic algorithm [112, 114].

Buffett et al. combine concepts of the approaches discussed above in [30], assuming that the opponent uses a similarity maximizing strategy. The challenge in this setting lies in the fact that the opponent is not guaranteed to concede in each turn. It is assumed that the opponent's similarity function is public or at least close to the agent's function. The applied heuristic is as follows: if the opponent presents a bid with a similarity higher than all previous bids, then it is certain that the opponent utility of this bid is lower than all previous bids, or else it would have been offered earlier. For the remaining relations, a probabilistic approach is introduced that estimates the round the other offers could likely have been considered for the first time. Combined, this results in a set of estimated preference relations between the presented offers. Note that the approach only learns the preference relations between the bids presented by the opponent and will therefore generally not result in a complete estimate of the opponent's preference profile.

The approaches discussed above focus on identifying a subset of the preference relations between all possible bids. If an agent needs to estimate the entire preference profile, stronger assumptions are required. A popular example is the frequency analysis heuristic [75, 194, 195], which is a relatively simple technique to estimate the opponent's preference profile in a multi-issue negotiation by keeping track of how often values occur in the opponent's offers. Together with Bayesian learning techniques (Sect. 5.3.2), frequency analysis is one of the most popular preference profiling technique used by the participants of the Automated Negotiating Agent Competition [10, 16, 200, 202]. It is an attractive technique especially in large outcome spaces, in which scalable learning methods are required. The main idea is that preferred values of an issue are offered relatively more often in a negotiation trace. For the issue weights, it works the other way around: if an issue changes value often, it is probably relatively unimportant to the opponent. In [194, 195], both the set of issue weights and value weights are estimated, while Hao and Leung [75–77] ignore the issue weights completely.

5.4 Learning the bidding strategy

A negotiating agent employs a *negotiation strategy* to determine its offer in a given negotiation state. The mapping function may range from a simple time-dependent function to a complex function that dynamically depends on the opponent's behavior.

Research on agent negotiators has given rise to a broad variety of bidding strategies that have been established both in the literature and in implementations [47, 59, 60, 86, 95, 125]. Examples of such general agent negotiators in the literature include, among others: Zeng and Sycara [206], who introduce a generic agent called *Bazaar*; Faratin et al. [60], who propose an agent that is able to make trade-offs in negotiations and is motivated by maximizing the joint utility of the outcome; Karp et al. [96], who take a game-theoretic view and propose a negotiation strategy based on game-trees; Jonker et al. [95], who propose a concession oriented strategy called *ABMP*; and Lin et al. [124], who propose an agent negotiator called *QOAgent*.

Learning the opponent's bidding strategy is clearly advantageous to a negotiating agent, as this would—in theory—allow an agent to exploit the bidding behavior of the opponent to reach the best possible deal. Learning the bidding strategy, however, is very challenging as there is a wide diversity of possible negotiation strategies. And worse: the opponent may change its behavior according to the offers that an agent makes [14]. That is, learning the opponent's strategy is a *moving target problem*, where the agent simultaneously attempts to acquire new knowledge about the opponent while optimizing its decisions based on what is currently known.

Two approaches have been followed to estimate the opponent's bidding strategy:

- *Regression analysis* (Sect. 5.4.1) If an outline of the opponent's strategy is known in the form of a formula with unknown parameters, then the problem of estimating the opponent's bidding strategy reduces to regression analysis.
- *Time series forecasting* (Sect. 5.4.2) On the other hand, if the opponent's strategy is unknown, time series forecasting can be applied to predict the opponent's future offers.

Both methods have been used rather extensively, and we will cover each of them in the sections below.

5.4.1 Learning the bidding strategy using regression analysis

An agent is generally unaware of the opponent's exact negotiation strategy, but might have knowledge about the type of strategy used. If such knowledge is available and can be captured in a formula with unknown parameters, the opponent's strategy can be estimated by applying regression analysis.

There are two main approaches to this problem. The first is given by Mudgal and Vassileva [139], who employ probabilistic influence diagrams to predict the opponent's counteraction to an offer in a single-issue negotiation. Bayesian learning is used to update a probabilistic influence diagram of the opponent, which yields the probability distribution for the next opponent's action only, so this can be viewed as a one-step regression method.

Another key approach is by Hou [85], who introduces a method to estimate the opponent's strategy in a single-issue negotiation assuming that a standard tactic dependent on time, behavior, or resources is used (as discussed by Faratin et al. [59]). Hou derives a model for the time-dependent and resource-dependent strategies and use a non-linear regression to estimate their parameters. The opponent is estimated to use the best matching model, except when the error is higher than a threshold, in which case the opponent is assumed to use a behavior-dependent tactic. Following a similar approach, Agrawal and Chari estimate the opponent's decision function as an exponential function [1]; and Haberland et al. [73], Ren and Zhang [161], and Yu et al. [204] present methods to estimate the decision function when the opponent employs a time-dependent tactic.

Brzostowski et al. [28] introduce a more general method than Hou to predict the opponent's bidding strategy, by applying non-linear regression to estimate the parameters of four complex models that mix time- and behavior-based components. The utility gained by using the model is significantly higher than their earlier method, which uses derivatives to estimate the opponent's strategy [29] (we will discuss this further in Sect. 5.4.2).

With more focus on application, Papaioannou et al. compare the performance of multiple estimators in predicting the opponent's bidding strategy [151, 153]. The estimators are used to predict the opponent's decision function, which is then used to determine which offer should be proposed in the final round to avoid non-agreement. The setting is a single-issue bilateral negotiation where a client and provider exchange offers in turn. Three parameter estimation methods are evaluated: one is based on polynomial interpolation using cubic splines; another uses 7th degree polynomial interpolation; the third is a genetic algorithm that evolves the parameters of a polynomial function. All methods significantly improve the acceptance ratio of the negotiation.

5.4.2 Learning the bidding strategy using time series forecasting

When little is known about the general structure of the opponent's bidding strategy, time series forecasting is a viable alternative to the regression-based methods described above. A time series is simply a set of observations that is sequentially ordered in time. In the context of negotiation, the time series typically consists of the utilities of offers received from the opponent, but causally related series can also be used (e.g., perceived cooperativeness of the opponent over time). Learning the opponent's bidding strategy then boils down to creating a forecast of the time series, using a set of statistical techniques and smoothing methods.

We identified four ways to do so: using *neural networks*, *derivatives*, *signal processing* methods, and *Markov chains*.

Artificial neural networks The most frequently used method to predict the opponent's offers is to represent the opponent's decision function by an artificial neural network. The network is first trained using a large database of previous negotiation exchanges and is then used to predict the next bid. Neural networks are very powerful and can be used to approximate complex functions; however, studying their structure will not in general give any additional insights in the function being approximated.

Oprea [146] was one of the first to demonstrate the potential of using neural networks to predict the opponent's future offers in bilateral negotiations. The approach focuses on single-issue negotiations and only takes one of the negotiation sides into account. The input neurons are the values for the last five opponent's bids, which means that the agent's own offers are assumed to have no influence on the opponent's behavior.

To predict the offers of a *human* negotiator, Carboneau et al. [35] use an artificial neural network in a specific domain consisting of four issues. They extend their approach in [36], in which all possible pairs of issues are allowed to serve as input to the neural network. While this significantly complicates the structure of the neural network, this allows to find patterns between issues. This also facilitates training an opponent model on a particular scenario and makes it easy to apply it to scenarios where one of the issues is removed from the negotiation domain. A similar approach is discussed by Lee and Ou-Yang [115], who are even able to predict the *value* for each issue; for this, four output neurons are created, each returning the value for one of the four issues. Despite that the model was trained using data derived from other opponents, the authors find a positive correlation between the actual and predicted values.

For the general multi-issue case, *multilayer perceptrons* (MLPs) can be used, as is done by Masvroula et al. [133]. MLPs are artificial neural networks where some nodes have a nonlinear activation function. Masvroula et al. test two networks in an experimental setting: a network in which each issue is approximated by a separate MLP, and a network where a single MLP is used for all issues. The amount of input neurons and hidden layer neurons are empirically determined for both networks, after which they are shown to reliably predict the opponent's next offer, with the single MLP network resulting in the lowest mean error. In more recent work [132], Masvroula investigates the performance of two artificial neural networks that learn the opponent's strategy without relying on historical knowledge. The first model is a simple MLP that is retrained every round, using the complete negotiation trace of the opponent. The second one is more advanced (and outperforms the first one), as the structure of the neural network is optimized in every round, using a genetic algorithm that rates neural networks based on their complexity and prediction error.

The methods above do not explicitly constrain the opponent's strategy. If it is known that the opponent employs a time-dependent tactic, work by Rau et al. and Papaioannou et al. [151, 153, 160] can be used. Owing to the reduced search space of time-dependent tactics, Rau et al. [160] find that the concession tactic and weight of every issue offered by the opponent can be learned from this process in an exact manner. Papaioannou compares the performance of five estimators for the opponent's bidding strategy, of which we have already discussed three regression-based estimators in Sect. 5.4.1. Of the remaining two estimators, one is based on a multi-layer perceptron neural network, and the other uses a radial basis function neural network [151, 153]. The latter estimator outperforms all other estimators when it comes to predicting the opponent's future offers. It achieves the lowest overall error and its application results in the the largest number of successful negotiations.

Derivatives Derivatives of the concession function of the opponent is another way to reveal a lot of information on the type of decision function used and to what extent the opponent responds to different negotiation factors. In a single-issue negotiation, given a sequence of opponent offers $b_i \in [0, 1]$, the derivatives of order k are defined as follows :

$$\Delta^1 b_i = b_{i+1} - b_i, \quad (6)$$

$$\Delta^{k+1} b_i = \Delta^k b_{i+1} - \Delta^k b_i. \quad (7)$$

Brzostowski et al. [29] argue that there are two main factors contributing to the overall behavior of a negotiating agent: a time-dependent and a behavior-dependent component. They make a prediction for both components and combine the results to estimate the opponent's behavior. The two predictions are combined based on two observational measures: the *time influence metric* and the *behavior influence metric*.

For the calculation of the *time influence metric*, it is assumed that a time-dependent tactic may be modeled by using a polynomial or an exponential function. If the opponent uses a pure time-dependent tactic, all derivatives should have the same sign:

$$\forall_{i,j,k} \operatorname{sgn}(\Delta^k b_i) = \operatorname{sgn}(\Delta^k b_j). \quad (8)$$

For each k th order derivative, Brzostowski et al. calculate how strongly the time-dependent assumption holds by comparing the amount of positive and negative signs within an order. Given a k th order derivative, if all the differences are either positive or negative, the time-dependent criterion is fully satisfied, otherwise the time-dependent assumption is violated. The results of all derivatives can be combined to create the time influence metric.

