Algorithms for Automated Multilateral Negotiations

Konstantia Xenou
Supervisor: George Chalkiadakis
ECE department, Technical University of Crete
September 2017

Contents

Abstract

In multi-attribute negotiations, two or more intelligent automated agents sharing common or conflicting interests negotiate over various distinct issues, usually under uncertainty about the characteristics of their negotiating partners. Agents should be able to adaptively adjust their behavior during the negotiation process, by adopting efficient and effective automated negotiation techniques. This is a challenging process, since negotiators are not usually willing to reveal their private information and preferences, so as to avoid being exploited by the other participants during the negotiation.

To overcome these problems and boost agent performance, a modeling of opponents' preferences and strategy is usually incorporated—with the understanding that uncertainty regarding opponent preferences always exists in real-world settings. Opponent modeling can be usually shown to assist the agent to achieve efficient agreements, and thus to significantly increase the quality of the negotiation outcome. A multitude of negotiation strategies and opponent models have been coined and studied over the years; regardless, empirically comparing them to each other is not a straightforward exercise.

To this end, the international Autonomous Negotiation Agents Competition (ANAC) was initiated in 2009, and is conducted utilizing a "standard", purpose-built, negotiation platform ("Genius"). Genius provides a uniform, accepted by all, way of comparing state-of-the-art agent strategies.

In this thesis, we systematically developed several different negotiating strategies, along with accompanying opponent models. We employed concepts found in the literature, implemented known strategies, and proposed novel ones. This process resulted to the creation of *thirteen (13)* distinct agents. The developed agents were pitted against previous ANAC participants, and also against each other, with an extensive evaluation being conducted on Genius. One of our strategies was selected and participated in the international ANAC-2017 competition. Our thesis presents a detailed evaluation and analysis of the performance of all our agents.

Abstract (in Greek)

Σε διαπραγματεύσεις πάνω σε πολλαπλά ζητήματα, δύο οι περισσότεροι αυτόνομοι πράκτορες που μπορεί να μοιράζονται κοινά ή διαφορετικά συμφέροντα, διαπραγματεύονται πάνω σε κάθε διακριτό θέμα ξεχωριστά. Συνήθως αυτή η διαδικασία γίνεται υπό αβεβαιότητα σε σχέση με τα χαρακτηριστικά των υπόλοιπων μελών της διαπραγμάτευσης. Για αυτό το λόγο, οι πράκτορες πρέπει να είναι ικανοί καθ' όλη την διάρκεια της διαπραγμάτευσης να προσαρμόσουν την συμπεριφορά τους, υιοθετώντας αποδοτικές αυτόνομες τεχνικές διαπραγμάτευσης. Η πρόκληση σε αυτό, έγκειται στο ότι οι συμμετέχοντες στην διαπραγμάτευση δεν είναι συνήθως πρόθυμοι να αποκαλύψουν τις προτιμήσεις τους, έτσι ώστε να μην μπορούν οι υπόλοιποι να αποσπάσουν στοιχεία σχετικά με αυτούς και να τα εκμεταλλευτούν προς δικό τους όφελος.

Για να ξεπεραστούν οι παραπάνω δυσκολίες και να βελτιωθεί η απόδοση ενός πράκτορα, συνήθως η υπάρχουσα στρατηγική συνοδεύεται από μία μοντελοποίηση του αντιπάλου. Βέβαια αξίζει να σημειώσουμε ότι σε προβλήματα του πραγματικού κόσμου, ή προσομοιώσεις αυτών, η αβεβαιότητα σχετικά με τις προτιμήσεις του αντιπάλου πάντα υπάρχει. Ωστόσο η μοντελοποίηση του αντιπάλου μπορεί να χρησιμοποιηθεί βοηθητικά, για να επιτευχθούν καλύτερες συμφωνίες και συνεπώς να προκύψουν καλύτερα αποτελέσματα στην εκάστοτε διαπραγμάτευση. Ένα πλήθος στρατηγικών διαπραγμάτευσης και τεχνικών μοντελοποίησης του αντιπάλου έχουν εφευρεθεί και μελετηθεί με το πέρασμα των χρόνων, ωστόσο δεν είναι πάντα εύκολο να γίνει η εμπειρική σύγκρισή τους.

Μία λύση σε αυτό ήρθε να δώσει ο παγκόσμιος διαγωνισμός αυτόνομων πρακτόρων διαπραγμάτευσης (ANAC), ο οποίος ξεκίνησε να διεξάγεται το 2009. Για την κάλυψη των απαιτήσεων του διαγωνισμού, δημιουργήθηκε μία νέα πλατφόρμα διαπραγματεύσεων ("Genius"). Το Genius προσφέρει έναν καθολικό τρόπο σύγκρισης των σύγχρονων στρατηγικών διαπραγμάτευσης.

Στην παρούσα διπλωματική εργασία, υλοποιήθηκαν συστηματικά διάφορες στρατηγικές διαπραγμάτευσης και δοκιμάστηκαν συνοδευόμενες από διαφορετικές τεχνικές μοντελοποίησης των αντίπαλων παικτών. Υλοποιήθηκαν υπάρχουσες προσεγγίσεις από την βιβλιογραφία, καθώς και νέες που βασίστηκαν σε υπάρχουσες. Η διαδικασία αυτή οδήγησε στην υλοποίηση δεκατριών (13) διαφορετικών πρακτόρων. Στην συνέχεια οι πράκτορες αυτοί αξιολογήθηκαν μέσα από την διεξαγωγή πρωταθλημάτων (τουρνουά) διαπραγματεύσεων μεταξύ τους, αλλά και έχοντας ως αντιπάλους πράκτορες που είχαν λάβει μέρος στον αμέσως προηγούμενο (2015) διαγωνισμό του ANAC για τον οποίο διαθέταμε εκτελέσιμα συμμετεχόντων πρακτόρων. Μία από τις στρατηγικές που υλοποιήσαμε, επιλέχθηκε για συμμετοχή - και συμμετείχε - στον παγκόσμιο διαγωνισμό ANAC-2017. Στη διπλωματική, γίνεται λεπτομερής αξιολόγηση και ανάλυση της απόδοσης όλων των αυτόνομων πρακτόρων που υλοποιήθηκαν.

Glossary of Notation

U: Utility function

D: Domain

I_n: Issue n

b_t: bid in negotiation round t

${}^{n}v_c$: Value c of issue n

w_i: weight of issue i

$e_c(x_n)$: Evaluation function

H^w: set of Hypothesis for opponent's weights

h_j: Hypothesis j

n: Number of issues

r_i^j: rank of weight i in the hypothesis j (in Bayesian Opponent Model)

r_i: Greedy ratio in issue i

$P(h_j/b_t)$: Probability of hypothesis i given bid proposed in time t

$u'(b_t)$: Utility value of a bid in time t

$c(t)$: Concession tactic

σ: spread of the conditional distribution

$p(a)$: Probability of action a

$Q(s,a)$: Q value of action a in state s

τ: Temperature (Boltzmann strategy) or target utility (smart meta-strategy)

θ: Aspiration level

k: Greedy choice (or number of rounds in Conan Strategy)

$Offer_{t,s_i}$: The offer for agent $i$at time t

IP: Initial price of an offer

RP: Reservation price

CR_{t,s_i}: The concession rate for agent $i$at time $t \in [0,1]$

T_{start}: start time of negotiation,

T_{end}: Deadline of the negotiation

E_t: Environmental factors effects

S_t: Self factors effects

CO: Number of committed offers

NS: Negotiating status

E_g: Eagerness of agent

1. Introduction

Negotiation is a form of decision-making where two (during "bilateral" negotiations) or more (during "multilateral" negotiations) parties jointly search a space of possible solutions with their goal being reaching a consensus [1]. Negotiating situations, or *bargaining*, are (often) described in Game Theory [50] as zero-sum games where a shift in the value along a single dimension means that one side is better off and the other is worse off. Thus, the self-interest of a negotiating party may be captured by a utility function. In the negotiation process, one party tries to maximize its utility, while following a specific behavior pattern that is decided by an established strategy [51].

In accordance to other game-theoretic settings, one can refer to a negotiation as being *cooperative* or *non-cooperative,* considering the willingness of the negotiator to form coalitions (or not). Moreover, many different negotiating settings or protocols exist. A negotiation may be *bilateral* or *multilateral*, depending on the number of players participating in it. The negotiating players usually do not have to agree over a *single* issue, but in *many different* ones (*multi-issue* or *multi-attribute negotiations*). Each issue's negotiating outcome may have *different impact* on a negotiator's performance, since every player has a *different preference* over a specific issue. Information about the opponent, may also vary from one negotiation to another. Opponents' preferences can be *known*, *partially known* or even entirely *unknown*.

Overall, aspects that define the *outcome* of a negotiation are the number of the participants, simultaneous negotiation over multiple attributes. Time (or other related) *constraints* and the ability to employ *learning* through the negotiation process, are very important too. *Automated negotiations*, conducted by *autonomous agents,* are seen as the (only) way to meet the challenges encountered in highly complex real-world negotiation settings [52].

In this thesis, we study non-cooperative, multilateral, multi-attribute time-constrained negotiations. Since the information about the opponents must be gained through the negotiation process, we consider the negotiation to be "closed". Moreover, except time limitations themselves, we are dealing with discounted environments; the agent's profit when reaching an agreement is decreased while getting closer to the deadline. Finally, the alternate offers protocol is adopted; an agent can either modify an incoming offer or propose a counter offer, until an agreement is reached.

We methodically developed several different negotiating strategies, along with accompanying opponent models. To do so, we employed concepts found in the literature, implemented known strategies, and proposed novel ones. As a result, *thirteen (13)* distinct agents were created.

To evaluate our agents, we employed *Genius*[1] a "standard", negotiation platform developed for the international Autonomous Negotiation Agents Competition (ANAC)[2] . Genius is an asset in the negotiations research field, since it provides a uniform, standard way of comparing state-of-the-art agent strategies. Within Genius, our agents were systematically pitted against past ANAC participants, and also against each other. One of our agents was selected for participation, and eventually participated, in the international ANAC-2017 competition.

The remainder of this thesis is structured as follows. Chapter 2 lays the background for this thesis, outlining the connections of negotiations to game (and, in particular, bargaining) theory. The various aspects that define the "rules" and the outcome of a negotiation are explained there in detail as well. The Autonomous Negotiation Agents Competition, Genius and the settings of the competition are presented in Chapter 3.

[1] http://ii.tudelft.nl/genius/

[2] http://web.tuat.ac.jp/~katfuji/ANAC2017/

Furthermore, key aspects of opponent modeling are analyzed in Chapter 4, as well as two popular opponent modeling techniques: *frequency based* and *Bayesian*. These two techniques were in fact used to build the opponent models of several agents developed in this thesis.

The implementation and evaluation process is described in Chapters 5 and 6. The developed strategies are analyzed and tested against previous ANAC participants in Chapter 5, but also against each other in Chapter 6. Each strategy explores a different hypothesis about the negotiations' outcome in the given setting. Concepts existing in the literature have been implemented, such as Boltzmann exploration, Maximum greedy tradeoffs and Conan algorithm. Those techniques have been tested as are, or altered to match our framework. Also new hybrid strategies are introduced in this thesis, combining aspects from the above.

The best -regarding utility outcome-strategy has been selected to participate in the ANAC competition, the results of which are presented in Chapter 7.

Finally, Chapter 8 concludes this thesis and outlines future work.

2. Background

In this chapter, we present a general background for negotiations and their connection to game and consequently bargaining theory, aspects of bargaining such as cooperative and non-cooperative, as well as a classification based on the information provided in a negotiation. Different types of negotiation are presented, regarding the number of negotiation participants and the number of issues that are under negotiation. The importance of learning and using heuristics is analyzed, as also the time limitations and problems that occur during a negotiation.

2.1 Negotiations

Negotiation is an important process to form alliances and to reach trade agreements. Research in the field of negotiation originates from various disciplines including economics, social science, game theory and artificial intelligence [1] [2] [3]. Automated agents can be even used next to a human negotiator, embarking on an important negotiation task. They can facilitate the efforts required of people during negotiations and assist them in complex negotiation processes. There may even be situations in which automated negotiators can replace the human negotiators [4]. Another possibility is for people to use these agents as a training tool, prior to actually performing the task. Thus, success in developing an automated agent with negotiation capabilities has great advantages and implications.