The *behavior influence metric* measures to what extent the opponent responds to the agent's actions. For each round i , the agent's own concession $\Delta^1 a_i$ is compared with the opponent's concession $\Delta^1 b_i$, resulting in a metric r_i for each round:

$$r_i = \frac{\Delta^1 a_i}{\Delta^1 b_i}. \quad (9)$$

The results for each round are then aggregated in a weighted sum called r . If r is equal to one, then the opponent makes concessions of the same size. For $r < 1$, the opponent is competitive, and if $r > 1$, the opponent is cooperative. This metric can be combined with a measure for the monotonicity of the sequence r_i to create the behavior influence metric, analogous to the time-dependent metric.

With this information, two predictions can be made of the future behavior of the opponent and then combined to yield the final forecast. The behavior-dependent prediction uses extrapolation of the behavior influence metric to predict the opponent's next offer. The second prediction is solely based on time and uses the negotiation history to determine the opponent's next offer. The prediction is based on the concavity and convexity of the opponent's concession curve as measured by the differentials. The opponent's time-dependent behavior can then be approximated by polynomials to make a prediction of the opponent's future offers. Again, if it is known the opponent uses a time-dependent strategy, more specific methods can be used to approximate the opponents concession curve, such as the derivative-based approach by Mok and Sundarray [137].

Signal processing A third way to forecast the opponent's offers is to employ techniques used in *signal processing*. This type of modeling technique has recently attracted attention from a number a negotiation researchers, and three main methods have been developed since 2010.

The first main approach is to use a *Gaussian process* to predict the opponent's decision function. During the negotiation, the opponent's bidding history is recorded as a series of ordered pairs of time and observed utility. Next, a Gaussian process regression technique is used to determine a Gaussian distribution of expected utility for each time step. Williams et al. [196,197] use this technique to estimate the optimal concession rate in a multi-issue negotiation with time-based discounts. Their approach can handle a wider range of scenarios compared to the derivation-based methods discussed above, because the opponent tactic can be more complicated than a weighted combination of time- and behavior-dependent tactics. To counter noise, only a small number of time-windows are sampled, from which only the maximum utility offered by the opponent is used to make the predictions. The strategy was implemented in the *IAMhaggler2011* [199] agent, which finished third in ANAC 2011 [10]. The agent performed much better than the others on large domains, however only performed averagely on small domains.

Chen and Weiss [44,46] also predict the opponent's preferences to determine the agent's optimal concession rate. Similar to Williams et al., the maximum offered utility in a set of time windows is recorded. This time, *discrete wavelet transformation* is used to decompose the signal in two parts: an approximation and a detail part. The idea is that the first captures the trend of the signal, whereas the latter contains the noise, which is therefore omitted. After an initial smoothing procedure, cubic spline interpolation is used to make a prediction for future time windows. The end result is a smooth function that indicates the maximum utility that can be expected in the future.

An alternative method employed by Chen and Weiss relies on *empirical mode decomposition* and *autoregressive moving averaging* [45]. The same procedure is used to sample

the opponent's decision function, but now, empirical mode decomposition is used to decompose the sampled signal into a finite set of components, after which autoregressive moving averaging is used to predict the future behavior of each of these components.

Finally, when the opponent is expected to change its strategy over time without signaling this explicitly to the agent, the work by Ozonat and Singhal [150] can be used to estimate the opponent's strategy in a multi-issue negotiation. Using switching linear dynamical systems, a technique commonly used in signal processing literature to model dynamical phenomena, the opponent's decision function is predicted in terms of what utility the agent can expect in the future.

Markov chains The final time series forecasting method relies on *Markov chains*. The idea is that the set of opponent strategies is known; however, it is undisclosed when the opponent changes its strategy. The set of strategies make up the states of a Markov chain, where the transition matrix represents the probability of going from one state—a strategy—to another. Narayanan and Jennings [140] use this method to model the opponent's strategy in a single issue negotiation and apply Bayesian learning to estimate the transition matrix, where each hypothesis presents a possible transition matrix. The hypotheses are updated each round using the received opponent's offers, and then used to derive a counter-strategy.

6 Measuring the quality of an opponent model

In the previous section, we provided an overview of several learning methods for each opponent attribute. A natural question with regard to agent design is: which of the depicted models is best for each attribute? Unfortunately, it is impossible to provide a conclusive answer to this, as most authors evaluated their opponent model in their own setting, and relative to their own baseline. A valuable direction for future work is therefore to compare the quality of these models in a common setting, or in any case, to use the existing models as baselines when designing new learning techniques.

However, even if we fix a common negotiation setting for every model, quantifying the quality of an opponent model is not straightforward, as a large number of different *quality measures* are being used, each with their own advantages and shortcomings, which impedes a fair comparison of different approaches. In this section, we provide an overview of the different types of measures found in the literature. To do so, we surveyed the most popular quality measures currently in use, and we show how they relate to the main benefits of opponent modeling.

In general, we found that the quality of an opponent model can be measured in two main ways: *accuracy measures* (Sect. 6.1), which measure how closely the opponent model resembles reality; and *performance measures* (Sect. 6.2), which measure the performance gain when a negotiation strategy is supplemented by a model. We provide an overview in Tables 3 and 4 of both types of quality measures. Moreover, based on this overview, we recommend in Sect. 6.3 which measures to use to evaluate an opponent model given a specific modeling aim.

6.1 Accuracy measures for opponent models

Accuracy measures are direct measures of opponent model quality, as they quantify the difference between the *estimate* and the *estimated*. We found accuracy measures for preference modeling methods (Sect. 5.3) and for strategy prediction methods (Sect. 5.4).

Table 3 Overview of accuracy measures used in the surveyed work

Accuracy measures	
Similarity between issue weights	[12,84,90]
Similarity between bidding strategies	[35–37,44,45,115,132,133,137,151,153,197]
Similarity between preference profiles	[12,31,32,84,163,165]

To the best of our knowledge there are no accuracy measures for the acceptance strategy and deadline, although some existing measures can be used for these instances as well. We will come back to this when we discuss the other accuracy measures below. Table 3 provides an overview of metrics found in the surveyed work.

Similarity between issue weights We can measure the accuracy of models that estimate the issue weights of the opponent's preference profile (assuming linear additive utility functions, see Sect. 5.3.1) in several ways. All of them use a distance metric between the issue weights $w = (w_1, \dots, w_n)$ of the real opponent preferences u_{op} and the issue weights $w' = (w'_1, \dots, w'_n)$ of the estimated preferences u'_{op} . One way to do so is to measure the distance between the issue weight vectors [90]:

$$d_{\text{Euclidean}}(w, w') = \sqrt{\sum_{i=1}^n (w_i - w'_i)^2}. \quad (10)$$

Of course, this measure can be used for scalars as well. When modeling the opponent's deadline (or reservation value) $x \in \mathbb{R}$ with an estimate x' , Eq. (10) simplifies to $d_{\text{Euclidean}}(x, x') = |x - x'|$.

Another way is to check whether the issue weights are *ranked* correctly [84] by evaluating all possible pairs of issues i_1, \dots, i_n :

$$d_{\text{rank}}(w, w') = \frac{1}{n^2} \sum_{j=1}^n \sum_{k=1}^n c_{w,w'}(i_k, i_j), \quad (11)$$

where $c_{w,w'}$ is the conflict indicator function, which is equal to one when the ranking of the weights of issues i_k and i_j differs between the two profiles, and zero otherwise. This measure is particularly useful when the utility values are of less concern than their relative importance. An alternative is to measure the *correlation* between the weight vectors [84]:

$$d_{\text{Pearson}}(w, w') = \frac{\sum_{i=1}^n (w_i - \bar{w})(w'_i - \bar{w}')}{\sqrt{\sum_{i=1}^n (w_i - \bar{w})^2 \sum_{i=1}^n (w'_i - \bar{w}')^2}}. \quad (12)$$

Note that this expression may be undefined, for example when all weights are equal.

Similarity between bidding strategies For opponent models that estimate the opponent's bidding strategy (Sect. 5.4), the accuracy of the model can be determined by comparing two equally sized vectors of length N , where one vector x denotes the actual behavior of

the opponent, and the other vector x' is the predicted behavior; e.g., the utilities of received offers at every time slot, or the values the opponent chooses for a particular issue. One way to measure the similarity between the two vectors is to calculate the mean squared error [115, 137]:

$$d_{\text{MSE}}(x, x') = \frac{1}{N} \sum_{i=1}^N (x_i - x'_i)^2. \quad (13)$$

A possible disadvantage of this metric is that it may overemphasize outliers. In this case, the mean absolute error is a viable alternative [115]:

$$d_{\text{MAE}}(x, x') = \frac{1}{N} \sum_{i=1}^N |x_i - x'_i|. \quad (14)$$

Both metrics quantify the error as a positive value, which has the disadvantage that the sign is lost: in some settings a positive error may be worse than a negative error. A metric taking this into account is the percentage of error [115], which is normally calculated separately for every element:

$$d_{\text{PCE}}(x_i, x'_i) = \frac{x'_i}{x_i} - 1. \quad (15)$$

Again, the exact value of the measure may not matter; e.g., when the negotiation agent tries to predict if the opponent will concede. In this case the correlation between the two vectors can be measured, for example by using Eq. (12) [115, 132, 133].

Similarity between preference profiles When opponent models estimate the opponent's preferences fully (as described in Sect. 5.3), the quality of these models depends on the similarity between the real opponent's preference profile u_{op} and the estimated profile u'_{op} for all bids ω in the outcome space Ω . Suppose the opponent uses a utility function $u_{op}(\omega)$ to calculate its utility for bid ω , then we define the opponent model's estimate of this function as $u'_{op}(\omega)$.

One approach is to calculate the absolute difference between all outcomes in Ω [12]:

$$d_{\text{abs}}(u_{op}, u'_{op}) = \frac{1}{|\Omega|} \sum_{\omega \in \Omega} |u_{op}(\omega) - u'_{op}(\omega)|. \quad (16)$$

A shortcoming of this approach is that the result is not invariant to scaling and translation. An alternative is to use the *ranking distance of bids* [84] measure that compares all preference orderings in a pairwise fashion, which is especially useful when ordinal preferences are involved:

$$d_{\text{rank}}(u_{op}, u'_{op}) = \frac{1}{|\Omega|^2} \sum_{\omega \in \Omega, \omega' \in \Omega} c_{<u, <u'}(u_{op}(\omega), u'_{op}(\omega')), \quad (17)$$

where $c_{<u, <u'}$ is the conflict indicator function, which is equal to one when the ranking of the outcomes ω and ω' differs between the two profiles, and zero otherwise. Identically, Buffett et al. count the amount of correctly estimated preference relations [31, 32]. A disadvantage of these approaches is their scalability because all possible outcome pairs need to be compared. This problem can be overcome by using a Monte Carlo simulation; however, a more efficient solution can be to use the *Pearson correlation of bids* [84], which is defined as follows:

$$d_{\text{Pearson}}(u_{op}, u'_{op}) = \frac{\sum_{\omega \in \Omega} (u_{op}(\omega) - \overline{u_{op}})(u'_{op}(\omega) - \overline{u'_{op}})}{\sqrt{\sum_{\omega \in \Omega} (u_{op}(\omega) - \overline{u_{op}})^2 \sum_{\omega \in \Omega} (u'_{op}(\omega) - \overline{u'_{op}})^2}}. \quad (18)$$

In this formula $\overline{u_{op}}$ is the average utility of all possible outcomes for the opponent, and $\overline{u'_{op}}$ the estimated average utility. As for the weight vectors, a downside of this measure, although unlikely to occur in practice, is that it is not defined for all inputs, for example when the opponent's utility is estimated to be constant.