In recent years there has been an increasing focus on the design of automated negotiators, i.e., autonomous agents capable of negotiating with other agents in a specific environment. Furthermore, many of these agents are designed to operate in specific and relatively simple scenarios, based on simplified assumptions that do not model well real-life settings where negotiation may be actually applied. For example, it is often assumed that the opponent strategies and preferences are known or partially known. This is unrealistic especially in multi-issue negotiations (i.e., reaching agreement over multiple issues simultaneously) [5], where agents can have different preferences for these issues, since they are unlikely to reveal them. Some work does consider settings with incomplete information [6] [7], but still assumes agents either to have partial information about the opponent's preferences, or probabilistic information about the opponents, which is again often not available in practice.

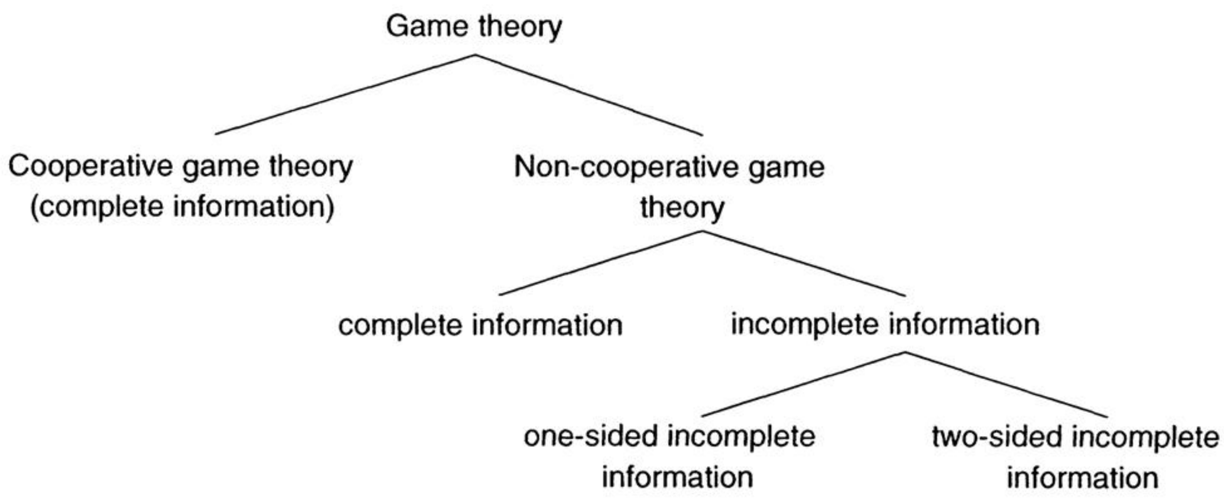
So, in designing proficient negotiating agents, standard game-theoretic approaches cannot be directly applied. Game theory models often assume complete information settings and perfect rationality. However, human behavior is diverse and cannot be captured by these models alone. Humans tend to make mistakes, and they are affected by cognitive, social and cultural factors [45].

For overcoming some of these limitations heuristic approaches can be used to design negotiating agents. When negotiating agents are designed using a heuristic method, an extensive evaluation is needed, typically through simulations and empirical analysis. Still, heuristic approaches are not always adequate, since an assumption made in the literature, is that negotiation interactions occur at fixed time intervals and that negotiation ends after a fixed number of rounds. In practice, however, negotiation usually happens in real time. In real-time negotiations, the time required to reach an agreement depends on the time agents need to process an offer (i.e., the amount of computation required to evaluate an offer and produce a counter offer). This is particularly important when utility is discounted, i.e. when the value of an agreement decreases over time, or when there is a deadline [8] [9] [10].

Furthermore, another problem in the negotiation field, is that many negotiation strategies are evaluated against relatively simple strategies, such as the time-dependent tactics, instead of the state-of-the-art. This is partly due to the fact that the results of the different implementations are difficult to compare, as different setups are used for experiments in negotiation environments.

2.2 Bargaining Theory

From the point of game theory [50], negotiations (or bargaining) can be modeled as a sequence. More specifically, bargaining theory is a part of game theory that studies bargaining games [11] [12]. We will use the same taxonomy of game theory shown in the figure below to organize bargaining theory as well [13].



Game theory can be divided into two branches: cooperative and non-cooperative game Theory [14]. Respectively, bargaining theory can be divided into cooperative and non-cooperative one.

2.2.1 Cooperative bargaining theory

Cooperative bargaining theory is concerned with the question of *what* binding agreement two bargainers would reach in an unspecified negotiation process given the set of all possible agreements on the utility that each bargainer achieves [13]. The classic work of Nash [15] provides a unique solution that satisfies four properties, which are called the "Nash axioms". The Nash axioms include:

1. The final outcome should not depend on how the players' utility scales are calibrated
2. The agreed payoff pair should always be individual rational and Pareto-efficient
3. The outcome should be independent of irrelevant alternatives
4. In symmetric situations, both players get the same utility.

Some other solution concepts in cooperative bargaining theory include the *Kalai-Smorodinsky bargaining solution* [16] and *weighted utilitarian (bargaining) solution* [17].

A cooperative bargaining model does not consider the negotiation process, but leaves the outcome to be determined by an axiom.

2.2.2 Non-Cooperative bargaining theory

Non-cooperative bargaining theory considers bargaining as a fully specified game. The game refers to the negotiation protocol that two players follow during the bargaining process. A negotiation protocol is the set of rules that govern the interaction between negotiators [46]. It covers the negotiation states (accepting proposals, negotiation closed), the events that cause negotiation states to change (no more bidders, bid accepted), and the valid actions of the participants in particular states (which messages can be sent by whom, to whom, at what stage) [18]. Based on the above, it is actually indicated that negotiation is an extensive-form game (Figure 1). In game theory extensive-form games are a specification of games that allow detailed representation of aspects in the game process, like the player's possible moves sequence at every different decision point, the possible outcomes these moves may have, or even the information about other players in order to make a decision.

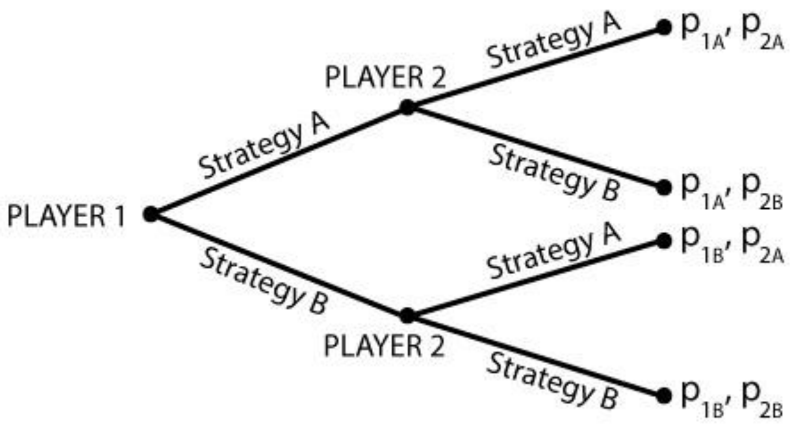


Figure 1. An example of an extensive-form game

The notion of an *equilibrium strategy* is usually used in this kind of games to define rational behavior of players, which jointly decide the outcome of a game. Strategy *equilibrium* is a profile of players' strategies so that no player could benefit by unilaterally deviating from his strategy in the profile, given that other players follow their strategies in the profile.

An equilibrium facilitates the prediction of the players' behavior, and hence also the outcome of the game. Some widely-used concepts of strategy equilibrium include "dominant strategy" equilibria, "Nash" equilibria and "sub-game perfect" equilibria. A *dominant strategy equilibrium* consists of a *dominant strategy* of each player, which is optimal for a player irrespective of the other players' strategies. The strategies chosen by all players are said to be in *Nash equilibrium* if no player can benefit by unilaterally changing his strategy. A sub-game perfect equilibrium refines the Nash equilibria in extensive-form games. In a sub-game *perfect equilibrium* (SPE) the strategies for each sub-game of the game tree constitute a Nash equilibrium [19].

2.2.3 Complete and Incomplete Information

A game is a *complete information* one if the preference profile (i.e. preference information) of a player is known to all other parties, otherwise it is called an *incomplete information* game. If both sides have private information, it is called a *two-sided incomplete information* game, otherwise if only one side has private information, it is called a *one-sided incomplete information* game [13].

For an incomplete information game, the equilibrium concept that is usually used is the "Bayes-Nash" equilibria. The strategies of players, which are associated with their private information, compose a *Bayes-Nash equilibrium* if no player can get higher benefit *on expectation* by unilaterally changing his strategy [20] [21]. Cooperative games are all based on complete information; it is assumed that the input to the axiomatic solution is common knowledge or that players share true information with each other [13]. In non-cooperative game theory players may withhold information or not be truthful with each other [22] [23].

2.3 Bilateral and Multilateral Negotiations

In the case of agent-to-agent negotiation, the agents should agree on the issues over which negotiation takes place. They should use the same established language and should follow the same rules or legitimate actions when interacting each other, namely the same protocol. Every agent must take into account the goals, the directives on what is authorized and not authorized to do, and the parameters influencing its behavior (e.g., time limits). The above apply in all negotiations, no matter how many parties are included in it.

A bilateral negotiation situation, is characterized by two agents who have a common interest in cooperation, but who have conflicting interests concerning the particular way of doing so [13]. Bilateral negotiation (or bargaining) refers to the corresponding attempt to resolve a bargaining situation; for example, to determine the particular form of cooperation and the corresponding payoffs for both negotiating participants [15].

An extended form of bilateral negotiation as described above, is multilateral negotiation, where more than two agents participate in the negotiation situation. Two types of processes are involved in multilateral negotiations, coalitional and consensual [24].

Forming coalitions, is the most broadly applicable way of simplifying, structuring and orienting multilateral negotiation, relating to both parties and issues [47]. Parties seek either to ensemble other parties into a coalition, or to divide potentially opposing groups that may have been formed into smaller parts, so as to absorb parties from them, or even just weaken them [24].

The other type of negotiation process is *consensusuation* [48], where the parties' reservation limits are ascertained beforehand; the proposal which manages to fall within those limits achieves acceptance without bargaining. Consensus is actually forming the grand coalition, a coalition including everyone participating in a multilateral negotiation. It is based on a decision rule under which “essentially abstention is an affirmative rather than a negative vote” [48].

Multilateral agreements arrive at a consensus when a coalition is formed by an unspecified but yet significant number of parties and the rest do not oppose. Parties not in agreement can abstain without blocking the outcome. In the case that opposing parties exist and they are not numerically significant, they can be left out [24].

2.4 Multi-attribute or Multi-Issue Negotiations

Multi-attribute negotiation is a negotiation that involves multiple issues and they need to be negotiated simultaneously. Usually it is characterized by the situations in which two or even more parties recognize that the differences of interest over multiple issues but also the value of cooperation exists between them and in which they want to seek a compromise agreement [25].

A multi-attribute negotiation is more complex and challenging than a single-attribute negotiation because of the following reasons [26]:

1. In a multi-attribute negotiation, the preference of an agent over multiple issues can be

complex. Preferences are modeled by a utility function (a mathematical formula) and agents make decisions based on this utility function. However, it is not trivial for a human to construct such a utility function over multiple issues, especially when preference over one issue have impact on the values of other issues.

2. The solution space is n-dimensional (n>1) rather than a single dimensional line as in a single-attribute negotiation. So, the necessity of designing a complex negotiation strategy appears: since the space is n-dimensional, every time an agent plans to concede, the direction of concession needs to be decided. Apparently, there are many choices on the concession direction; to concede on issue 1,…, n or different combinations of the issues. Moreover, the concession direction decision making, may also depends on the opponent's preference because conceding on an issue more important to the opponent can make the offer more alluring.
3. In multi-attribute negotiations "Win-Win" situations exist. For rational agents, the ideal result is to discover a Pareto-optimal (or Pareto-efficient) solution. A Pareto-optimal solution is one which cannot be improved further without sacrificing someone's utility; if there is another solution from which one of the agents can get more than this Pareto-optimal solution, then the other agent must get less by that other solution. We refer to a multi-attribute negotiation model as efficient, when agents will reach a Pareto-optimal agreement in the negotiation, if a zone of agreement exists. The collection of Pareto optimal bids is called the Pareto optimal frontier (Figure 2).