All preference profiles metrics can also be applied to assess the quality of acceptance strategy models. After each negotiation round, the opponent's acceptance strategy can be asked to provide the acceptance probability for a set of bids. Next, for these bids the actual and predicted acceptance probability can be compared using one of the metrics above.

6.2 Performance measures for negotiation strategies

The ultimate aim of employing an opponent model is to increase overall performance of the negotiation, which is why performance measures are the most popular quality measure. The most popular way is to measure the gain in utility of the outcomes due to the usage of an opponent modeling technique. Other measures that the agent designer might choose are the duration of the negotiation (i.e., how fast the agent is able to reach agreements), or fairness of the outcome (i.e., whether the agreement satisfies all negotiation parties).

Sometimes performance measures can be incorporated directly into the utility function, as is the case for discounted utility through time. However, it is usually advisable to have multiple independent performance measures available, especially when the designer wishes to assess several aspects of the negotiation outcome.

To measure the quality of an opponent model, the model can be applied by a set of agents that compete against various opponents on a number of negotiation domains. Ideally, the opponent model is tested in combination with multiple negotiation strategies to minimize the influence of how the model is applied by the strategy. Note that the measurements strongly depend on the negotiation setting [10], which therefore should be chosen with care: an opponent model may appear to be of low quality when its assumptions are not satisfied. This effect can be minimized by testing the model in a large and balanced set of negotiation settings, as discussed in [11, 12]. Furthermore, as performance measures only consider the quality of the outcome, we recommend to also include the accuracy measures of Sect. 6.1 for benchmarking purposes.

Table 4 provides an overview of the types of performance measures we found in existing work.

Average utility Average utility is by far the most popular performance measure. A common application is to consider the average utility of an agent with and without an opponent model against a group of opponents on several domains (see for example [45, 75, 143]).

Distance to a fair outcome Similar to the average utility, a large number of authors are concerned with achieving a fair outcome. Fairness is an important aspect, especially when the parties expect to conduct future negotiations. The Nash solution ω_{Nash} in a bilateral negotiation is defined as the outcome that maximizes the product of the utilities:

$$\omega_{\text{Nash}} = \max_{\omega \in \Omega} u_a(\omega) \cdot u_b(\omega). \quad (19)$$

Table 4 Overview of performance measures used in the surveyed work

Performance measures	
Average utility	[11, 12, 28, 29, 44, 45, 50, 51, 57, 58, 72, 73, 75, 81–83, 85, 92, 94, 123, 124, 133, 140, 143, 149–151, 155, 173, 183, 184, 188, 194, 197, 204, 207]
Distance to a fair outcome	[11, 81, 82, 137]
Distance to Pareto frontier	[11, 113, 143, 155, 160]
Joint utility	[50, 90, 103, 113, 123, 124, 137, 146, 149, 160, 183, 184, 205–207]
Percentage of agreements	[1, 9, 30, 72, 73, 90, 103, 139, 143, 146, 149, 151, 153, 160, 165, 166, 183, 184, 188, 204]
Robustness	[10, 43]
Time of agreement	[7–9, 11, 50, 72, 103, 113, 137, 146, 149, 155, 165, 166, 183, 184, 188, 205–207]
Trajectory analysis	[81, 82, 155]

If u_{\max_a} and u_{\max_b} are the highest possible utilities achievable by agent a and b in a negotiation, then the Kalai-Smorodinsky is defined as the Pareto optimal bid ω for which:

$$\frac{u_{\max_a}}{u_{\max_b}} = \frac{u_a(\omega)}{u_b(\omega)}. \quad (20)$$

Distance to a fair outcome is then calculated as the average distance to a fair solution, such as the Nash solution [11, 81, 82, 137] or Kalai-Smorodinsky [11, 81, 82].

Distance to Pareto frontier An opponent model of the opponent's preferences aids in identifying Pareto optimal bids. For this type of model—assuming it is applied by a bidding strategy that takes the opponent's utility into account—the distance to the nearest Pareto optimal bid directly correlates with the model's quality (see for example [11, 143, 155]). Minimizing this distance to the Pareto-optimal frontier improves fairness and the probability of acceptance.

Joint utility An alternative method to measure the fairness of an outcome is to calculate the joint utility. The majority of the authors simply use the sum of the utility of the final outcome for the agents (see for example [123, 124]). An alternative used by several authors [146, 205–207] is to consider the normalized joint utility:

$$u_{\text{joint}} = \frac{(P - R_s)(R_b - P)}{(R_b - R_s)^2}. \quad (21)$$

In this equation, P is the agreed upon price, and R_b and R_s are the reservation prices of the buyer and seller respectively. Note that this definition is only applicable to single-issue negotiations. An alternative measure for multi-issue negotiations used by Jazayeriy et al. [90] is the geometric mean:

$$u_{\text{joint}} = \sqrt{u_a \cdot u_b}, \quad (22)$$

where u_a and u_b are the utilities achieved by the agents. An attractive property of this metric is that when the utilities are highly unbalanced, this formula better reflects unfairness than by simply calculating the sum of the utilities.

Percentage of agreements An opponent model may lead to better bids being offered to the opponent, possibly avoiding non-agreement. In situations where an agreement is always better than no agreement, the percentage of agreements is a direct measure of success (see

for example [9,30,72,73]). An important disadvantage is that the acceptance ratio does not capture the quality of the agreement, thus it is advised to also measure the average utility.

Buffett et al. Mudgal and Vassileva, and Agrawal and Chari use a related measure in which they calculate how often one agent outperforms the other with regard to the final outcome [1,30,139]. A disadvantage of this method is that an agent might outperform other agents, but still reach a bad outcome. An alternative metric is applied by Robu and Poutré [165,166], which calculates how often an outcome is reached that maximizes social welfare.

Robustness Many of the performance measures only give a fairly narrow view of the performance of the agents, as they do not consider the interactions *between* different strategies. For instance, an agent may be exploitable by opponent strategies that were not expected in the design phase, requiring a switch to a different strategy [10]. To make this notion precise, game theory techniques can be combined with evolutionary modeling [10,43]—referred to as *evolutionary game theory* (EGT)—to measure the *robustness* of negotiation strategies. EGT is used to measure the distribution of negotiation strategies evolving over time, assuming that the players may switch their agent's strategy based on the pay-off matrix to maximize their utility against their opponents. The authors show that some agents that work well in a static setup perform poorly in an open environment in which players can change strategy.

Time of agreement Various authors measure the duration of the negotiation (e.g., [103,149,155]), or the communication load, because in practical settings there is often a non-negligible cost associated with both. Opponent models can lead to earlier agreements, and thereby reduce costs. An important disadvantage of this metric is that while an opponent model may lead to an earlier agreement, the quality of the outcome for the agent might be lower.

Trajectory analysis The quality of bidding strategies can be measured by analyzing the percentage and relative frequency of certain types of moves [81]. For example, unfortunate moves are offers that decrease the utility for both agents at the same time. Theoretically, a perfect opponent model of the opponent's preferences would allow an agent to prevent any such unfortunate moves. A disadvantage of this method is that it highly depends on the concession strategy that is used in combination with the opponent model.

6.3 Benchmarking opponent models

As a starting point towards quantifying the quality of existing opponent models, and thereby the transition from theoretical agents to practical negotiation agents, this section provides guidelines on how to select the appropriate quality measures for a given opponent model. These guidelines are based on the overview provided above, and our previous work on quantifying the quality of opponent models that estimate the opponents preference profile [11,12]. We argue that a benchmark should consist of three components: a set of accuracy measures, a set of performance measures, and a fair tournament setup.

Both types of measures should be included, as a high accuracy demonstrates the approach is successful in its own right, independent of any other factors, while a good performance shows that the model is correctly applied, and fits into the agent design as a whole.

The construction of a fair negotiation setup is a challenge in its own right and worthy of a survey of its own. The negotiation strategies and negotiation scenarios of an experimental setup should be carefully selected to not favor particular types of modeling approaches. On the other hand, the setup should be representative of what can be expected in practical negotiation. Another challenge is that a balanced tournament requires access to a large set of

Table 5 An overview of performance measures applied in the literature and how they relate to opponent modeling aims

Aim	Performance measure	References to applications in existing work
Minimize negotiation cost	Average utility	[151]
	Percentage of agreements	[72, 73, 90, 103, 143, 146, 149, 151, 153, 160, 183, 184]
	Time of agreement	[7–9, 11, 50, 72, 103, 113, 137, 146, 149, 155, 165, 166, 184, 188, 205–207]
Adapt to the opponent	Average utility	[28, 29, 44, 45, 57, 72, 73, 75, 81, 85, 92, 133, 140, 150, 155, 173, 197]
	Percentage of agreements	[1, 139, 204]
	Robustness	[10, 43]
	Trajectory analysis	[81, 155]
Reach win–win agreements	Average utility	[11, 12, 50, 51, 81, 83, 94, 123, 124, 143, 149, 155, 183, 184, 188, 194, 207]
	Distance to Pareto frontier	[11, 113, 143, 155, 160]
	Distance to a fair outcome	[11, 137]
	Percentage of agreements	[30, 166, 188]
	Joint utility	[50, 90, 103, 113, 123, 124, 137, 146, 149, 160, 183, 184, 205–207]

negotiation strategies and scenarios designed for an identical negotiation setting. We refer to [11, 12], which discuss benchmarks for quantifying the performance and accuracy of a large set of preference models.

In Sect. 5, we distinguished four types of opponent models: models for the *acceptance strategy*, *deadline*, *preference profile*, and *bidding strategy*. Below, we provide accuracy measure recommendations for each one of them.

1. **Acceptance strategy** For models that estimate the reservation value (Sect. 5.1.1) we recommend to use the *percentage of error* (Eq. (15), p. 34) as it quantifies the signed distance, which for this type of models is especially important; a reservation value which is estimated too low may lead to non-agreement. For the models discussed in (Sect. 5.1.2) that estimate the acceptance probability for every bid in the outcome space, distance metrics such as the *ranking distance of bids* (Eq. (17), p. 34) or the more scalable *Pearson correlation of bids* (Eq. (18), p. 34) are most suitable, assuming it is possible to request the opponent's acceptance probability of every bid.
2. **Deadline** There is a close relation between the reservation value and the deadline as they are both scalars that become more easy to estimate near the end of a negotiation. For deadlines, it is particularly important to not overestimate the actual deadline, as this can lead to non-agreement, and hence decreased performance. Therefore, similar to the reservation value, we recommend to use the *percentage of error* (Eq. (15), p. 34).
3. **Preference profile** We discussed four types of approaches to estimate the preference profile. Some assess only the issue weights, others model a larger part of the preference

relations, up to the complete preference profile. For the issue weights (Sect. 5.3.1), we suggest to use the *Euclidean distance* between the actual and estimate issue weights (Eq. (10), p. 33), as in most cases, it is crucial to get them exactly right, and it is not sufficient if they are merely correlated with the real weights, or in the right order. However, for larger subspaces of the preference profile, we can do with less precision. For the other approaches, we propose to use the *ranked distance of bids* (Eq. (17), p. 34) or the *Pearson correlation of bids* (Eq. (18), p. 34), especially when computational performance is an issue.

4. **Bidding strategy** All bidding strategy models produce an estimate of the bids that will be offered, or of the utility of these bids. The distance between the actual and predicted utility is best expressed as a single value using the *mean squared error* [115, 137] (Eq. (13), p. 33). In contrast to the *mean absolute error* (Eq. (14), p. 34), this metric puts more emphasize on outliers, which we believe is important as this type of model is often used to exploit the opponent. We do not recommend to use the *percentage of error* (Eq. (15), p. 34) as it is not straightforward how to aggregate the data, and does not allow for easy comparison of models.

While the selection of an accuracy measure solely depends on the specific type of the opponent model, the right choice for a performance measure depends on the aim of the agent designer. Therefore, Table 5 gives an overview of how existing work has assessed the performance of opponent models, together with the associated aim(s) with which the models have been applied. Depending on the aim, we recommend the following measures:

1. **Minimize negotiation cost** The negotiation cost is dependent on the time passed before reaching an agreement and the percentage of successful negotiations. The straightforward measure to use for this aim is the *time of agreement* (measured in terms of rounds or real time passed) when the utility is discounted or when negotiation rounds incur a cost. When there are multiple negotiations, the *percentage of agreements* needs to be taken into account. To augment these measures we recommend to also measure the *average utility* to ensure that negotiation performance is not sacrificed to minimize costs.
2. **Adapt to the opponent** By adapting, an agent firstly aims to optimize its *average utility*. In addition to this metric, we recommend to use *trajectory analysis* to get a better insight into the application of the opponent model by the agent, and how it is influenced by the opponent. Assuming a multitude of possible opponent strategies, the *robustness* of the strategy could be evaluated using *evolutionary game theory* to validate that conversely, opponent adaptation does not lead to exploitation of the agent.
3. **Reach win-win agreements** The quality of a win-win agreement can be expressed in many ways, from its *distance to the Pareto frontier* to its *joint utility*. However, many of the important characteristics of a win-win agreement are already captured by measuring the *distance to a fair outcome*; therefore, we propose this measure to quantify win-win solutions.

7 Conclusion

The research field of opponent modeling in negotiation is constantly evolving, driven by more than twenty years of interest in automated negotiation and negotiation agent design in particular. In this work, we survey opponent modeling techniques used in bilateral negotiations, and we discuss all possible ways in which opponent models are employed to benefit negotiation agents. There are essentially two main opponent model categories: models that

learn *what the opponent wants*, in terms of its reservation value, deadline, and preference order, and secondly, models that learn *what the opponent will do*, in terms of the opponent's bidding strategy and acceptance strategy. In this comprehensive survey, we create a taxonomy of currently existing opponent models, based on what opponent attributes are learned, which specific methods are used to model the attribute, and finally, what general learning techniques are applied in each case.

There exists a clear relation between every opponent attribute and the corresponding learning techniques. Bayesian learning is the most popular technique, and can be applied to learn any of the four opponent attributes we distinguish in our exposition. Time series forecasting and regression techniques are the second most popular techniques, and can be used whenever a trend can be established in the opponent's behavior, which holds true for almost all learning tasks, except preference profile estimation. Other techniques, such as reinforcement learning, have not been used so far, but this might be only a matter of time. Learning what the opponent will *do* is even more challenging than finding out what the opponent *wants*, because the former depends on the latter. This is why the most advanced machine learning techniques are required for this case, ranging from artificial neural networks to signal processing techniques.

For each type of opponent attribute, there are a large number of different learning methods to choose from. Generally, it is unfeasible to compare these models with each other, as most authors evaluate them in their own specific negotiation setting. This is no surprise, given the fact that no universal model of negotiation has been adopted yet, let alone a common method to compare different learning techniques. This seems to call for a *negotiation benchmark* that could reliably compare different approaches that have been taken so far and those that will emerge in the future. As a first step, an additional contribution of this work is that we discuss all performance and accuracy metrics that are used to evaluate the quality of opponent models, and how each metric helps to quantify the opponent model benefits we have outlined in this survey. Consistently applying these measures would greatly improve comparability of results, and would provide insight in possible improvements to existing models and how they can be combined to augment each other.

The many different approaches and assumptions of the opponent modeling literature raise the additional challenge of pinpointing gaps in current research. Based on our analysis, we found a number of additional directions for future research:

Learning the opponent's reservation value and deadline in multi-issue negotiations against arbitrary opponents. Currently, all opponent models that estimate the reservation value or deadline do so in a single-issue negotiation about quantitative issue, using Bayesian learning and/or non-linear regression. Furthermore, most models assume that the general form of the opponent's decision function is known, except for its parameters. It would be interesting to extend current techniques to negotiations in which the opponent is not constrained to use a particular strategy and the opponent's preference order is unknown beforehand. Since this is a multi-layered learning process of preference learning and strategy prediction, neural networks would be a natural candidate for this task.

Modeling the opponent's preference profile in highly complex environments. We found that many models assume linear additive utility functions when estimating the opponent's preferences. In practice, there are often interaction effects between issues, which cannot be captured using linear utility functions. While we discussed some models able to capture non-linear preferences, scalability appears to be a general issue. A direction for future work is to develop opponent models that can model non-linear preferences in large outcome spaces.

Estimating the opponent's acceptance strategy in multi-issue negotiations with arbitrary opponents. We found few models that estimate the opponent's acceptance strategy in multi-issue negotiations, and those that do either assume the opponent's preferences are known (or can be easily estimated) or ignore the effect of time pressure. An important direction for future work is a more general solution method to estimate what and when the opponent will accept, and to define measures for their accuracy.

Development of accuracy metrics for models estimating the reservation value, deadline, and acceptance strategy. As we discussed in Sect. 6, accuracy measures are still lacking for these three types of models. The first two types could be calculated as the distance between the actual and estimated value, but more advanced methods are not yet determined. Formulating an accuracy measure for the acceptance strategy may be the most challenging of all, and could be an important step towards more advanced strategies for multi-session negotiations.

Finally, we focused on learning about the opponent, and not so much on the reverse problem: what the opponent learns, or should learn, about us; i.e., *second order* opponent models. On the one hand, we might actively attempt to keep the opponent from learning about us to avoid exploitation, but there is a balance to be maintained as well; it is, after all, also important for both parties to jointly explore the outcome space to reach a win-win agreement. Models that quantify the opponent's knowledge about us can act as an essential link in combining different opponent modeling techniques. For example, we could employ a well-established preference estimation method to assess how the opponent acted according to its own utility. Then, we can apply one of the regression or forecasting techniques we have discussed to deduce the opponent's strategy according to its own utility. In combination with what the opponent knows about us, this can predict what the agent will do in the future, this time according to *our* utility. In such a setting, it may be worthwhile to study techniques that are able to reveal exactly the right kind of information in order to reach the most beneficial outcome.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Agrawal, M. K., & Chari, K. (2009). Learning negotiation support systems in competitive negotiations: A study of negotiation behaviours and system impacts. *International Journal of Intelligent Information Technologies*, 5(1), 1–23.
2. An, B., & Lesser, V. R. (2012). 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. Studies in computational intelligence* (Vol. 383, pp. 145–149). Berlin: Springer.
3. An, B., Lesser, V. R., & Sim, K. M. (2011). Strategic agents for multi-resource negotiation. *Autonomous Agents and Multi-Agent Systems*, 23(1), 114–153.
4. An, B., Sim, K. M., Tang, L. G., Miao, C. Y., Shen, Z. Q., & Cheng, D. J. (2008). Negotiation agents' decisionmaking using Markov chains. In T. Ito, H. Hattori, M. Zhang, & T. Matsuo (Eds.), *Rational robust, and secure negotiations in multi-agent systems. Studies in computational intelligence* (Vol. 89, pp. 3–23). Berlin: Springer.
5. Axelrod, R. (1984). *The evolution of cooperation*. New York: Basic Books.
6. Axelrod, R., & Dion, D. (1988). The further evolution of cooperation. *Science*, 242(4884), 1385–1390.