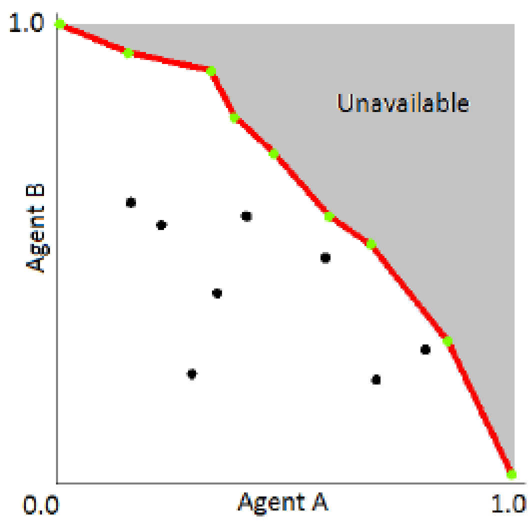


Figure 2: This is an example of a bid space. Dots represent the possible agreements between agent A and agent B. The red line is the Pareto frontier: the collection of pareto optimal bids

This kind of negotiation is referred to as "competitive", because since multiple criteria are involved, and players attach different importance to each criterion, outcomes that are better for both parties than the current offer may exist. This is called an "integrative" negotiation [13]. Still, not all multi-attribute negotiations are integrative; if the utility function of an agent is the weighted sum of all attributes, the negotiation can be equivalently transformed into multiple independent single-attribute negotiations, one for each attribute. This is possible because the utility associated to one attribute is independent of the values of the other attributes when the preference function is in a linear form.

2.5 Negotiation Heuristics

Usually, it is computationally cumbersome to search for a strategy equilibrium, especially when an analytical solution is not known. This is because of the fact that an equilibrium involves outguessing regress. *Outguessing regress* means an agent's decision depends on its beliefs about other agents and no accurate prediction or expectation can be made about the choices of others [13]. In reality agents have bounded rationality, that is, their computation and reasoning resource is limited; hence, playing a game following the equilibrium strategy is not necessary expected. Artificial intelligence (AI) approaches help the players locate an approximate solution strategy according to principles of bounded rationality by utilizing heuristic search and evaluation techniques [49]. Heuristics are "shortcuts" to reduce the complexity and effort involved in the reasoning process [27].

Faratin, [28] defines a range of computationally feasible heuristic strategies and tactics that can be employed to negotiating agents in order to generate offers or counteroffers and evaluate proposals in multi-attribute negotiations. The *tactics* are simple functions that are used to generate an offer, or counter offer, based on different criteria. [13]

A *strategy* [13] is the way in which an agent changes the weights of the different tactics over time. In the negotiation model agents propose offers alternatively following their strategies. Each agent has a scoring (i.e. evaluating) function that is used to rate the offers received. If an agent receives an offer that has value greater than the value of the counter offer that it is ready to submit in the next step (i.e. his threshold value), then it accepts. Otherwise, the counter offer is submitted.

The negotiation tactics include:

- The **time-dependent tactics,** an agent submits offers that change monotonically from the minimum (best) to the maximum (worst) of the deal; the rate of the increase is based on the time left until the end of the negotiation (i.e. the deadline) [13]. Faratin, [28] distinguish two families of changing rate functions, with the rate of change being a polynomial or an exponential function of time. The two functions model a Boulware or a conceder agent respectively. A *Boulware* agent does not concede until close to the last moment, while a *conceder* gives up quickly.
- The **resource-dependent tactics** are similar to the time-dependent ones, using the same functions but with the difference that resource-dependent tactics, have dynamic value of the maximum available resource, or the rate function is changed based on an estimation of the amount of a particular resource.
- The **behavior-dependent tactics** compute the next offer based on the previous attitude of the negotiation opponent. These tactics are especially important in cooperative problem-solving negotiation settings, since opponents' behavior must be taken into consideration [13].

2.6 Learning in Negotiations

Learning is a very important aspect in negotiations, because the participants have to deal with an unknown environment and usually an unknown opponent. That makes it necessary for knowledge to be acquired and learning techniques to be implemented. Learning is important because it can lead to successful conclusions about what is in best interest for an agent, enable him to behave flexibly and to adapt to environment changes.

So, learning can be used for two main purposes:

1. Observe and update beliefs about the preferences and behaviors of the other parties and adjust strategy accordingly.
2. Adjust a negotiation strategy as to achieve better deals, based on previous games/sessions

outcome or information gained.

These concepts are further analyzed in the Chapter 4 of the thesis, the chapter dealing with opponent modeling.

2.7 Time Issues in Negotiation

Time is a crucial factor in some negotiation formats. Generally speaking the most common effects of time [8] on the negotiation process are:

- Discounting: Benefits received immediately are preferred to the same benefits received in the future
- Bargaining cost: The bargaining process itself incurs some cost/utility to an agent
- Sudden termination: There is a hard deadline beyond which the negotiation cannot be continued or is useless

The need of pressure (time, resources, need to reach an agreement) in a negotiation process is necessary otherwise it would not be worthwhile to negotiation, because there would be no need/desire to reach an agreement [29]. By including specifically time pressure to a negotiation setting, the whole dynamics of negotiation changes. The agents do not have an unlimited time to negotiate, thus having to concede to some degree in order to reach an agreement [33].

As a consequence, a proposed strategy must be taking into account all possible time constraints to be efficient. Still, the knowledge of time limitations and especially the deadline of a negotiation is information that can be easily handled and be incorporated into a negotiation strategy.

3. The Automated Negotiating Agents Competition

The goals and purposes of the Automated Negotiating Agents Competition are briefly presented in this Chapter. Information is provided regarding the negotiation model used in the competition, as well as the challenges the participants face during the negotiation process. Finally, the Genius negotiation platform and the negotiation protocol that Genius uses are introduced.

3.1 General Design of ANAC

The automated negotiating agents' competition (ANAC) has been organized in order to help focus research on negotiating automated agents. The purpose of the competition [30] is to facilitate the research in the bilateral and multilateral multi-issue closed negotiation area. The principal goals of the ANAC competition are as follows [30]:

- Encouraging the design of agents that can proficiently negotiate in a variety of circumstances
- Objectively evaluating different bargaining strategies
- Exploring different learning and adaptation strategies and opponent models
- Collecting state-of-the-art negotiating agents, negotiation domains, and preference profiles, and making them available and accessible for the negotiation research community.

ANAC encourages the design of agents that can negotiate in a variety of circumstances. This means the agents should be able to negotiate against any type of opponent within arbitrary domains. Examples of such environments include online markets, patient care-delivery systems, virtual reality and simulation systems used for training. The use of open environments is important as the automated agent needs to be able to interact with different types of opponents, who have different characteristics.

Overall, the setup of ANAC was designed to make a balance between several concerns, including:

- Strategic challenge: the game should present difficult negotiation domains in a real-world setting with real-time deadlines.
- Multiplicity of issues on different domains, with a priori unknown opponent preferences.
- Realism: realistic domains with varying opponent preferences.
- Clarity of rules, negotiation protocols, and agent implementation details.

3.2 Negotiation Model

The parties negotiate over issues, and every issue has an associated range of alternatives
or values. A negotiation outcome consists of a mapping of every issue to a value, and the set of all possible outcomes is called the negotiation domain. The domain is common knowledge to the negotiating parties and stays fixed during a single negotiation session.

It is further assumed that both parties have certain preferences prescribed by a preference profile. These preferences can be modeled by means of a utility function U that maps a possible outcome to a real-valued number in the range [0, 1]. In contrast to the domain, the preference profile of the player is private information.
Finally, the interaction between negotiating parties is regulated by a negotiation protocol that defines the rules of how and when proposals can be exchanged.

3.3 Domains and Preference profiles

The specifications of the domain and preferences, such as the constitution and valuation of issues, can be of great influence on the negotiation outcome. We assume that all agents have complete information about the domain, but the preference profile of the player is private information. Thus, if a strategy attempts to tailor its offers to the needs of the opponent, it is required to model the opponent. As the amount of information exchanged during the negotiation is limited in a closed negotiation, the size of the domain has a big impact on the learning capabilities of the agents.

Due to the sensitivity to the domain specifics, negotiation strategies have to be assessed on negotiation domains (Figure 3) of various sizes and of various complexities. Therefore, several domains with different characteristics are selected in the competition.

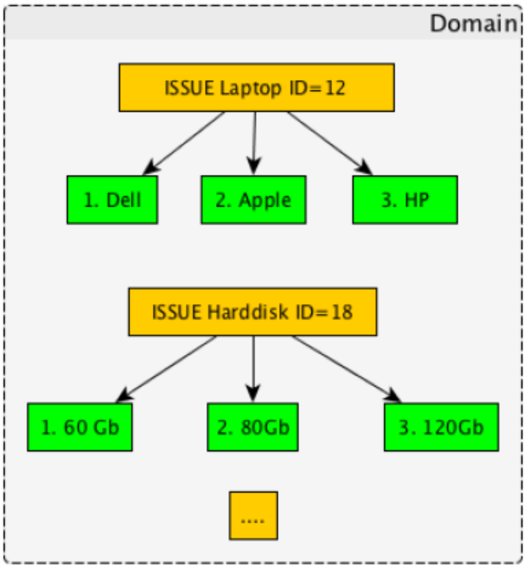


Figure 3: An example domain for laptop negotiation

The Domain describes which issues are the subject of the negotiation and which values an issue can attain [31]. A domain contains n issues (1):

$$D = (I_1, \ldots., I_n) \qquad (1)$$

Each issue i consists of k values (2):

$$I_i = ({}^{i}v_1, \ldots., {}^{i}v_k) \qquad (2)$$

Combining these concepts, an agent can formulate a Bid: a mapping from each issue to a chosen value (3):

$$b = ({}^{i}v_c, \ldots., {}^{n}v_c) \qquad (3)$$

The Utility Space specifies the preferences of the bids for an agent using an evaluator. It is basically just a function that maps bids into a real number in the range [0,1] where 0 is the minimum utility and 1 is the maximum utility of a bid. A common form of the Utility space is the Additive Utility Space. Such a space is additive because each of the issues in the domain have their own utility of their own. Figure 4 shows a picture of a utility space for the example domain that we gave in Figure 3.

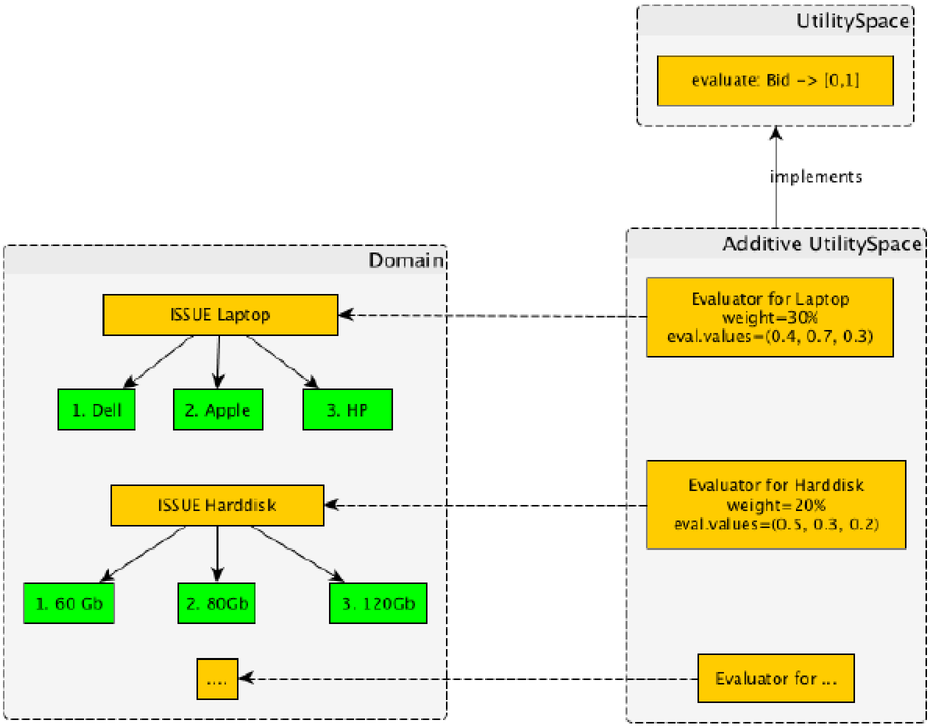


Figure 4: An example additive utility space for the laptop domain

In an additive space the evaluator also specifies the importance of the issue relative to the other issues in the form of a weight. The weights of all issues sum up to 1.0 to simplify calculating the utility of a bid. The utility is the weighted sum of the scaled evaluation values as in Eq. 4.

$$U(v_c^i, \dots, v_c^n) = \sum_{i=1}^{n} w_i \frac{eval(v_c^i)}{\max(eval(I_i))} \qquad (4)$$

Negotiation strategies can also depend on whether preferences of the negotiating parties are opposed or not. The notion of weak and strong opposition can be formally defined. *Strong opposition* is typical of competitive domains, when a gain for one party can be achieved only at a loss for the other party. Conversely, *weak opposition* means that both parties achieve either losses or gains simultaneously. Negotiation strategies may depend on the opposition of the preferences.

3.4 Challenges of the competition

The selection of real-life domains containing multiple issues and preference profiles that are unknown to the opponent provides a challenging framework for agents to compete. The main problems an agent faces in the competition are the following:

- Incomplete information: Suppose an agent wants to model the utility function of the opponent. Because of incompleteness of information about the opponent, it has to learn the opponent's preferences during the negotiation by studying the proposal behavior. The protocol only allows proposals to be exchanged, so the communication between the agents is very limited. This prevents agents to share information about their preferences, other than by their proposal behavior. Consequently, if agents want to use an opponent model to make effective proposals, they have to make use of a sophisticated learning technique.

- Real-time deadline: Dealing with all of the above while taking into account a real-time deadline. Agents should be more willing to concede near the deadline, as a non-agreement final result, yields zero utility for both agents. A real-time deadline also makes it necessary to employ a strategy to decide when to accept an offer. Deciding when to accept involves some prediction whether or not a significantly better opportunity might occur in the future.

Some parts of ANAC are less demanding: the utility functions are (linearly) additive functions, so there are no dependencies between issues. This means the utility of an offer can be effectively computed and conversely, a proposal of a certain utility can be easily generated. Moreover, additive utility functions make it fairly easy to enumerate proposals from best to worst. Secondly, agents are completely aware of their own preferences and the corresponding utility values.

3.5 The Genius Benchmark

A number of successful negotiation strategies already exist in the literature, however, the results of the different implementations are difficult to compare, as various setups are used for experiments in negotiation environments. An additional goal of ANAC is to build a community in which work on negotiating agents can be compared by standardized negotiation benchmarks to evaluate the performance of both new and existing agents. So, an environment that allows evaluating agents in a negotiation competition has been introduced.

Genius[3] is a General Environment for Negotiation with Intelligent multi-purpose Usage Simulation. Genius helps facilitating the design and evaluation of automated negotiators' strategies. It allows easy development and integration of existing negotiating agents, and can be used to simulate individual negotiation sessions, as well as tournaments between negotiating agents in various negotiation scenarios.

The design of general automated agents that can negotiate proficiently is a challenging task, as the designer must consider different possible environments and constraints. Genius can assist in this task, by allowing the specification of different negotiation domains and preference profiles by means of a graphical user interface. It can be used to train human negotiators by means of negotiations against automated agents or other people, to teach or to provide insights of generic automated negotiating agents. Additionally, it provides an easily accessible framework to develop negotiating agents via a public API. This setup makes it straightforward to implement an agent and to focus on the development of strategies that work in a general environment.

Genius incorporates several mechanisms that aim to support the design of a general automated negotiator. The first mechanism is an analytical tool-box, which provides a variety of tools to analyze the performance of agents, the outcome of the negotiation and its dynamics. The second mechanism is a repository of domains and utility functions. Lastly, it also comprises repositories of automated negotiators. Also, Genius enables the evaluation of different strategies used by automated agents that were designed using the tool. This is an important contribution as it allows researchers to empirically and objectively compare their agents with others in different domains and settings.

[3] http://ii.tudelft.nl/genius/

3.6 Stacked Alternating Offers Protocol (SAOP)

The SAOP protocol has been introduced and used in the ANAC competition [32]. According to this protocol, all of the participants around the table get a turn per round; turns are taken clock-wise around the table, also known as a Round Robin schedule. One of the negotiating parties starts the negotiation with an offer that is observed by all others immediately. Whenever an offer is made, the next party in line can take the following actions:

1. Accept the offer
2. Make a counter offer (thus rejecting and overriding the previous offer)
3. Walk away (thereby ending the negotiation without any agreement)

This process is repeated in a turn-taking clock-wise fashion until reaching a termination condition is met. The termination condition is met, if a unanimous agreement or a deadline is reached, or if one of the negotiating parties ends the negotiation.

4. Opponent Modeling in Negotiations

As mentioned earlier, learning during a negotiation is crucial, especially since negotiation is typically an incomplete information game. The participants do not know their opponent's preferences or strategy, information that if it was known, it could help the agent himself improve his strategy and take into account the opponent's behavior to the formation of a proposal. Therefore, in order to reach better and earlier agreements, an agent can apply learning techniques to construct a model of the opponent. Usually negotiators are unwilling to share information in competitive situations to avoid exploitation by the other party. 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.

Having a good opponent model is a key factor in improving the quality of the negotiation outcome [33] and can further increase the benefits of automated negotiation, including the following: reaching win-win agreements, minimizing negotiation cost by avoiding non-agreement and finally, avoiding exploitation by adapting to the opponent's behavior during the negotiation. Experiments have shown that by employing opponent models, automated agents can reach more efficient outcomes than human negotiators [34] [35]. In the rest of this chapter we discuss opponent strategies in ANAC, along with challenges faced (such as exploration). We also present the opponent models used in our approach.

4.1 Classification of Learning Strategies

Learning Strategies are distinguished by the amount of inference the learner performs on the information provided. A suggested taxonomy of learning techniques [36] is summarized below, capturing the amount of effort required by the learner.

1. **Rote learning and direct implanting of new knowledge** – No inference or other transformation of the knowledge is required on the part of the learner.
2. **Learning from instruction** – Acquiring knowledge from a teacher or another organized source (i.e. a textbook)
3. **Learning by analogy -** Acquiring new facts or skills by transforming and augmenting existing knowledge that bears strong similarity to the desired new concept or skill into a form effectively useful in the new situation.
4. **Learning of examples** – Given a set of examples and counterexamples of a concept, the learner induces a general concept description that describes all of the positive examples and none of the counterexamples.
5. **Learning from observation and discovery (Unsupervised Learning)** – A general form of inductive learning that includes discovery systems, theory-formation tasks, the creation of classification criteria to form taxonomic hierarchies and similar tasks without benefit of an external teacher.

4.2 Online and Offline Learning

Learning techniques can be applied in different ways, depending on the information accessible in a specific format. The most distinguished models are online and offline learning models. In online learning, data becomes available in a sequential manner with respect to time and usually model parameters are constantly updated in a manner that allows to predict the future based on what actually ended up happening. On the other hand, in offline learning models the sequence of elements is known to the learner in advance, usually by a training period or a fixed dataset.

4.3 Exploration vs Exploitation

A challenge in the area of learning is the exploration – exploitation trade off [37]. Agents usually go through the same environment many times in order to learn optimal actions. Balancing exploration and exploitation is particularly important since an agent may have found a goal on one path, but there may be a better one on another path. Without exploration, the agent will always return to first option, and the better one will never be perceived. It is even possible for the goal to lie behind low reward areas that the agent would avoid without exploration. On the other hand, if the agent explores too much, there is a risk of not being able to stick to a path. In that case the agent is not really learning because it cannot exploit its knowledge, so acts as there is no prior information. Thus, it is important to find a good balance between the two, to ensure that the agent is learning but in the same time taking the optimal actions.

There are many types of strategies that achieve a balance between exploration and exploitation [37], with the most basic being *State–action value updating strategies* (e.g. Sarsa learning, Q-learning) and *Action selection strategies* (e.g. ε-greedy, Boltzmann). In both types, previous actions of the agent that lead to a good outcome are taken into account, and determine the following action. This knowledge gained by the agent is represented by a value. The main difference between these strategies, is that each one of them, determines this value in a different way.

Since we wanted to incorporate exploration and exploitation to some of the agents developed in this thesis, we decided to use Boltzmann exploration, since it appears to be very efficient, compared to other methods [37]. The Boltzmann distribution is defined in Eq. (5):

$$p(a) = \frac{\mathrm{e}^{\frac{Q[s,a]}{\tau}}}{\sum \alpha \cdot \mathrm{e}^{Q\frac{[s,a]}{\tau}}} \qquad (5)$$

Thus, Boltzmann exploration prescribes that an action a s selected with $p(a)$ probability. Notice that $p(a)$ depends on the perceived quality of executing action a at state s as captured in $Q(s,a)$ values maintained by the agent; while τ is a "temperature" factor, used essentially to specify a degree of randomization in action selection [37].

Intuitively, the use of temperature value signifies that, when τ is high we are in a phase of exploration, while when it is low we are in a phase of exploitation. **In general: if τ decreases higher actions with higher Q-values are chosen. Also,** when τ → 0 the best action–value is chosen.

4.4 Problems and Rationality in Opponent Learning in ANAC

As mentioned earlier, the use of an opponent model to learn opponents' preferences or strategy is almost inevitable, concerned closed negotiations under uncertainty, since no prior knowledge exists. Still, there are cases [38] that it does not always guarantee a better outcome for an agent and it may be preferable not to employ one at all:

- The model can be a poor representation of the opponent's preferences
- If the model consistently suggests unattractive bids for the opponent
- Learning the model can be computationally expensive and can therefore influence the number of bids that can be explored
- Assumptions about the opponent that do not reflect reality well

Concerning the 4th case, a set of assumptions has been tested and proposed [38] for use in opponent modeling in the ANAC setting:

1. ***The concession of the opponent follows a particular function.*** Some opponent modeling techniques assume that the opponent uses a given time-based bidding strategy.

Modeling the opponent then reduces to estimating all issue weights such that the predicted utility by the modeled preference profile is close to the assumed utility.

2. ***The first bid made by the opponent is the most preferred bid.*** The best bid is the selection of the most preferred value for each issue, and thereby immediately reveals which values are the best for each issue. Many agents start with the best bid.
3. ***There is a direct relation between the preference of an issue and the times its value is significantly changed.*** To learn the issue weights, some models assume that the amount of times the value of an issue is changed is an indicator for the importance of the issue. The validity of this assumption depends on the distribution of the issue and value weights of the opponent's preference profile and its bidding strategy.
4. ***There is a direct relation between the preference of a value and the frequency it is offered.*** A common assumption to learn the value weights is to assume that values that are more preferred are offered more often. Similar to the issue weights assumption, this assumption strongly depends on the agent's strategy and domain.

Furthermore, the main factors of how a negotiating scenario influences an opponent model are revealed [38]:

1. ***Domain size***. In general, the larger the domain, the less likely a bid is a Pareto-bid. Furthermore, domains with more bids are likely more computationally expensive to model. Thus, the influence of the time/exploration trade-off is higher.
2. ***Bid distribution***. The bid distribution quantifies how bids are distributed. We define bid distribution as the average distance of all bids to the nearest Pareto-bid. The bid distribution directly influences the performance gain attainable by a model.
3. ***Opposition***. We define opposition as the distance from the Kalai-point to complete satisfaction. The opposition of a domain influences the number of possible agreements, and opponent models may be help in locating them more easily.

4.5 Choosing an Opponent Model for ANAC

In the ANAC competition, each agent is described by a preference profile; a different one for every domain. This preference profile represents the private valuation of possible negotiation outcomes. Learning the preference profile assists in locating mutually beneficial outcomes and recognizing potential for meaningful concessions. Four approaches have been used so far to estimate the opponent's preference information [39]:

1. **Importance of the issues.** It is often easier to estimate the weight of all issues under negotiation, rather than the preference over all outcomes. The idea is to analyze the opponent's concessions, assuming that stronger concessions are made on issues that are valued less.

2. **Classify the negotiation trace**. Given the opponent's negotiation actions, we can determine which opponent type is most likely, and subsequently apply a classification algorithm to categorize preferences of the opponent.

3. **Aggregate negotiation data.** When offline data is available, we can derive the opponent's preference pro- file from a large database of negotiation traces from similar – but not identical – opponents.

4. **Importance of outcomes**. A popular technique is the frequency analysis heuristic. The main idea is that preferred values of an issue are offered relatively more often in a negotiation trace. For the issue weights, the opposite holds: an issue is likely unimportant if its value changes often.

Between different approaches to opponent modeling, two appear to prominent: *Bayesian opponent models* and *Frequency based models*.

- *Bayesian opponent models* generate hypotheses about the opponent's preferences [40]. The models presuppose that the opponent's strategy adheres to a specific decision function; for example, a time-dependent strategy with a linear concession speed. This is then used to update the hypotheses using Bayesian learning.
- *Frequency based models* learn the issue and value weights separately [38] [55]. The issue weights are usually calculated based on the frequency that an issue *changes* between two offers. The value weights are often calculated based on the frequency of *appearance* in offers.