7. Aydoğan, R., & Yolum, P. (2006). Learning consumer preferences for content-oriented negotiation. In *AAMAS workshop on business agents and the semantic web (BASeWEB)* (pp. 43–52). New York: ACM Press.
8. Aydoğan, R., & Yolum, P. (2012). The effect of preference representation on learning preferences in negotiation. In T. Ito, M. Zhang, V. Robu, S. Fatima, & T. Matsuo (Eds.), *New trends in agent-based complex automated negotiations. Studies in computational intelligence* (Vol. 383, pp. 3–20). Berlin: Springer.
9. Aydoğan, R., & Yolum, P. (2012). Learning opponent's preferences for effective negotiation: An approach based on concept learning. *Autonomous Agents and Multi-Agent Systems*, 24, 104–140.
10. Baarslag, T., Fujita, K., Gerding, E. H., Hindriks, K. V., Ito, T., Jennings, N. R., et al. (2013). Evaluating practical negotiating agents: Results and analysis of the 2011 international competition. *Artificial Intelligence*, 198, 73–103.
11. Baarslag, T., Hendriks, M. J. C., Hindriks, K. V., & Jonker, C. M. (2012). Measuring the performance of online opponent models in automated bilateral negotiation. In Thielscher, M., & Zhang, D. (Eds.), *AI 2012: Advances in artificial intelligence. Lecture Notes in Computer Science* (Vol. 7691, pp. 1–14). Berlin: Springer.
12. Baarslag, T., Hendriks, M. J. C., Hindriks, K. V., & Jonker, C. M. (2013). Predicting the performance of opponent models in automated negotiation. In *International joint conferences on web intelligence (WI) and intelligent agent technologies (IAT), 2013 IEEE/WIC/ACM* (Vol. 2, pp. 59–66).
13. Baarslag, T., & Hindriks, K. V. (2013). Accepting optimally in automated negotiation with incomplete information. In *Proceedings of the 2013 international conference on autonomous agents and multi-agent systems, AAMAS '13* (pp. 715–722). Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.
14. Baarslag, T., Hindriks, K. V., & Jonker, C. M. (2011). Towards a quantitative concession-based classification method of negotiation strategies. In Kinny, D., Hsu, J. Y., Governatori, G., & Ghose, A. K. (Eds.), *Agents in principle, agents in practice. Lecture notes in computer science* (Vol. 7047, pp. 143–158). Berlin: Springer.
15. Baarslag, T., Hindriks, K. V., & Jonker, C. M. (2013). A tit for tat negotiation strategy for real-time bilateral negotiations. In Ito, T., Zhang, M., Robu, V., & Matsuo, T. (Eds.), *Complex automated negotiations: Theories, models, and software competitions. Studies in computational intelligence*. (Vol. 435, pp. 229–233). Berlin: Springer.
16. Baarslag, T., Hindriks, K. V., Jonker, C. M., Kraus, S., & Lin, R. (2012). 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. Studies in computational intelligence* (Vol. 383, pp. 113–135). Berlin: Springer-Verlag.
17. Bates, D. M., & Watts, D. G. (1988). *Nonlinear regression analysis and its applications*. New York: Wiley.
18. Beam, C., & Segev, A. (1997). Automated negotiations: A survey of the state of the art. *Wirtschaftsinformatik*, 39(3), 263–268.
19. Binmore, K., & Vulkan, N. (1999). Applying game theory to automated negotiation. *Netnomics*, 1(1), 1–9.
20. Bosse, T., & Jonker, C. M. (2005). Human vs. computer behaviour in multi-issue negotiation. In *Proceedings of the rational, robust, and secure negotiation mechanisms in multi-agent systems, RRS '05* (pp. 11–24). Washington, DC: IEEE Computer Society.
21. Bosse, T., Jonker, C. M., van der Meij, L., Robu, V., & Treur, J. (2005). A system for analysis of multi-issue negotiation. In R. Unland, M. Calisti, & M. Klusch (Eds.), *Software agent-based applications, platforms and development kits. Whitestein series in software agent technologies* (pp. 253–279). Basel: Birkhäuser.
22. Bosse, T., Jonker, C. M., van der Meij, L., & Treur, J. (2008). Automated formal analysis of human multi-issue negotiation processes. *Multiagent and Grid Systems*, 4(2), 213–233.
23. Boutilier, C., Brafman, R. I., Domshlak, C., Hoos, H. H., & Poole, D. (2004). CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research*, 21, 135–191.
24. Bragt, D. D. B., & La Poutré, J. A. (2003). Why agents for automated negotiations should be adaptive. *Netnomics*, 5(2), 101–118.
25. Braun, P., Brzostowski, J., Kersten, G. E., Kim, J. B., Kowalczyk, R., Strecker, S., & Vahidov, R. M. (2006). e-negotiation systems and software agents: Methods, models, and applications In *Intelligent decision-making support systems. Decision engineering* (pp. 271–300). London: Springer.
26. Brazian, D., & Boutilier, C. (2008). Elicitation of factored utilities. *AI Magazine*, 29(4), 79–92.

27. Broekens, J., Jonker, C. M., & Meyer, J.-J. C. (2010). Affective negotiation support systems. *Journal of Ambient Intelligence and Smart Environments*, 2(2), 121–144.
28. Brzostowski, J., & Kowalczyk, R. (2006). Adaptive negotiation with on-line prediction of opponent behaviour in agent-based negotiations. In *Proceedings of the IEEE/WIC/ACM international conference on intelligent agent technology, IAT '06* (pp. 263–269). Washington, DC: IEEE Computer Society.
29. Brzostowski, J., & Kowalczyk, R. (2006). Predicting partner's behaviour in agent negotiation. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems, AAMAS '06* (pp. 355–361). New York: ACM.
30. Buffett, S., Comeau, L., Spencer, B., & Fleming, M. W. (2006). Detecting opponent concessions in multi-issue automated negotiation. In *Proceedings of the 8th international conference on electronic commerce: The new e-commerce: innovations for conquering current barriers, obstacles and limitations to conducting successful business on the internet, ICEC '06* (pp. 11–18). New York: ACM.
31. Buffett, S., & Spencer, B. (2005). Learning opponents' preferences in multi-object automated negotiation. In *Proceedings of the 7th international conference on electronic commerce, ICEC '05* (pp. 300–305). New York: ACM.
32. Buffett, S., & Spencer, B. (2007). A bayesian classifier for learning opponents' preferences in multi-object automated negotiation. *Electronic Commerce Research and Applications*, 6(3), 274–284.
33. Bui, H. H., Venkatesh, S., & Kieronska, D. H. (1995). *An architecture for negotiating agents that learn*. Technical report, Department of Computer Science, Curtin University of Technology, Perth.
34. Bui, H. H., Venkatesh, S., & Kieronska, D. H. (1999). Learning other agents' preferences in multi-agent negotiation using the bayesian classifier. *International Journal of Cooperative Information Systems*, 8(4), 273–293.
35. Carbonneau, R. A., Kersten, G. E., & Vahidov, R. M. (2008). Predicting opponent's moves in electronic negotiations using neural networks. *Expert Systems with Applications*, 34(2), 1266–1273.
36. Carbonneau, R. A., Kersten, G. E., & Vahidov, R. M. (2011). Pairwise issue modeling for negotiation counteroffer prediction using neural networks. *Decision Support Systems*, 50(2), 449–459.
37. Carbonneau, R. A. & Vahidov, R. M. (2014). What's next? Predicting the issue a negotiator would choose to concede on. In *Group decision and negotiation 2014: Proceedings of the joint international conference of the INFORMS GDN section and the EURO Working Group on DSS* (p. 52). EWG-DSS.
38. Carnevale, P. J. D., & Lawler, E. J. (1986). Time pressure and the development of integrative agreements in bilateral negotiations. *The Journal of Conflict Resolution*, 30(4), 636–659.
39. Carnevale, P. J. D., O'Connor, K. M., & McCusker, C. (1993). Time pressure in negotiation and mediation. In O. Svenson & A. J. Maule (Eds.), *Time pressure and stress in human judgment and decision making* (pp. 117–127). New York: Springer.
40. Chatterjee, K. (1996). Game theory and the practice of bargaining. *Group Decision and Negotiation*, 5(4–6), 355–369.
41. Chen, J.-H., Chao, K.-M., Godwin, N., Reeves, C., & Smith, P. (2002). An automated negotiation mechanism based on co-evolution and game theory. In *Proceedings of the 2002 ACM symposium on applied computing, SAC'02* (pp. 63–67). New York: ACM.
42. Chen, L., & Pu, P. (2004). *Survey of preference elicitation methods*. Technical report, Ecole Polytechnique Federale de Lausanne (EPFL), IC/2004/67.
43. Chen, S., Hao, J., Weiss, G., Tuyls, K., & Leung, H.-F. (2014). Evaluating practical automated negotiation based on spatial evolutionary game theory. In Lutz, C. & Thielscher, M. (Ed.), *KI 2014: Advances in artificial intelligence. Lecture Notes in Computer Science*. (Vol. 8736, pp. 147–158). Heidelberg: Springer International Publishing.
44. Chen, S., & Weiss, G. (2012). An efficient and adaptive approach to negotiation in complex environments. In De, R., Luc, B., Christian, D., Didier, D., Patrick, F., Paolo, H., Fredrik, L., & Peter J. F. (Eds.), *ECAI. Frontiers in artificial intelligence and applications*. (Vol. 242, pp. 228–233). Amsterdam: IOS Press.
45. Chen, S., & Weiss, G. (2012). A novel strategy for efficient negotiation in complex environments. In I. J. Timm & C. Guttman (Eds.), *Multiagent system technologies. Lecture notes in computer science*. (Vol. 7598, pp. 68–82). Berlin: Springer.
46. Chen, S., & Weiss, G. (2014). OMAC: A discrete wavelet transformation based negotiation agent. 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. Studies in computational intelligence*. (Vol. 535, pp. 187–196). Japan: Springer.
47. Cheng, C.-B., Henry Chan, C.-C., & Lin, K.-C. (2006). Intelligent agents for e-marketplace: Negotiation with issue trade-offs by fuzzy inference systems. *Decision Support Systems*, 42(2), 626–638.
48. Chevalayre, Y., Dunne, P. E., Endriss, U., Lang, J., Lemaître, M., Maudet, N., et al. (2006). Issues in multiagent resource allocation. *Informatica*, 30, 3–31.

49. Chevaleyre, Y., Endriss, U., Estivie, S., & Maudet, N. (2004). Multiagent resource allocation with k-additive utility functions. In *Proceedings of the DIMACS-LAMSADE workshop on computer science and decision theory* (pp. 83–100).
50. Coehoorn, R. M., & Jennings, N. R. (2004). Learning an opponent's preferences to make effective multi-issue negotiation trade-offs. In *Proceedings of the 6th international conference on electronic commerce, ICEC'04* (pp. 59–68). New York: ACM.
51. Šerban, L. D., Silaghi, G. C., & Litan, C. M. (2012). 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. Series of studies in computational intelligence* (pp. 159–165). Berlin: Springer-Verlag.
52. de Jonge, D. (2015) *Negotiations over large agreement spaces*. PhD thesis, Universitat Autònoma de Barcelona.
53. Domshlak, C., Hüllermeier, E., Kaci, S., & Prade, H. (2011). Preferences in AI: An overview. *Artificial Intelligence*, 175(7–8), 1037–1052.
54. Druckman, D., & Olekalns, M. (2008). Emotions in negotiation. *Group Decision and Negotiation*, 17(1), 1–11.
55. Dzung, R.-J., & Lin, Y.-C. (2005). Searching for better negotiation agreement based on genetic algorithm. *Computer-Aided Civil and Infrastructure Engineering*, 20(4), 280–293.
56. Fabregues, A., Navarro, D., Serrano, A., & Sierra, C. (2010). DipGame: A testbed for multiagent systems. In *Proceedings of the 9th international conference on autonomous agents and multiagent systems. AAMAS'10*. (Vol. 1, pp. 1619–1620). Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.
57. Fang, F., Xin, Y., Yun, X., & Haitao, X. (2008). An opponent's negotiation behavior model to facilitate buyer-seller negotiations in supply chain management. In *International symposium on electronic commerce and security*.
58. Farag, G. M., AbdelRahman, S. E.-S., Bahgat, R., & A-Moneim, A. M. (2010). Towards KDE mining approach for multi-agent negotiation. In *The 7th international conference on informatics and systems (INFOS)* (pp. 1–7). Barcelona: IEEE.
59. Faratin, P., Sierra, C., & Jennings, N. R. (1998). Negotiation decision functions for autonomous agents. *Robotics and Autonomous Systems*, 24(3–4), 159–182.
60. Faratin, P., Sierra, C., & Jennings, N. R. (2002). Using similarity criteria to make issue trade-offs in automated negotiations. *Artificial Intelligence*, 142(2), 205–237. International conference on multiagent systems 2000.
61. Fatima, S. S., Wooldridge, M. J., & Jennings, N. R. (2002). Multi-issue negotiation under time constraints. In *AAMAS'02: Proceedings of the first international joint conference on autonomous agents and multiagent systems* (pp. 143–150). New York: ACM.
62. Fatima, S. S., Wooldridge, M. J., & Jennings, N. R. (2002). Optimal negotiation strategies for agents with incomplete information. In *Revised papers from the 8th international workshop on intelligent agents VIII, ATAL'01* (pp. 377–392). London: Springer-Verlag.
63. Fatima, S. S., Wooldridge, M. J., & Jennings, N. R. (2003). Optimal agendas for multi-issue negotiation. In *Proceedings of the second international joint conference on autonomous agents and multiagent systems, AAMAS'03* (pp. 129–136). New York: ACM.
64. Fatima, S. S., Wooldridge, M. J., & Jennings, N. R. (2005). A comparative study of game theoretic and evolutionary models of bargaining for software agents. *Artificial Intelligence Review*, 23(2), 187–205.
65. Fatima, S. S., Wooldridge, M. J., & Jennings, N. R. (2006). Multi-issue negotiation with deadlines. *Journal of Artificial Intelligence Research*, 27, 381–417.
66. Ficici, S. G., & Pfeffer, A. (2008). Modeling how humans reason about others with partial information. In *Proceedings of the 7th international joint conference on autonomous agents and multiagent systems. AAMAS'08*. (Vol. 1, pp. 315–322). Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.
67. Fürnkranz, J. (2001). Machine learning in games: A survey. In J. Fürnkranz & M. Kubat (Eds.), *Machines that learn to play games* (pp. 11–59). Commack, NY: Nova Science Publishers, Inc.
68. Gal, Y., Grosz, B. J., Kraus, S., Pfeffer, A., & Shieber, S. (2005). Colored trails: A formalism for investigating decision-making in strategic environments. In *Proceedings of the 2005 IJCAI workshop on reasoning, representation, and learning in computer games* (pp. 25–30).
69. Gerding, E. H., Bragt, D. D. B., & La Poutré, J. A. (2000) *Scientific approaches and techniques for negotiation: A game theoretic and artificial intelligence perspective*. Technical report, CWI (Centre for Mathematics and Computer Science), Amsterdam, The Netherlands.
70. Gode, D. K., & Sunder, S. (1993). Allocative efficiency in markets with zero intelligence (ZI) traders: Market as a partial substitute for individual rationality. *Journal of Political Economy*, 101(1), 119–137.

71. Guttman, R. H. & Maes, P. (1999). Agent-mediated integrative negotiation for retail electronic commerce. In P. Noriega & C. Sierra (Eds.), *Agent mediated electronic commerce. Lecture notes in computer science* (Vol. 1571, pp. 70–90). Berlin: Springer.
72. Gwak, J., & Sim, K. M. (2011). Bayesian learning based negotiation agents for supporting negotiation with incomplete information. In *Proceedings of the international multicongference of engineers and computer scientists* (Vol. 1, pp. 163–168).
73. Haberland, V., Miles, S., & Luck, M. (2012). Adaptive negotiation for resource intensive tasks in grids. In *STAIRS* (pp. 125–136).
74. Hadfi, R. & Ito, T. (2014). Addressing complexity in multi-issue negotiation via utility hypergraphs. In *Proceedings of the twenty-eighth AAAI conference on artificial intelligence*.
75. Hao, J. & Leung, H.-F. (2012). ABiNeS: An adaptive bilateral negotiating strategy over multiple items. In *Proceedings of the 2012 IEEE/WIC/ACM international joint conferences on web intelligence and intelligent agent technology WI-IAT'12* (Vol. 2, pp. 95–102). Washington, DC: IEEE Computer Society.
76. Hao, J., & Leung, H.-F. (2014). CUHK agent: 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. Studies in computational intelligence* (Vol. 535, pp. 171–179). Japan: Springer.
77. Hao, J., Song, S., Leung, H.-F., & Ming, Z. (2014). An efficient and robust negotiating strategy in bilateral negotiations over multiple items. *Engineering Applications of Artificial Intelligence*, 34, 45–57.
78. Haykin, S. (1994). *Neural networks: A comprehensive foundation* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.
79. He, M., Jennings, N. R., & Leung, H.-F. (2003). On agent-mediated electronic commerce. *IEEE Transactions on Knowledge and Data Engineering*, 15(4), 985–1003.
80. Hindriks, K. V. & Jonker, C. M. (2009). Creating human-machine synergy in negotiation support systems: Towards the pocket negotiator. In *Proceedings of the 1st international working conference on human factors and computational models in negotiation, HuCom'08* (pp. 47–54). New York: ACM.
81. Hindriks, K. V., Jonker, C. M., & Tykhonov, D. (2009). The benefits of opponent models in negotiation. In *Proceedings of the 2009 IEEE/WIC/ACM international joint conference on web intelligence and intelligent agent technology* (Vol. 2, pp. 439–444). New York: IEEE Computer Society.
82. Hindriks, K. V., Jonker, C. M., & Tykhonov, D. (2011). Let's dans! An analytic framework of negotiation dynamics and strategies. *Web Intelligence and Agent Systems*, 9(4), 319–335.
83. Hindriks, K. V. & Tykhonov, D. (2008). Opponent modelling in automated multi-issue negotiation using bayesian learning. In *Proceedings of the 7th international joint conference on autonomous agents and multiagent systems, AAMAS'08* (Vol. 1, pp. 331–338). Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.
84. Hindriks, K. V., & Tykhonov, D. (2010). Towards a quality assessment method for learning preference profiles in negotiation. In W. Ketter, J. A. La Poutré, N. Sadeh, O. Shehory, & W. Walsh (Eds.), *Agent-mediated electronic commerce and trading agent design and analysis. Lecture notes in business information processing* (Vol. 44, pp. 46–59). Berlin: Springer.
85. Hou, C. (2004). Predicting agents tactics in automated negotiation. In *Proceedings of the IEEE/WIC/ACM international conference on intelligent agent technology* (pp. 127–133). New York: IEEE Computer Society.
86. Ito, T., Hattori, H., & Klein, M. (2007). Multi-issue negotiation protocol for agents: Exploring nonlinear utility spaces. In *Proceedings of the 20th international joint conference on artificial intelligence, IJCAI'07* (pp. 1347–1352). San Francisco, CA: Morgan Kaufmann Publishers Inc.
87. Ito, T., Klein, M., & Hattori, H. (2008). A multi-issue negotiation protocol among agents with nonlinear utility functions. *Multiagent and Grid Systems*, 4(1), 67–83.
88. Ito, T., Zhang, M., Robu, V., Fatima, S., & Matsuo, T. (2011). *New trends in agent-based complex automated negotiations* (Vol. 383). Berlin: Springer Science & Business Media.
89. Jazayeriy, H., Azmi-Murad, M., Nasir Sulaiman, M., & Udzir, N. I. (2011). A review on soft computing techniques in automated negotiation. *Scientific Research and Essays*, 6(24), 5100–5106.
90. Jazayeriy, H., Azmi-Murad, M., Sulaiman, N., & Udzir, N. I. (2011). The learning of an opponent's approximate preferences in bilateral automated negotiation. *Journal of Theoretical and Applied Electronic Commerce Research*, 6(3), 65–84.
91. Jennings, N. R., Faratin, P., Lomuscio, A. R., Parsons, S., Wooldridge, M. J., & Sierra, C. (2001). Automated negotiation: Prospects, methods and challenges. *Group Decision and Negotiation*, 10(2), 199–215.
92. Ji, S.-J., Zhang, C.-J., Sim, K.-M., & Leung, H.-F. (2014). A one-shot bargaining strategy for dealing with multifarious opponents. *Applied Intelligence*, 40(4), 557–574.

93. Jones, M. C., Marron, J. S., & Sheather, S. J. (1996). A brief survey of bandwidth selection for density estimation. *Journal of the American Statistical Association*, 91(433), 401–407.
94. Jonker, C. M., & Robu, V. (2004). Automated multi-attribute negotiation with efficient use of incomplete preference information. In *Proceedings of the third international joint conference on autonomous agents and multiagent systems, AAMAS'04* (Vol. 3, pp. 1054–1061). Washington, DC: IEEE Computer Society.
95. Jonker, C. M., Robu, V., & Treur, J. (2007). An agent architecture for multi-attribute negotiation using incomplete preference information. *Autonomous Agents and Multi-Agent Systems*, 15, 221–252.
96. Karp, A. H., Wu, R., Chen, K.-Y., & Zhang, A. (2004). A game tree strategy for automated negotiation. In *Proceedings of the 5th ACM conference on electronic commerce, EC'04* (pp. 228–229). New York: ACM.
97. Kawaguchi, S., Fujita, K., & Ito, T. (2012). 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. Series of studies in computational intelligence* (pp. 137–144). Berlin: Springer-Verlag.
98. Keeney, R. L., & Raiffa, H. (1976). *Decisions with multiple objectives*. Cambridge: Cambridge University Press.
99. Kersten, G. E., & Lai, H. (2007). Negotiation support and e-negotiation systems: An overview. *Group Decision and Negotiation*, 16(6), 553–586.
100. Van Kleef, G. A., De Dreu, C. K. W., & Manstead, A. S. R. (2004). The interpersonal effects of emotions in negotiations: A motivated information processing approach. *Journal of Personality and Social Psychology*, 87(4), 510.
101. Klein, M., Faratin, P., Sayama, H., & Bar-Yam, Y. (2003). Negotiating complex contracts. *Group Decision and Negotiation*, 12, 111–125.
102. Klein, M., & Lu, S. C.-Y. (1989). Conflict resolution in cooperative design. *Artificial Intelligence in Engineering*, 4(4), 168–180.
103. Klos, T. B., Somefun, K., & La Poutré, J. A. (2011). Automated interactive sales processes. *IEEE Intelligent Systems*, 26(4), 54–61.
104. Kolomvatsos, K., Anagnostopoulos, C., & Hadjiefthymiades, S. (2013). Determining the optimal stopping time for automated negotiations. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 99, 1–1.
105. Kowalczyk, R., Ulieru, M., & Unland, R. (2003). Integrating mobile and intelligent agents in advanced e-commerce: A survey. In J. G. Carbonell, J. Siekmann, R. K., Jörg P. M., Huaglority Tianfield, & R. Unland (Eds.), *Agent technologies, infrastructures, tools, and applications for E-services. Lecture notes in computer science* (Vol. 2592, pp. 295–313). Berlin: Springer.
106. Kraus, S. (1997). Negotiation and cooperation in multi-agent environments. *Artificial Intelligence*, 94(1–2), 79–97.
107. Kraus, S. (2001). *Strategic negotiation in multiagent environments*. Cambridge, MA: MIT Press.
108. Kraus, S., & Lehmann, D. (1995). Designing and building a negotiating automated agent. *Computational Intelligence*, 11(1), 132–171.
109. Kraus, S., Wilkenfeld, J., & Zlotkin, G. (1995). Multiagent negotiation under time constraints. *Artificial Intelligence*, 75(2), 297–345.
110. Kröse, B. & van der Smagt, P. (1993). An introduction to neural networks. Amsterdam: University of Amsterdam
111. Lai, G., Sycara, K. P., & Li, C. (2008). A decentralized model for automated multi-attribute negotiations with incomplete information and general utility functions. *Multiagent and Grid Systems*, 4(1), 45–65.
112. Lau, R. Y. K. (2005). Adaptive negotiation agents for e-business. In *Proceedings of the 7th international conference on electronic commerce, ICEC'05* (pp. 271–278). New York: ACM.
113. Lau, R. Y. K., Li, Y., Song, D., & Chi-Wai Kwok, R. (2008). Knowledge discovery for adaptive negotiation agents in e-marketplaces. *Decision Support Systems*, 45(2), 310–323.
114. Lau, R. Y. K., Tang, M., Wong, O., Milliner, S. W., & Phoebe Chen, Y.-P. (2006). An evolutionary learning approach for adaptive negotiation agents. *International Journal of Intelligent Systems*, 21(1), 41–72.
115. Lee, C. C., & Ou-Yang, C. (2009). A neural networks approach for forecasting the supplier's bid prices in supplier selection negotiation process. *Expert Systems with Applications*, 36(2, Part 2), 2961–2970.
116. Leonardz, B. (1973). *To stop or not to stop. Some elementary optimal stopping problems with economic interpretations*. Stockholm: Almqvist & Wiksell.
117. Lewicki, R. J., Saunders, D. M., Barry, B., & Minton, J. W. (2003). *Essentials of negotiation*. Boston, MA: McGraw-Hill.
118. Li, C., Giampapa, J., & Sycara, K. P. (2003). *A review of research literature on bilateral negotiations*. Technical report, Robotics Institute, Pittsburgh, PA.