Still both modeling approaches have disadvantages. More specifically, Bayesian models make strong assumptions about the opponent's strategy, whereas frequency based models assume knowledge about the value distribution of the issues of a preference profile and place weak restrictions on the opponent's negotiation strategy. Generally, the Bayesian models are far more computationally expensive; however, it is unknown if they are more accurate.

Based on the above, we decided to include and test two different opponent models to accompany the strategies tested in this thesis, a Bayesian opponent model [40] and a frequency one [41], along with the necessary alterations needed to much our multilateral framework.

4.5.1 A Frequency-based Opponent Model

The Agent Gahboninho [41] was first introduced in the ANAC 2011 competition. The underlying strategy it implemented follows the assumption it will be matched with other automated agents which vary in their compromising level. Thus, it tries to tackle strong opposition by putting pressure on the opponent. The stubbornness of the agent is also balanced based on the behavior of its opponent in order to achieve higher utilities.

This agent uses a proficiently structured frequency opponent model to learn the preference profile of the opponent (weights and values per negotiating issue). The opponent model updates agent's beliefs every time the opponent makes an offer (bid). As the updates continue through the negotiation, a firm belief, about weights and values of the opponent in every issue, is formed.

We chose to use Gahboninho opponent model, between other ANAC participants in this thesis, for the following reasons. First of all, this agent was ranked 1st in the qualification rounds and 2nd at the finals of ANAC '11, so overall is a skillful agent. Note that the ANAC '11 setting was for bilateral negotiation while in this thesis we study multilateral negotiation. Still, the specific negotiation protocol we use [32] has similar form in both bilateral and multilateral negotiations. Actually, the second is an extended form of the first one. That allows us to use the existing opponent model. Moreover, Gahboninho opponent model has been previously chosen to be tested [38] and appeared to be very efficient compared to other ANAC participant agents' profiles.

4.5.2 A Bayesian Opponent Model

We also incorporated in our algorithms a Bayesian Learning algorithm that was proposed in [40], in order to effectively learn the preferences of an opponent in multi-issue negotiations from bid exchanges in negotiations with imperfect information. The assumptions used in this model are generic with the most important being the assumption about issue independence and a rationality one that assumes agents use a concession-based strategy.

We assume that utility functions modeling the preferences of an agent, are linearly additive functions and are defined by a set of weights w_i (or priorities) and corresponding evaluation functions $e_i(x_i)$ for each of n issues as described in Eq. 6:

$$u(b_t) = \sum_{i=1}^{n} w_i \, e_i(x_i \in b_t) \qquad (6)$$

Utility function has a range in [0,1], the range of the evaluation functions is in [0,1] and the weights are normalized such that their sum equals 1.

In order to learn an opponent's preference profile or utility function U we need to learn both the issue priorities or weights w_i as well as the evaluation functions $e_i(x_i)$.

A set of hypotheses H^w about the private weights of an opponent as the set of all possible rankings of weights is defined [40]. Then real-valued numbers are associated with a $h_j \in H^w$ about weights, which can be computed as a linear function of the rank and also ensures weights are normalized (Eq. 7).

$$w_i = 2\frac{r_i^j}{n(n+1)} \qquad (7)$$

Where r_i^j is the reverse rank of a hypothesis' value about a specific weight w_i in the hypothesis h_j and n is the number of issues.

An additional structure is imposed on the evaluation functions in order to be able to learn a preference profile. A hypothesis space of predefined function types is introduced, in order to facilitate the learning of an opponent's preferences over issue values. This is done based on the shape of three type of functions that model assumptions about preferences over issues:

- *downhill shape*: minimal issue values are preferred over other issue values and the evaluation of issue values decreases linearly when the value of the issue increases
- *uphill shape*: maximal issue values are preferred over other issue values and the evaluation of issue values increases linearly when the value of the issue increases; -
- *triangular shape*: a specific issue value somewhere in the issue range is valued most and evaluations associated with issues to the left ("smaller") and right ("bigger") of this issue value linearly decrease.

A probability distribution is associated with each hypothesis; this allows other types of functions to be approximated by associating different probabilities with various hypotheses. Based on the probability distributions, different types of evaluating functions can result. The evaluating function does not have to exactly match the evaluation function existing in the hypothesis table, but it can be an approximation of one or a composition of more than one.

Hypotheses' probability distributions are going to be updated every time a bid is received from the opponent using Bayes' Rule (Eq. 8).

$$P(h_j/b_t) = \frac{P(h_j)P(b_t/h_j)}{\sum_{k=1}^{m} P(h_k)P(b_t/h_k)} \qquad (8)$$

Where the conditional probability $P(b_t/h_j)$ represents the probability that bid b_t might have been proposed given hypothesis h_j and $P(h_j)$ is the current probability of hypothesis h_j. Note that the entire hypothesis space is 1.

Still, $P(b_t/h_j)$ needs to be computed in order to be used in Bayes' rule. So, information about the utility the opponent associates with bid is needed. Since this kind of information is not available, it is assumed that opponent has a concession-based strategy and especially a time-dependent one, where agent starts with the maximal utility bid and moves towards its reservation value when approaching deadline. This kind of strategy can be described by a monotonically decreasing function. All the above actually describe the rationality assumption and allow to model it as a probability distribution associated with a range of tactics; as a result, each utility associated with an opponent's bid thus also has an associated probability.

Eq. 9 is used to model the conditional distribution:

$$P\left(b_t/h_j\right) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(u(b_t/h_j) - u'(b_t))^2}{2\sigma^2}} \qquad (9)$$

The predicted utility of the next opponent's bid $u'(b_t)$ is estimated as: $u'(b_t) = u'(b_{t-1}) - c(t)$, where $c(t)$ is the most plausible model of the opponent's negotiation concession tactic and $u'(b_t)$ is a linear function $u'(b_t) = 1 - 0.05t$.

Also σ denotes the spread of the conditional distribution used in the above function and defines the certainty of the agent about the opponent's negotiation tactics. If an agent is certain about the utility of an opponent's bid but thenσcan be set to a low value. A higher level of certainty increases the learning speed, since hypotheses predicting an incorrect utility value of a bid in that case would get assigned an increasingly lower probability, and vice versa. In our setting σ was set to 0.5 as we are completely agnostic regarding the opponent's bid.

5. Developing Our Agents

The purpose of this thesis is to develop and explore different negotiation strategies – existing and new ones, accompanied with the two opponent models mentioned in Chapter 4, in order to evaluate, compare them to each other and discover the most efficient among them. Except the implemented algorithms different assumptions about the negotiation outcome have also been tested through the evaluation and experimental process.

The benchmark of Genius has been used for the experimental procedures and all the agents have been implemented according to the ANAC rules and restrictions, since a second purpose of the thesis was to actually select one of those agents to compete in the ANAC '17 competition.

The development of the agents was done methodically, by progressively equipping them with negotiation strategies and opponent models found in the literature, or by adopting aspects of them to create new strategies. Every agent's performance was separately evaluated in actual tournaments following the ANAC rules. This process led to the creation of 13 autonomous agents presented in Table 1. This chapter describes the systematic process we used to develop those agents.

Agents
1. Boltzmann Strategy without opponent model
2. Boltzmann Strategy with Gahbonihno opponent model, modeling opponents as one
3. Boltzmann Strategy with Gahbonihno opponent model, modeling separately the opponents
4. Boltzmann Strategy with Bayesian opponent model, modeling separately the opponents
5. Smart Meta-Strategy with Bayesian Opponent Model
6. Smart Meta-Strategy with Gahbonihno Opponent Model
7. Hybrid MGT - Smart Meta Strategy Agent with Bayesian Opponent Model
8. Hybrid MGT - Smart Meta Strategy Agent with Gahbonihno Opponent Model
9. Conan Strategy Agent
10. Conan Strategy with use of opponents' weights and Bayesian Opponent Model
11. Conan Strategy with use of opponents' weights and Gahbonihno Opponent Model
12. Conan Strategy incorporated in Hybrid MGT - Meta Strategy with Bayesian Opponent Model
13. Conan Strategy incorporated in Hybrid MGT - Meta Strategy with Gahbonihno Opponent Model

Table 1: List with all the implemented agents

For choosing the opponents, agents from ANAC 2015 (the top 3 among them) where used, since it was the most recent available ones at the time the thesis started. Each one of the agents participated in tournaments against those agents. The opponent agents are presented in Table 2.

Opponent Agents	
Atlas3	1^{st} in ANAC '15
ParsAgent	2^{nd} in ANAC '15
RandomDance	3^{rd} in ANAC '15
Sengoku	
AgentH	
PokerFace	
Mercury	
AresParty	

Table 2: ANAC '15 agents used as opponents for the evaluation process

It must be noted that the original thought was to choose only ANAC winners as opponents (ANAC is held since 2010); this unfortunately was not possible since earlier ANAC competitions were for bilateral negotiation, so the agents followed another communication protocol that made them ineligible to be used in multilateral negotiations.

Regarding the agents' evaluation, it was considered prudent to conduct two kinds of tournaments for every agent. One with all the opponent agents on Table 2 and a smaller one including only the top three of ANAC'15, in order to observe how the agents' performance change according to the number of the negotiation participants.

The negotiation domains used, were also from previous ANAC competitions and were chosen based on their complexity, number of attributes and discount factor; thus, diversity is ensured and agents can be evaluated under different circumstances. In total 10 domains were used; each of them including various preference profiles. Analytical information about the domains is presented in Table 3. Furthermore, to achieve accurate results, every agent participated in 3 tournaments per domain.

During the whole experimental process described in this chapter, 78 tournaments and approximately 906.048 rounds were held. Those numbers do not refer to the tournaments held between the implemented agents themselves (i.e. those described in Chapter 6 below), but just with ANAC '15 participants as opponents.

Domain	Profiles	Discount	Opposition	Reserv. Value
Party	4	0	medium	0
Triangular Fight	3	0,1	low	0,3
Smart Grid	3	0	medium	0
University	4	0	medium	0
Japan Trip	3	0,9	medium	0,9
Domain Ace	3	0,2	low	0,3
K Domain	3	0,7	strong	0,2
Symposium	6	0	medium	0
Electric Vehicle	3	0,3	strong	0,5
Bank Robbery	5	0	strong	0,3

Table 3: Domains used in the tournaments held. Reservation value refers to the minimum value agents get when deadline is reached and an agreement has not been made

Since the Genius framework has been used, agents were established and initialized based on the requirements of Genius' version available at the time. Tournaments involve the negotiation of the agents in all of the domains in Table 3. Each domain has different preference profiles that are symbolized by a utility function as mentioned in Chapter 3. Thus, all the participant agents negotiated with all the different possible preference profiles and turns. Every session is considered to be in an unknown environment for the agent since the information about the opponents (i.e. preferences) and the domain itself is not allowed to be carried through sessions. Every time the agent competes, information about which domain he negotiates in, his preferences etc. must be gained from learning or other immediate techniques (e.g. random sampling in the utility space). The only bid in the utility space that is directly accessible by the agent in the beginning of the negotiation, is the one which provides him the maximum utility outcome.

To learn the utility space, our agents used a utility-space (i.e. domain) size-dependent technique.

The size of the utility space actually represents the number of possible bids existing in a domain. The more issues are under negotiation and the more different values these issues have, the number of possible bids increases, consequently the utility space increases as well. Note that our setting required only discrete value domain issues to be used.

Based on the above, in small utility spaces the agents collected all the possible bids at the beginning of the session; agents were searching throughout the utility space and then sorted possible bids based on the utility outcome. In big utility spaces this was impossible, since the agents are supposed to act in a specific timeline and this process is very computational and time expensive. So, the agents either did not manage to search the whole space or submit an offer. This situation created a bug and the whole negotiation was shut down by the system. So, a more efficient way of learning the utility space had to be implemented. The agents, managed their timeline, in order to save enough time to decide their move and in the time left were searching the utility space, sampling as many values as they could randomly and sorting them accordingly. Then in each round, the new bids offered by the opponents were examined; if they did not already exist in the registered bids, they were added sorted.

The Genius platform allows user to custom-select the timeline of the negotiation and the number of agents participating in it per round. In our implementation the ANAC settings were used; each round lasts 180 seconds and information about the remaining time agent has to make his offer, is provided to the agent in normalized form ($t \in [0,1]$) by Genius during the negotiation. Furthermore, only three agents participated per round; a tournament ended when all the agents have played with one another, with all different preference profiles and turns.