119. Li, C., Giampapa, J., & Sycara, K. P. (2006). Bilateral negotiation decisions with uncertain dynamic outside options. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 36(1), 31–44.
120. Liang, Y.-Q., & Yuan, Y. (2008). Co-evolutionary stability in the alternating-offer negotiation. In *IEEE conference on cybernetics and intelligent systems* (pp. 1176–1180).
121. Lin, R., & Kraus, S. (2010). Can automated agents proficiently negotiate with humans? *Communications of the ACM*, 53(1), 78–88.
122. Lin, R., Kraus, S., Baarslag, T., Tykhonov, D., Hindriks, K. V., & Jonker, C. M. (2014). Genius: An integrated environment for supporting the design of generic automated negotiators. *Computational Intelligence*, 30(1), 48–70.
123. Lin, R., Kraus, S., Wilkenfeld, J., & Barry, J. (2006). An automated agent for bilateral negotiation with bounded rational agents with incomplete information. In *Proceedings of the 2006 conference on ECAI 2006: 17th European conference on artificial intelligence* (pp. 270–274). The Netherlands: Amsterdam.
124. Lin, R., Kraus, S., Wilkenfeld, J., & Barry, J. (2008). Negotiating with bounded rational agents in environments with incomplete information using an automated agent. *Artificial Intelligence*, 172(6–7), 823–851.
125. Lin, R., Oshrat, Y., & Kraus, S. (2009). Investigating the benefits of automated negotiations in enhancing people's negotiation skills. In *AAMAS'09: Proceedings of the 8th international conference on autonomous agents and multiagent systems* (pp. 345–352). Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.
126. Lomuscio, A. R., Wooldridge, M. J., & Jennings, N. R. (2003). A classification scheme for negotiation in electronic commerce. *Group Decision and Negotiation*, 12(1), 31–56.
127. Lopez-Carmona, M. A., Marsa-Maestre, I., Klein, M., & Ito, T. (2012). Addressing stability issues in mediated complex contract negotiations for constraint-based, non-monotonic utility spaces. *Autonomous Agents and Multi-Agent Systems*, 24(3), 485–535.
128. Marsa-Maestre, I., Klein, M., Jonker, C. M., & Aydoğan, R. (2014). From problems to protocols: Towards a negotiation handbook. *Decision Support Systems*, 60, 39–54.
129. Marsa-Maestre, I., Lopez-Carmona, M. A., Ito, T., Zhang, M., Bai, Q., & Fujita, K. (2014). *Novel insights in agent-based complex automated negotiation* (Vol. 535). Tokyo: Springer.
130. Marsa-Maestre, I., Lopez-Carmona, M. A., Velasco, J. R., Ito, T., Klein, M., & Fujita, K. (2009). Balancing utility and deal probability for auction-based negotiations in highly nonlinear utility spaces. In *Proceedings of the 21st international joint conference on artificial intelligence, IJCAI'09* (pp. 214–219). San Francisco: Morgan Kaufmann Publishers Inc.
131. Mason, J. C., & Handscomb, D. C. (2002). *Chebyshev polynomials*. London: Taylor & Francis.
132. Masvoula, M. (2013). Forecasting negotiation counterpart's offers: A focus on session-long learning agents. In *COGNITIVE 2013, the fifth international conference on advanced cognitive technologies and applications* (pp. 71–76).
133. Masvoula, M., Halatsis, C., & Martakos, D. (2011). Predictive automated negotiators employing risk-seeking and risk-averse strategies. In L. Iliadis & C. Jayne (Eds.), *Engineering applications of neural networks. IFIP advances in information and communication technology* (Vol. 363, pp. 325–334). Boston: Springer.
134. Masvoula, M., Kanellis, P., & Martakos, D. (2010). A review of learning methods enhanced in strategies of negotiating agents. In *Proceedings of the 12th international conference on enterprise information systems* (pp. 212–219).
135. Matos, N., Sierra, C., & Jennings, N. R. (1998). Determining successful negotiation strategies: An evolutionary approach. In *Proceedings international conference on multi agent systems* (pp. 182–189).
136. McTear, M. F. (1993). User modelling for adaptive computer systems: A survey of recent developments. *Artificial Intelligence Review*, 7(3–4), 157–184.
137. Mok, W. H., & Sundarraj, R. P. (2005). Learning algorithms for single-instance electronic negotiations using the time-dependent behavioral tactic. *ACM Transactions on Internet Technology*, 5(1), 195–230.
138. Motulsky, H. J., & Ransnas, L. A. (1987). Fitting curves to data using nonlinear regression: A practical and nonmathematical review. *The FASEB Journal*, 1(5), 365–374.
139. Mudgal, C., & Vassileva, J. (2000). Bilateral negotiation with incomplete and uncertain information: A decision-theoretic approach using a model of the opponent. In *Proceedings of the 4th international workshop on cooperative information agents IV, the future of information agents in cyberspace, CIA'00* (pp. 107–118). London: Springer-Verlag.
140. Narayanan, V., & Jennings, N. R. (2006). Learning to negotiate optimally in non-stationary environments. In M. Klusch, M. Rovatsos, & T. R. Payne (Eds.), *Cooperative information agents X. Lecture notes in computer science* (Vol. 4149, pp. 288–300). Berlin: Springer.
141. Nash, J. F. (1950). The bargaining problem. *Econometrica*, 18(2), 155–162.

142. Nguyen, T. D. & Jennings, N. R. (2004). Coordinating multiple concurrent negotiations. In *Proceedings of the third international joint conference on autonomous agents and multiagent systems, AAMAS'04* (Vol. 3, pp. 1064–1071). Washington, DC: IEEE Computer Society.
143. Niemann, C., & Lang, F. (2009). 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. Studies in computational intelligence* (Vol. 233, pp. 119–137). Berlin: Springer.
144. Nisan, N. (2006). *Bidding languages. Combinatorial auctions*. Cambridge, MA: MIT Press.
145. Oliver, J. R. (2005). On learning negotiation strategies by artificial adaptive agents in environments of incomplete information. In S. O. Kimbrough & D. J. Wu (Eds.), *Formal modelling in electronic commerce. International handbooks on information systems* (pp. 445–461). Berlin: Springer.
146. Oprea, M. (2002). An adaptive negotiation model for agent-based electronic commerce. *Studies in Informatics and Control*, 11(3), 271–279.
147. Osborne, M. J., & Rubinstein, A. (1990). *Bargaining and markets (Economic theory, econometrics, and mathematical economics)*. New York: Academic Press.
148. Osborne, M. J., & Rubinstein, A. (1994). *A course in game theory* (1st ed.). Cambridge, MA: The MIT Press.
149. Oshrat, Y., Lin, R., & Kraus, S. (2009). Facing the challenge of human-agent negotiations via effective general opponent modeling. In *Proceedings of the 8th international conference on autonomous agents and multiagent systems, AAMAS'09* (Vol. 1, pp. 377–384). Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.
150. Ozonat, K. & Singhal, S. (2010). Design of negotiation agents based on behavior models. In L. Chen, P. Triantafillou, & T. Suel (Eds.), *Web information systems engineering—WISE 2010. Lecture notes in computer science* (Vol. 6488, pp. 308–321). Berlin: Springer.
151. Papaioannou, I. V., Roussaki, I. G., & Anagnostou, M. E. (2008). Neural networks against genetic algorithms for negotiating agent behaviour prediction. *Web Intelligence and Agent Systems*, 6(2), 217–233.
152. Papaioannou, I. V., Roussaki, I. G., & Anagnostou, M. E. (2009). A survey on neural networks in automated negotiations. In J. R. Rabuñal, J. Dorado, & A. Pazos (Eds.), *Encyclopedia of artificial intelligence* (pp. 1524–1529). Hershey, PA: IGI Global.
153. Papaioannou, I. V., Roussaki, I. G., & Anagnostou, M. E. (2011). Multi-modal opponent behaviour prognosis in e-negotiations. In J. Cabestany, I. Rojas, & G. Joya (Eds.), *Advances in computational intelligence. Lecture notes in computer science* (Vol. 6691, pp. 113–123). Berlin: Springer.
154. Pruitt, D. G. (1981). *Negotiation behavior*. New York: Academic Press.
155. Abdel Rahman, S., Bahgat, R., & Farag, G. M. (2011). Order statistics bayesian-mining agent modelling for automated negotiation. *Informatica: An International Journal of Computing and Informatics*, 35(1), 123–137.
156. Rahwan, I., Kowalczyk, R., & Pham, H. H. (2002). Intelligent agents for automated one-to-many e-commerce negotiation. *Australian Computer Science Communications*, 24(1), 197–204.
157. Raiffa, H. (1953). Arbitration schemes for generalized two-person games. *Annals of Mathematics Studies*, 28, 361–387.
158. Raiffa, H. (1982). *The art and science of negotiation: How to resolve conflicts and get the best out of bargaining*. Cambridge, MA: Harvard University Press.
159. Raiffa, H., Richardson, J., & Metcalfe, D. (2003). *Negotiation analysis: The science and art of collaborative decision making*. Cambridge, MA: Harvard University Press.
160. Rau, H., Tsai, M.-H., Chen, C.-W., & Shiang, W.-J. (2006). Learning-based automated negotiation between shipper and forwarder. *Computers & Industrial Engineering*, 51(3), 464–481.
161. Ren, F. & Zhang, M. (2007). Predicting partners' behaviors in negotiation by using regression analysis. In Z. Zhang & J. Siekmann (Eds.), *Knowledge science, engineering and management. Lecture notes in computer science*. (Vol. 4798, pp. 165–176). Berlin: Springer.
162. Ren, Z., & Anumba, C. J. (2002). Learning in multi-agent systems: A case study of construction claims negotiation. *Advanced Engineering Informatics*, 16(4), 265–275.
163. Restifcar, A. & Haddawy, P. (2004). Inferring implicit preferences from negotiation actions. In *International symposium on artificial intelligence and mathematics, Fort Lauderdale, FL, USA*.
164. Robinson, W. N. (1990). Negotiation behavior during requirement specification. In *Proceedings of the 12th international conference on software engineering* (pp. 268–276).
165. Robu, V. & La Poutre, J. A. (2006). Retrieving the structure of utility graphs used in multi-item negotiations through collaborative filtering of aggregate buyer preferences. In *Proceedings of the 2nd international workshop on rational, robust and secure negotiations in MAS*. Berlin: Springer.