Regarding the agent's architecture, a basic structure that manages the negotiation protocol is provided by the Genius platform in order to ensure the agents' compatibility with the benchmark. There the developer can incorporate the desirable strategy and opponent model developed in Java language.

The same framework described above was used in all implementations; a slight change (i.e. one different opponent agent), may affected in different ways the outcome of the negotiation, since the whole process is not static, but depends from many different factors. So, we wanted to reserve as much parameters as possible fixed, in order to have a fair comparison between the strategies. All the following sections, demonstrate the developed strategies along with the theoretical concepts that were used. Tournaments' summarized results are also provided (the analytical results per domain can be found in appendix in Chapter 9), as well as comments and conclusions regarding agents' evaluation.

5.1 Boltzmann Strategy without opponent model

The purpose of this implementation, was mainly the familiarization with the concept of negotiations and the tournament setting. A rather simple strategy was developed and tested. We did not use an opponent model, in order to have a point of reference to compare the results with the rest of the agents developed and observe whether an opponent model makes a noticeable difference in the negotiation process and final outcome.

5.1.1 The Strategy

The strategy of this agent is founded on the concept of Boltzmann exploration; a common way to trade off exploration and exploitation. This algorithm has been chosen because of its capability to explore widely the possible action's effects and not only the ones who guarantee a good outcome [42].

The Boltzmann probability defined in Eq. 5 in Section 4.3 was incorporated in the acceptance and offer strategy as described below. The agent accepts a bid when the proposed utility is bigger or equal to a threshold, that represents the lower bound of the possible bids he will accept. The threshold (Eq. 10) depends from the Boltzmann Probability and the bids in the utility space that have better outcome for the agent in descending order (next best bid). Only one exception is made in the previous rule; the agent always accepts if recommended utility is bigger than 0.95.

$$Threshold = Boltzmann\ Probability * MyNextBid \qquad (10)$$

We define two possible states for the agent (Eq. 11):

$$\boldsymbol{state} = \boldsymbol{accept} \text{ or } \boldsymbol{state} = \boldsymbol{decline}\ (\boldsymbol{counter\ offer}) \qquad (11)$$

This means that the agent can either accept an offer or decline it and make a counteroffer instead. Q values for a utility outcome when state = decline is computed in every round. So, the total probability to decline a given utility regardless the bid itself it is recursively built; bids with same utility outcome for the agent, are “grouped” in the same probability. In other words, **agent does not care about the bid itself but only for the outcome**.

So, the probability to accept an offer is (Eq. 12):

$$\boldsymbol{P}(\boldsymbol{accept}) = 1 - \boldsymbol{P}(\boldsymbol{decline}) \qquad (12)$$

When round starts τ (temperature) factor (Eq. 13):

$$\boldsymbol{\tau = 0.01 \rightarrow \text{very close to } 0} \qquad (13)$$

This affects Boltzmann probability in the way described in Eq. 14 and Eq. 15.

$$\boldsymbol{P}(\boldsymbol{decline}) \rightarrow \textbf{high} \qquad (14)$$
$$\boldsymbol{P}(\boldsymbol{accept}) \rightarrow \textbf{low} \qquad (15)$$

So, when the game starts, the agent is unlikely to accept an offer. Agent starts to concede when getting closer to the deadline, or -in discounted domains- in an earlier stage defined in Eq. 16.

$$\boldsymbol{deadline * discountFactor} \qquad (16)$$

Regarding temperature,

$$\tau \in \mathbb{R}: \{0.01 < \tau < 0.25\}$$

with τ = 0.25 nearly before the game ends.

5.1.2 Results and Conclusions

In this first trial, the “Boltzmann Agent without opponent model” (or, BoltzmannAgent for short) actually managed to win in the small tournament and reach 4^{th} place at the big tournament. In more detail he won at 1 domain in the big tournament and 4 in the small one and the overall utility outcome sums up to 6,61 and 7,11 respectively (Table 4 & 5). It appears that the agent has similar results on both discounted and undiscounted domains, thus it is assumed that discount factors have been handled properly. Actually, agents’ best scores (Table 4) are in discounted domains, with high valued discount factor.

DOMAIN	BIG TOURNAMENT		SMALL TOURNAMENT	
	Score	Rank	Score	Rank
1	0.5250	7	0.6265	2
2	0.6588	4	0.5680	3
3	0.4957	6	0.6225	1
4	0.7619	7	0.8993	1
5	0.9051	2	0.9000	1
6	0.5299	2	0.5489	2
7	0.5996	4	0.6108	2
8	0.7118	7	0.7402	2
9	0.7171	1	0.7759	2
10	0.7057	6	0.8186	1
TOTAL	6.6112	4	7.1109	1

Table 4: Tournament results per domain

Agent	*Score*
BoltzmannAgent	7.110978
ParsAgent	7.032726
RandomDance	6.624709
Atlas3	6.592038

Agent	*Score*
RandomDance	6.847333
PokerFace	6.846875
Sengoku	6.724322
BoltzmannAgent	6.611237
Atlas3	6.487455
AgentH	6.145815
ParsAgent	6.003393
AresParty	6.003482
Mercury	5.831788

Table 5: BoltzmannAgent' ranking for small and big tournament

Considering that no opponent model was used, the results are quite satisfying. The agent negotiates in a self – centered way, taking into account strictly his own outcomes. The only parameters that affect the agent, are discount factors and deadline. It is observed that this strategy reinforced agent's unwillingness to cooperate. Consequently, this often lead the negotiation to the point where no agreement was reached in deadline. This negative feature, we aimed to overcome by adopting an opponent model.

5.2 Boltzmann Strategy with Gahboninho opponent model, modeling opponents as one

Agent's Gahboninho Opponent Model [41] is now employed in the previous version of the agent (Section 5.1). The opponent model, equips agent with information about the opponent, one of them being the ability to estimate opponent's expected utility on a given bid. As a result, the case of an opponent gaining bigger utility than our agent in a specific bid can be computed. This information is used to parameterize the offer(bid) threshold utility of the agent's strategy.
Another hypothesis was put to the test by this implementation; mutual opponent modeling in multilateral negotiation. In the SOAP protocol [32] each agent only negotiates only with two other ones – the same during a session, no matter how many agents participate in total in the negotiation. Thus, in every round the agent sends one offer and receives another one (or an agreement). The time space between the leaving and the incoming offer is the time where the rest of the agents that our agent does not directly communicate, send their counter offers.

Since 3 agents participating in the negotiation, the possible negotiation outcomes for one round are presented in Table 6.

	1	2	3	4
Agent1	make an offer	make an offer	make an offer	make an offer
Agent2	accept	accept	counteroffer	counteroffer
Agent3	accept	counteroffer	accept	counteroffer

Table 6: Possible negotiation outcomes for one round

When Agent1 starts the negotiation no matter what happen in between, the offer that actually gets back to Agent1 can be either his own (that means all the agents agreed on his proposal), either an offer that all the other two agents agreed on, or just the offer of Agent3. In all cases, Agent1's "link" to the negotiation process, is Agent3, because his offer carries "inside" also the previous agent agreement or disagreement. That is why it seemed appealing to evaluate modeling multiple opponents as one, based only on the last received offer.

5.2.1 The Strategy

The strategy from 5.1 was used, but a new parameter is now introduced, **Loss,** which expresses the deviation between the agent's expected utility and the opponent's. Loss controls the temperature variable in Boltzmann distribution, in order to make the agent more competitive or cooperative, depending on the case. More specifically the following things may happen:

1. if **Loss = 0**, agent and the opponent get the same utility
2. if **Loss < 0**, agent gets a bigger utility than the opponent
3. if **Loss > 0**, agent gets a smaller utility than the opponent

Value of Loss combined with deadline and discount factor, increases or decreases the <u>Temperature</u> of Boltzmann Exploration. When agent understands that there is a big deviation between his and opponent's utilities, the alteration of temperature will make him more or less cooperative, in order to decrease this deviation even if he does not win the round eventually.

Since the temperature alteration is also affected by time and the discount factor as mentioned above, the whole process is separated in time- divided stages which represent the state of the game, with the most crucial to be near before the deadline. Since timeline is in normalized form in [0,1], these stages are also equally distributed in [0,1] (e.g. [0.9, 1] is the last stage). Our agent inspects the Loss factor throughout the negotiation. Depending on how big is the loss, the agent will lower his expectations in order for an agreement to be reached (e.g. temperature is gradually decreased) while getting closed to the deadline. On the other hand, if the agent is satisfied (e.g. Loss<0), temperature will remain steady or increase (if the opponent seems conceding) during the negotiation. If an agreement is not reached and time is ending, temperature will decrease in the last stage.

5.2.2 Results and Conclusions

The "Boltzmann agent with the Gahbonihno opponent model and mutual modeling of the opponents" (or, BoltzMutModel Agent) reached 3rd place in the small tournament and 5th place in the big one. In more detail, it won at 4 domains in the small tournament and none in the big one and the overall utility outcome sums up to 7.10 and 6.61 respectively (Table 7 & 8).

DOMAIN	BIG TOURNAMENT		SMALL TOURNAMENT	
	Score	Rank	Score	Rank
1	0.5659	6	0.6937	2
2	0.6621	5	0.6076	3
3	0.5078	6	0.5829	1
4	0.7804	8	0.8953	1
5	0.8953	6	0.9000	1
6	0.4918	5	0.5149	3
7	0.5696	6	0.6245	3
8	0.7672	6	0.7407	2
9	0.6826	3	0.7245	3
10	0.7284	4	0.8185	1
TOTAL	6.6105	5	7.1032	3

Table 7: Tournament results per domain

Agent	*Score*
ParsAgent	7.740972
RandomDance	7.280739
BoltzMutModel	7.1032044
Atlas3	6.601226

Agent	*Score*
ParsAgent	7.148971
PokerFace	7.036215
RandomDance	6.991998
Sengoku	6.838919
BoltzMutModel	6.610508
Atlas3	6.527311
AgentH	6.243412
AresParty	6.183331
Mercury	5.870981

Table 8: BoltzMutModel Agent' ranking for small and big tournament

The agent's utility outcome did not change drastically compared to that of the agent developed in Section 5.1. Although its utility increased in more than half of the domains, the decrease on the rest was so drastically that the overall utility stayed nearly the same as Section 5.1. Since agent's average utility was steady and social welfare increased, it seems that this strategy rather benefited the opponents than the agent himself. We can assume that either mutual opponent modeling did not work properly as expected, or that the way opponent model was incorporated, amplified agent's heard headedness and the last-seconds concede was not enough for an agreement to be reached. More information (e.g. separate opponent modeling at Section 5.3) is needed to decide the cause of agent's results regression.

5.3 Boltzmann Strategy with separate modeling of the opponents

5.3.1 The Strategy

This agent uses the exact same strategy as agent in Section 5.2, but now the opponents are modeled separately. Consequently, Loss factor is represented as the average loss from both opponents. We evaluate the strategy along with the two opponent models mentioned in Chapter 4.

5.3.2 Results and Conclusions

Both agents, Boltzmann agent with Gahbonihno opponent model (or, BoltzGahbOpModel agent) and Boltzmann agent with Bayesian opponent model (or, BoltzBayesOpModel agent) won in the small tournament and got 6th place in the big one (Tables 9). In more detail, Boltzmann Agent with Gahboninho opponent model won in all domains in the small tournament and four in the big one and the overall utility outcome sums up to 6.95 and 6.31 respectively (Table 9 (a), 10 (a) & 11 (a)). Boltzmann Agent with Bayesian opponent model won in 7/10 domains in the small tournament and none in the big one and the overall utility outcome sums up to 7 and 6.42 respectively (Table 9 (b), 10 (b) & 11 (b)).

Comparing the two opponent models with each other when using the same strategy, it seems that utility-wise the Bayesian opponent model is preferable, since it provides the same results in ranking with the Gahboninho one but the overall utility is slightly increased. At the same time, if we disregard utility gained, we can observe that in small tournaments the use of Gahboninho opponent model guarantees a first place in every single domain.

DOMAIN	BIG TOURNAMENT		SMALL TOURNAMENT	
	Score	Rank	Score	Rank
1	0.4343	9	0.5718	1
2	0.6859	1	0.5423	1
3	0.4525	7	0.5902	1
4	0.6898	8	0.8562	1
5	0.9099	1	0.9000	1
6	0.5548	1	0.4937	1
7	0.5692	5	0.6444	1
8	0.6630	8	0.7470	1
9	0.7033	1	0.8500	1
10	0.6610	7	0.7576	1
TOTAL	6.3179	6	6.9536	1

Table 9(a): Tournament results per domain for BoltzGahbOpModel agent

DOMAIN	BIG TOURNAMENT		SMALL TOURNAMENT	
	Score	Rank	Score	Rank
1	0.5015	7	0.6616	1
2	0.6412	5	0.5541	4
3	0.4984	6	0.6208	1
4	0.7424	8	0.8999	1
5	0.9065	2	0.9000	1
6	0.5257	2	0.5649	2
7	0.5101	7	0.4672	4
8	0.7026	7	0.7494	1
9	0.6966	2	0.7746	2
10	0.6969	6	0.8104	1
TOTAL	6.4223	6	7.0030	1

Table 9(b): Tournament results per domain for BoltzBayesOpModel agent

Agent	Score
BoltzGahbOpModel	6.953644
ParsAgent	6.350583
RandomDance	6.093117
Atlas3	6.047309

Table 10(a): Agents' ranking for small tournament – BoltzGahbOpModel agent

Agent	Score
BoltzBayesOpModel	7.003337
ParsAgent	6.895629
RandomDance	6.708873
Atlas3	6.580881

Table 10(b): Agents' ranking for small tournament – BoltzBayesOpModel agent

Agent	*Score*
ParsAgent	6.654447
PokerFace	6.650063
RandomDance	6.627495
Sengoku	6.519486
Atlas3	6.329817
BoltzGahbOpModel	6.317929
AgentH	6.121870
AresParty	5.755018
Mercury	5.635086

Table 11(a): Agents' ranking for big tournament – BoltzGahbOpModel agent

Agent	*Score*
ParsAgent	6.833019
RandomDance	6.808137
PokerFace	6.795616
Sengoku	6.697523
Atlas3	6.501043
BoltzGahbOpModel	6.422346
AgentH	6.121856
AresParty	5.863820
Mercury	5.785524

Table 11(b): Agents' ranking for big tournament – BoltzBayesOpModel agent

Regarding the strategy itself, by observing the results in real-time as the negotiation goes on, it seems that the agent faces a difficulty reaching an agreement, as was the case for its previous versions also (i.e. those of Sections 5.1 and 5.2). In a negotiation when no agreement is reached all agents get zero utility. The agent prefers not reaching an agreement (i.e. to gain zero utility) than lowering his threshold. Still, in the rounds agreement is reached, the utility outcome is very high for the agent. In the small tournament the agent actually is the winner because of that; since only 4 agents participate in the negotiation, our agent is negotiating almost in every round, so many rounds end up with zero utility gain for all the agents in the negotiation. His big utility outcome in other rounds manages to cover this difference, so he conquers 1st place. In the big tournament instead, his occasionally high utility gain is not enough to balance with the amount of zero utility outcomes, since there are a lot more rounds that the agent does not participate in, so the other parties reach agreements without him and increase their utilities.

To sum up, the use of the Boltzmann Strategy by various agents (i.e. those in Sections 5.1, 5.2, 5.3), we observe that the best utility and ranking outcome was achieved from the agent in Section 5.1; the Boltzmann strategy agent, without the use of an opponent model. This is a conflicting result, since the use of an opponent model is expected to boost the negotiation results; instead here the agent's performance deteriorated. This either means that the way the opponent model was incorporated in the strategy was problematic, or that although Boltzmann Strategy does not model the opponent itself, it is kind of an autonomous strategy since it already incorporates learning from past rounds. The strategies examined in the following sections, that are actually in much need of an opponent model in order to be implemented.

5.4 Smart Meta-Strategy

In this section, an effort is made to improve the previous agent, by totally changing the strategy, but keeping the same opponent models in order to notice how they perform along a different strategy.

Specifically, the **Smart meta-strategy** described in [40] has been re-implemented. The basic idea of this strategy, is to propose a counter-offer that has the same utility as the previous bid of the agent but improves the utility of the opponent whenever possible. This strategy is actually suggested to be used combined with Bayesian Opponent Modeling [40]**.** In this thesis, the strategy is tested with both opponent models, as already mentioned (the Bayesian and the Gahbonihno one).

5.4.1 The Strategy

The agent starts with proposing a bid that has maximizes utility given his preferences and accepts a bid when the utility of that bid is higher than the utility of its own last bid or the utility of the bid it would otherwise propose next. Otherwise, the agent will propose a counter-offer.

The basic idea of the smart meta-strategy is to propose a counter-offer that has the same utility (lies on the same utility isocurve) as the previous bid of the agent but improves the utility of the opponent whenever possible. Formally, the strategy searches for a bid b_{t+1}that satisfies Eq. 17.

$$b_{t+1} = argmax(\text{ʉ}(b)), b \in \{x/|u_{own}(x) - \tau| \leq \delta\} \qquad (17)$$

where u_{own}denotes the agent's own utility function and
τ: a target utility
$\{x/|u_{own}(x) - \tau| \leq \delta\}$: the utility iso-curve that has the same utility for the agent (within a small interval [τ-δ, τ+δ]), but might have different utilities for the opponent.

The strategy selects a bid from the iso-curve that maximizes the expected utility of the opponent. If it is not possible to find a bid that thus improves the utility of the opponent, a concession step will be performed after performing smart steps (i.e. steps that stay on the same iso-curve and try to improve the next bid for the opponent by using the updated opponent model).

As mentioned, offers are accepted or not, based in the utility target that is set by the bidding strategy. Sometimes a concession step must be made, by decreasing the target utility $\tau$of their next bid. τ is decreased by a fixed concession step c that depends on time elapsed and the discount factor (Eq. 18).

$$\tau = \tau * c \qquad (18)$$

5.4.2 Results and Conclusions

Both Smart Meta-strategy agents (or, SmartMetaGahb and SmartMetaBayes) won in the small tournament, the agent with Gahboninho opponent model got the 4th and the Bayesian one the 5th place in the big tournament (Tables 12). In more detail, both opponent model Smart-meta Agents won in 9/10 domains in the small tournament and in 2/10 in the big one. The agent with the Gahboninho opponent model gained overall utility 7.25 and 6.57(Table 12 (a), 13 (a) & 14 (a)). Kind of the same implies for Smart-meta Agent with Bayesian opponent model, his utility outcome sums up to 6,93 and 6.54 respectively (Table 12 (b), 13 (b) & 14 (b)).

For this strategy, is quite clear that Gahboninho would be the preferred opponent model. Not only Gahboninho opponent model lead to better ranking position, but also contributed in the utility outcome and social welfare. So, although this strategy was originally proposed to be applied along a Bayesian opponent model, in our setting better results came of a frequency based one.

DOMAIN	BIG TOURNAMENT		SMALL TOURNAMENT	
	Score	Rank	Score	Rank
1	0.4842	7	0.6239	1
2	0.6842	1	0.5520	1
3	0.5134	7	0.6326	1
4	0.7480	7	0.9259	1
5	0.8996	4	0.9000	1
6	0.5184	2	0.4582	2
7	0.5624	6	0.6530	1
8	0.7255	7	0.8473	1
9	0.7291	1	0.8376	1
10	0.7095	4	0.8195	1
TOTAL	6.5728	4	7.2504	1

Table 12(a): Tournament results per domain for SmartMetaGahb agent

DOMAIN	BIG TOURNAMENT		SMALL TOURNAMENT	
	Score	Rank	Score	Rank
1	0.4783	7	0.6022	1
2	0.6854	1	0.5470	1
3	0.4877	7	0.6513	1
4	0.7417	7	0.6513	1
5	0.8993	4	0.8999	1
6	0.5193	2	0.4687	2
7	0.5738	6	0.6275	1
8	0.7243	7	0.8467	1
9	0.7265	1	0.8467	1
10	0.7095	4	0.8173	1
TOTAL	6.5463	5	6.9387	1

Table 12(b): Tournament results per domain for SmartMetaBayes agent

Agent	*Score*
SmartMetaGahb	7.250402
ParsAgent	6.864512
RandomDance	6.456938
Atlas3	6.381303

Table 13(a): Agents' ranking for small tournament – SmartMetaGahb agent

Agent	*Score*
SmartMetaBayes	6.938778
RandomDance	6.281899
Atlas3	6.210976
ParsAgent	6.201596

Table 13(b): Agents' ranking for small tournament – SmartMetaBayes agent

Agent	*Score*
ParsAgent	6.792917
PokerFace	6.776029
Sengoku	6.654860
SmartMetaGah	6.572860
Atlas3	6.417707
AgentH	6.245342
AresParty	5.850221
RandomDance	5.774083
Mercury	5.704133

Table 14(a): Agents' ranking for big tournament – SmartMetaGahb agent

Agent	*Score*
ParsAgent	6.771313
PokerFace	6.753369
RandomDance	6.746760
Sengoku	6.634664
SmartMetaBayes	6.546394
Atlas3	6.426501
AgentH	6.224636
AresParty	5.851595
Mercury	5.696180

Table 14(b): Agents' ranking for big tournament – SmartMetaBayes agent

Regarding the strategy itself, by observing the results in real-time as the negotiation goes on, it seems that the agent is more cooperative than the previous strategies. More agreements are reached during the negotiation and the agent seems to perform a controlled concede. Smart Meta-Agent outperforms the previous agents (i.e. those in Sections 5.1, 5.2, 5.3) considering ranking position. The same does not imply regarding overall utility gain. The fact that the agent became more cooperative, made him lower his threshold, so agreements are made but with less outcome.

5.5 Hybrid MGT - Smart Meta-Strategy

In this section a new hybrid strategy is proposed in order to reinforce the previous agent (Section 5.4); aspects of the Maximum Greedy Tradeoffs Algorithm [43] are incorporated into the Smart Meta-strategy one.

Most of the introduced strategies in the ANAC competition, as well as the evaluated ones in this thesis so far, aim to maximize the overall utility of a bid during the negotiation process. Here a new hypothesis is tested; to maximize the agent's utility per issue, depending on the agent's weight in it (but also the opponents'), instead of targeting the best overall utility. Specifically, since agents have difference preference profiles, they may have strong or weak opposition towards an issue. Thus, they may not be concerned about all the issues in the negotiating bid, but the specific one that is going to drastically maximize their utility.

5.5.1 The Strategy

In general, the MGT Algorithm generates Pareto-optimal offers with perfect information about the opponent. Since in our case, there is incomplete information, we take advantage of the Opponent Modeling to estimate what is needed. The information we get from the above combination, is used to conduct a learning method to reveal the greedy order (agenda) and generate near-Pareto-optimal offers. The algorithm is further explained below.

We have already defined the utility function of the agent (Eq. 19).

$$u(b_t) = \sum_{i=1}^{n} w_i e_i(x_i \epsilon b_t) \qquad (19)$$

The maximization of the utility function, is actually the aspiration of an agent, since the wants to gain as much utility as possible by the end of the negotiation.

We define the *greedy choice*, as the offer that the agent proposes during the negotiation on an issue i, that has maximum worth to the opponent, but meanwhile, it does not affect significantly the utility outcome of our agent. Practically this means that the agent accepts or proposes an offer on an issue that is not high value weighted for the agent, since this action will not change drastically its utility described in Eq.19.

For any issue i we define the greedy ratio:

$$r_i = \frac{w_i'}{w_i} \qquad (20)$$

where

w_i: agent's weight in issue i

$w_i{}'$: the opponent's weight in the same issue

A greedy choice k is an issue which minimizes the greedy ratio for all issues that Eq. 21 applies. This means that r_k is the lowest greedy ratio for the agent between all issues under negotiation. Thus, this is the issue where our agent cannot gain much utility.

$$r_k \leq r_i \qquad (21)$$

So, we define greedy order, as the taxonomy of issues, regarding their greedy ratio. It is obvious that the opponent's greedy order is the different from our agent's, since each negotiation participant has different weight over an issue.