166. Robu, V., Somefun, K., & La Poutré, J. A. (2005). Modeling complex multi-issue negotiations using utility graphs. In *Proceedings of the fourth international joint conference on autonomous agents and multiagent systems, AAMAS'05* (pp. 280–287). New York: ACM.
167. Ros, R., & Sierra, C. (2006). A negotiation meta strategy combining trade-off and concession moves. *Autonomous Agents and Multi-Agent Systems*, 12, 163–181.
168. Rosenschein, J. S. (1986). *Rational interaction: Cooperation among intelligent agents*. PhD thesis, Stanford University, Stanford, CA.
169. Rosenschein, J. S., & Zlotkin, G. (1994). *Rules of encounter: Designing conventions for automated negotiation among computers*. Cambridge, MA: MIT Press.
170. Rubin, J. Z., & Brown, B. R. (1975). *The social psychology of bargaining and negotiation*. New York: Academic press.
171. Rubin, J., & Watson, I. (2011). Computer poker: A review. *Artificial Intelligence*, 175(5–6), 958–987.
172. Rubinstein, A. (1982). Perfect equilibrium in a bargaining model. *Econometrica*, 50(1), 97–109.
173. Saha, S., Biswas, A., & Sen, S. (2005). Modeling opponent decision in repeated one-shot negotiations. In *Proceedings of the fourth international joint conference on autonomous agents and multiagent systems, AAMAS'05* (pp. 397–403). New York: ACM.
174. Saha, S. & Sen, S. (2005). A bayes net approach to argumentation based negotiation. In I. R., Pavlos Moraitis, & C. Reed (Eds.), *Argumentation in multi-agent systems. Lecture notes in computer science* (pp. 208–222). Berlin: Springer.
175. Sánchez-Anguix, V., Valero, S., Julián, V., Botti, V., & García-Fornes, A. (2013). Evolutionary-aided negotiation model for bilateral bargaining in ambient intelligence domains with complex utility functions. *Information Sciences*, 222, 25–46.
176. Sánchez-Pagés, S. (2004). *The use of conflict as a bargaining tool against unsophisticated opponents*. ESE Discussion Papers 99, Edinburgh School of Economics, University of Edinburgh.
177. Sandholm, T. & Vulkan, N. (1999). Bargaining with deadlines. In *Proceedings of the sixteenth national conference on artificial intelligence and the eleventh Innovative applications of artificial intelligence conference, AAAI'99/IAAI'99* (pp. 44–51). Menlo Park, CA: American Association for Artificial Intelligence.
178. Sandholm, W. H. (2010). *Population games and evolutionary dynamics*. Cambridge, MA: MIT press.
179. Schadd, F. C., Bakkes, S., & Spronck, P. H. M. (2007). Opponent modeling in real-time strategy games. In *8th International conference on intelligent games and simulation (GAME-ON 2007)* (pp. 61–68).
180. Schatzmann, J., Weillhammer, K., Stuttle, M., & Young, S. (2006). A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The Knowledge Engineering Review*, 21(2), 97–126.
181. Sierra, C., Faratin, P., & Jennings, N. R. (1997). A service-oriented negotiation model between autonomous agents. In M. Boman, & W. van de Velde (Eds.), *Proceedings of the 8th European workshop on modelling autonomous agents in multi-agent world, MAAMAW-97. Lecture notes in artificial intelligence*. (Vol. 1237, pp. 17–35). Berlin: Springer-Verlag.
182. Silaghi, G. C., Șerban, L. D., & Litan, C. M. (2010). A framework for building intelligent SLA negotiation strategies under time constraints. In J. Altmann & O. F. Rana (Eds.), *Proceedings of economics of grids, clouds, systems, and services: 7th international workshop*. (Vol. 6296, pp. 48). New York: Springer-Verlag Inc.
183. Sim, K. M., Guo, Y., & Shi, B. (2007). Adaptive bargaining agents that negotiate optimally and rapidly. In *IEEE congress on evolutionary computation* (pp. 1007–1014). New York: IEEE.
184. Sim, K. M., Guo, Y., & Shi, B. (2009). BLGAN: Bayesian learning and genetic algorithm for supporting negotiation with incomplete information. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(1), 198–211.
185. Slembeck, T. (1999). *Reputations and fairness in bargaining—experimental evidence from a repeated ultimatum game with fixed opponents*. Technical report, EconWPA.
186. Smith, R. G. (1980). The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, 29(12), 1104–1113.
187. Sofer, I., Sarne, D., & Hassidim, A. (2012). Negotiation in exploration-based environment. In *Proceedings of the twenty-sixth AAAI conference on artificial intelligence*.
188. Somefun, K. & La Poutré, J. A. (2007). A fast method for learning non-linear preferences online using anonymous negotiation data. In M. Fasli & O. Shehory (Eds.), *Agent-mediated electronic commerce, automated negotiation and strategy design for electronic markets. Lecture notes in computer science* (Vol. 4452, pp. 118–131). Berlin: Springer.
189. Sycara, K. P. (1985). Arguments of persuasion in labour mediation. In *Proceedings of the 9th international joint conference on artificial intelligence* (Vol. 1, pp. 294–296). San Francisco, CA: Morgan Kaufmann Publishers Inc.

190. Sycara, K. P. (1988). Resolving goal conflicts via negotiation. In *Proceedings of the 7th national conference on artificial intelligence, St. Paul, MN, August 21–26, 1988*. (pp. 245–250).
191. Tu, T., Wolff, E., & Lamersdorf, W. (2000). Genetic algorithms for automated negotiations: A FSM-based application approach. In *Proceedings of the 11th international workshop on database and expert systems applications, DEXA'00* (pp. 1029–1033). Washington, DC: IEEE Computer Society.
192. Vahidov, R. M., Kersten, G. E., & Saade, R. (2014). An experimental study of software agent negotiations with humans. *Decision Support Systems*, 66, 135–145.
193. van den Herik, J., Donkers, J., & Spronck, P. H. M. (2005). Opponent modelling and commercial games. In K. Graham, & L. Simon (Eds.), *Proceedings of the IEEE 2005 symposium on computational intelligence and games* (pp. 15–25).
194. van Galen Last, N. (2012). 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. Studies in computational intelligence* (pp. 167–174). Berlin: Springer-Verlag.
195. van Krimpen, T., Looije, D., & Hajizadeh, S. (2013). Hardheaded. In T. Ito, M. Zhang, V. Robu, & T. Matsuo (Eds.), *Complex automated negotiations: Theories, models, and software competitions. Studies in computational intelligence* (Vol. 435, pp. 223–227). Berlin: Springer.
196. Williams, C. R. (2012) *Practical strategies for agent-based negotiation in complex environments*. PhD thesis, University of Southampton.
197. Williams, C. R., Robu, V., Gerding, E. H., & Jennings, N. R. (2011). Using gaussian processes to optimise concession in complex negotiations against unknown opponents. In *Proceedings of the twenty-second international joint conference on artificial intelligence, IJCAI'11* (Vol. 1, pp. 432–438). Menlo Park, CA: AAAI Press.
198. Williams, C. R., Robu, V., Gerding, E. H., & Jennings, N. R. (2012). 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. Studies in computational intelligence* (pp. 151–158). Berlin: Springer-Verlag.
199. Williams, C. R., Robu, V., Gerding, E. H., & Jennings, N. R. (2013). 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. Studies in computational intelligence* (Vol. 435, pp. 209–212). Berlin: Springer.
200. Williams, C. R., Robu, V., Gerding, E. H., & Jennings, N. R. (2014). An overview of the results and insights from the third automated negotiating agents competition (ANAC 2012). In M.-M. Ivan, M. A. Lopez-Carmona, T. Ito, M. Zhang, Q. Bai, & K. Fujita (Ed.), *Novel insights in agent-based complex automated negotiations. Studies in computational intelligence* (Vol. 535, pp. 151–162). Japan: Springer.
201. Wu, M., de Weerd, M., & La Poutré, J. A. (2013). Acceptance strategies for maximizing agent profits in online scheduling. In D. Esther, R. Valentin, O. Shehory, S. Stein, & A. Symeonidis (Eds.), *Agent-mediated electronic commerce. Designing trading strategies and mechanisms for electronic markets. Lecture notes in business information processing* (Vol. 119, pp. 115–128). Berlin: Springer.
202. Yaakov G. & Ilany, L. (2015). The fourth automated negotiation competition. In K. Fujita, T. Ito, M. Zhang, & V. Robu (Eds.), *Next frontier in agent-based complex automated negotiation. Studies in computational intelligence* (Vol. 596, pp. 129–136). Japan: Springer.
203. Yang, Y. (2012). A review of strategy design and evaluation of software negotiation agents. In *Proceedings of the 14th annual international conference on electronic commerce, ICEC'12* (pp. 155–156). New York: ACM.
204. Chao, Y., Ren, F., & Zhang, M. (2013). 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. Studies in computational intelligence* (Vol. 435, pp. 75–93). Berlin: Springer.
205. Zeng, D. & Sycara, K. P. (1997). Benefits of learning in negotiation. In *Proceedings of the fourteenth national conference on artificial intelligence and ninth conference on innovative applications of artificial intelligence, AAAI'97/IAAI'97* (pp. 36–41). Menlo Park, CA: AAAI Press.
206. Zeng, D., & Sycara, K. P. (1998). Bayesian learning in negotiation. *International Journal of Human-Computer Studies*, 48(1), 125–141.
207. Zhang, M., Tan, Z., Zhao, J., & Li, L. (2008). A bayesian learning model in the agent-based bilateral negotiation between the coal producers and electric power generators. In *International Symposium on intelligent information technology application workshops, 2008. IITAW'08* (pp. 859–862).
208. Zlotkin, G. & Rosenschein, J. S. (1989). Negotiation and task sharing among autonomous agents in cooperative domains. In *Proceedings of the 11th international joint conference on artificial intelligence, IJCAI'89* (Vol. 2, pp. 912–917). San Francisco, CA: Morgan Kaufmann Publishers Inc.