The above observations have been incorporated in the Smart- meta strategy in the following way:

For each issue i , two values are computed, $r_i{}'$ the and $r_i{}'$, where:

r_i': the greedy ratio of our agent in issue i

$r_i{}'$: the greedy ratio of the opponent issue i

So, one of the following things may occur:

1. $r_i \approx 1$, if the agent has the same or similar weight with the opponent in issue i
2. $r_i < 1$, if the opponent has bigger weight in issue i than the agent
3. $r_i > 1$, if the agent has bigger weight in issue i than the opponent

Note that the values r_i are computed based on the Opponent Model, and may change in every round, since the hypothesis about the opponent preference profile is building round by round. So, based on r_i the agent may face the following situations:

1. $r_i \approx 1$ **and** $r_i{}' \approx 1$
2. $r_i < 1$ **and** $r_i{}' \approx 1$
3. $r_i > 1$ **and** $r_i{}' \approx 1$
4. $r_i < 1$ **and** $r_i' < 1$
5. $r_i > 1$ **and** $r_i' > 1$
6. $r_i > 1$ **and** $r_i' < 1$

The agent's goal is to take advantage of some of the above situations, so that he can reach a better agreement, since they actually represent the low, medium or strong opposition agents may have on an issue.

Of course, it must be taken into consideration that the differences may be small, so the following observations are not only going to be modeled strictly.

<u>1st Observation:</u> In case 1, is possible that all participants have low, medium or high opposition. In case they have low, meaning this is an issue with low weight for all of them, there is no need for our agent to be competitive in this specific issue, of course without falling under his fixed reservation value for this issue. This will confuse the opponents, as it is going to seem that our agent compromises, when actually he is not, since he has not a lot to gain from this specific issue.

<u>2nd Observation:</u> In cases 2 and 3, our agent has actually one conflicting opponent, the one with the strong preference to this issue. So, our agent tries to take advantage of the second agent's low preference for this profile, to take him by his side, and put mutual pressure to the third agent who either has conflicting profiles with our agent, or he is the only one with the possibility of a big outcome for this issue given the weight distribution.

<u>3nd Observation:</u> Case 5, is actually the most important for our agent, since it represents a negotiating issue with maximum profit for him and only him. It is crucial for the agent to achieve big outcome in this issue, so he remains quite stubborn in his offer in this part of the bid, with the only thing affecting him being deadline and discount factor.

Cases 4 and 6 do not show any specific interest, so they are not taken into account, and the agent generates offers on those issues based on version2 bidding and accepting strategy.

5.5.2 Results and Conclusions

The Hybrid MGT-Smart Meta-strategy agents (or, HybridMgtGahb and HybridMgtBayes) won in the small tournament, and got the 5th place in the big tournament (Tables 15). In more detail, both hybrid MGT-Smart-meta Agents won in 10/10 domains in the small tournament and in 2/10 in the big one. The agent with the Gahboninho opponent model gained overall utility 7.26 and 6.49 (Table 15 (a), 16 (a) & 17 (a)). Hybrid MGT-Smart-meta Agent with Bayesian opponent model, summed up utility outcome 7.20 and 6.5 respectively (Table 15 (b), 15 (b) & 15 (b)).

The two implementations of this agent have almost identical results, so we cannot say this strategy performed better with a particular one.

DOMAIN	BIG TOURNAMENT		SMALL TOURNAMENT	
	Score	Rank	Score	Rank
1	0.4761	7	0.6530	1
2	0.6816	1	0.5470	1
3	0.4543	7	0.6123	1
4	0.7655	7	0.9430	1
5	0.9084	2	0.9007	1
6	0.5620	1	0.5165	1
7	0.5699	5	0.6935	1
8	0.6869	7	0.7460	1
9	0.7063	2	0.8271	1
10	0.6844	6	0.8218	1
TOTAL	6.4958	5	7.2613	1

Table 15(a): Tournament results per domain for HybridMgtGahb agent

DOMAIN	BIG TOURNAMENT		SMALL TOURNAMENT	
	Score	Rank	Score	Rank
1	0.4770	7	0.6475	1
2	0.6851	1	0.5220	1
3	0.4440	7	0.6495	1
4	0.7664	7	0.9331	1
5	0.9082	2	0.9000	1
6	0.5629	1	0.4913	1
7	0.5777	4	0.6367	1
8	0.6848	7	0.7488	1
9	0.7089	2	0.8179	1
10	0.6842	6	0.8320	1
TOTAL	6.5000	5	7.2092	1

Table 15(b): Tournament results per domain for HybridMgtBayes agent

Agent	*Score*
HybridMgtGahb	7.261328
ParsAgent	6.689380
RandomDance	6.316138
Atlas3	6.185008

Table 16(a): Agents' ranking for small tournament – HybridMgtGahb agent

Agent	*Score*
HybridMgtBayes	7.209276
ParsAgent	6.658453
RandomDance	6.384802
Atlas3	6.184660

Table 16(b): Agents' ranking for small tournament – HybridMgtBayes agent

Agent	Score
ParsAgent	6.757372
RandomDance	6.692533
PokerFace	6.651355
Sengoku	6.580025
HybridMgtGahb	6.495823
Atlas3	6.386691
AgentH	6.156548
AresParty	5.853842
Mercury	5.653455

Table 17(a): Agents' ranking for big tournament – HybridMgtGahb agent

Agent	Score
ParsAgent	6.741607
RandomDance	6.680253
PokerFace	6.666688
Sengoku	6.590154
HybridMgtBayes	6.500759
Atlas3	6.369100
AgentH	6.155796
AresParty	5.853859
Mercury	5.666448

Table 17(b): Agents' ranking for big tournament – HybridMgtBayes agent

Regarding the strategy itself, by observing the results in real-time as the negotiation goes on, it seems that the agent's cooperativity as in Section 5.4 has preserved. Only a few of the negotiations do not reach an agreement before deadline.

The agents have outstanding results in the small tournament, since they gained the higher utility outcome that has been reached so far in the evaluation process. Although the opponent model difference, the results are almost the same, meanwhile in Section 5.4, Smart-Meta strategy by itself had different outcome for each opponent model, with the Gahboninho outperforming the Bayesian one. Despite the high utility outcome and ranking position in the small tournament, when agent negotiates with a larger number of agents, the results are not equally good; they are actually worse than those of the agents in Section 5.4, especially regarding the agent with the Gahboninho opponent model whose utility not only decreased, but the agent also dropped one place in the rankings, as compared to its performance in Section 5.4.

5.6 Conan Strategy

The Conan Agent is one of the most recently introduced agents that can be found in the literature [44]. In this section we developed an agent who aims to implement the Conan strategy, along with the alterations required to fit in our setting. The strategy itself, is based on heuristics that model explicitly the environment of the negotiation and the self (individual) factors of the agent.

As a consequence, this is a strategy without need of an opponent model; all the needed information can be retrieved from the environment and the agent, as also partial information about the opponents' behavior can be gained.

5.6.1 The Strategy

In this strategy, offers are generating based on Eq. 22:

$$Offer_{t,s_t} = IP + (RP - IP) * CR_{t,s_t} \qquad (22)$$

where,

$Offer_{t,s}$: the offer for agent i at time t

IP: the initial price of an offer

RP: reservation price

$CR_{t,s}$: the concession rate for agent i at time $t \in [0,1]$

The Concession's rate value is not static but changes, depending on the time left for the negotiation to end (i.e. timeline of the negotiation) (Eqs .23, 24 and 25).

$$\text{if } t = T_{start}: \quad CR_{t,s} = 0 \tag{23}$$

$$\text{if } t = T_{end} - 1: \quad CR_{t,s} = 0.99 \tag{24}$$

$$otherwise: \quad CR_{t,s} = w_{self}S_t + w_{env}E_t \tag{25}$$

where,

T_{start}: start time of negotiation,

T_{end}: the deadline of the negotiation

E_t: effect of environmental factors

S_t: effect of self factors

The weights w_{self}, w_{env} represent the importance of the self and environmental factors respectively. In our implementation, we assume that in the above equation the environment's weight is 1 and the agent's weight is the average weight regarding each issue of the domain.

Regarding environmental factors (e.g. number of agents participating in the negotiation), the information needed in the originally proposed formula is already known in our setting. So, we defined a fixed value to describe them. Thus, we have set environmental factors: $E_t = 0.11$.

A formula to describe self-factors as well, is proposed in the Conan Strategy (Eq. 26):

$$S_t = 0.25 * (\frac{1}{CO} + NS + T_{end} + E_g) \tag{26}$$

where,

CO: Number of committed offers

NS: Negotiating status

E_g: Eagerness of the agent

Eagerness, represents the willingness of the agent to concede in the negotiation in order for an agreement to be reached. In our setting, this factor must be taking into account that discounted domains may exist in the negotiation tournament. In case there is no discount factor in the domain, Eagerness is fixed and equals to 0.5. Else, $E_g = 1 - DF$, where DF is discount factor.

Another factor is also introduced, the negotiating status which represents the progress of the negotiation. It is calculated based on the factor C, which is opponent's concession rate for the last $\emptyset$ offers. In our setting, $\emptyset = 1$, so negotiation status is based only in the previous offer. This value has been chosen, because in discounted domains the behavior of an agent may change drastically from one round to another. So, we chose to define the negotiation status only from the last offer, to achieve as much accuracy as possible.

To calculate the concession rate C of an agent, the Borda Method has been used [53]. We define that concession rate declares if an agent is compatible, Moderate Compatible or Incompatible. With Borda Method mapping, a status for each agent is created. This is actually like a "quick" opponent model implementation.

Let $\delta_{i,t}$ be the number of points agent i gets from Borda Method in round at time t . If the opponent concedes in the last round, he gets 1 point from Borda Method, else he gets 3 points. Then for each agent Eq.27 applies.

$$\delta_i = \delta_i + \delta_{i,t} \qquad (27)$$

When δ_i increases, it means that the opponent is not willing to cooperate and the opposite when it decreases.

So, negotiation status is described in Eq. 28:

$$NS = \frac{\delta_i - 2k}{3k} \qquad (28)$$

where k is the number of rounds.

Note that NS decreases as the opponent becomes more cooperative. Since the agent negotiates with two opponents at the same time in our setting, NS is computed as the average of the two opponents' negotiating status (Eq. 29).

$$NS = \frac{NS_1 + NS_2}{2} \qquad (29)$$

5.6.2 Results and Conclusions

Conan agent (or, ConanAgent) reached 4th place in the small tournament and 8th in the big one. Regarding utility outcome, he gained 6.33 and 6.20 respectively (Tables 18, 19). Agent's results, are actually the worst observer so far in both ranking and utility. He did not even manage to win in any domain in either tournaments.

DOMAIN	BIG TOURNAMENT		SMALL TOURNAMENT	
	Score	**Rank**	**Score**	**Rank**
1	0.5547	9	0.6026	4
2	0.6101	6	0.5641	3
3	0.5350	9	0.5881	4
4	0.8030	8	0.8205	4
5	0.8092	8	0.8808	4
6	0.4519	7	0.4267	3
7	0.4525	9	0.4283	4
8	0.7084	9	0.7408	4
9	0.6031	7	0.5856	4
10	0.6732	9	0.6940	4
TOTAL	6.2015	8	6.3319	4

Table 18: ConanAgent's Tournament results per domain

Agent	*Score*
ParsAgent	8.795433
RandomDance	8.321433
Atlas3	7.445446
ConanAgent	6.331920

Agent	*Score*
ParsAgent	7.604397
PokerFace	7.453699
Atlas3	7.412369
RandomDance	7.404556
Sengoku	7.188032
AresParty	6.730279
AgentH	6.496510
ConanAgent	6.201551
Mercury	6.102859

Table 19: ConanAgent's ranking for small and big tournament

Definitely, the fact that an opponent model has not been used, affected the agent's performance. This strategy is originally proposed without an opponent model, but this did not work out well in our setting. Probably this has to do with the fact that this strategy is self-centered and takes into account mostly environmental factors, that in our setting are not important enough to determine the outcome, as knowledge about the opponents is. Still, we have already evaluated a strategy without opponent model in Section 5.1, with much more promising results than this one. Thus, the fact that an opponent model has not been incorporated in the strategy is crucial, but it is not totally responsible for the agent's bad performance.

5.7 Conan Strategy with use of opponents' weights

Citing the results in Section 5.6, we decided to extent the Conan strategy by incorporating opponent modeling into it. This new version of the strategy considers that environmental factors are opponent-depended. This new implementation was tested with the Gahboninho and Bayesian opponent model as well.

5.7.1 The Strategy

In the originally proposed strategy, concession rate (Eq. 25) is considered to be a linear function, depending from agent's preference (i.e. the weights) over an issue and the environmental factors.

$$CR_{t,s} = w_{self} S_t + w_{env} E_t \qquad (25)$$

where S_t (i.e. effect of self-factors), E_t (i.e. effect of environmental factors) and weights are as described in Section 5.6.

An improvement of Conan strategy is proposed here; We decided to alter the original concession rate function, in order to incorporate agent's beliefs about opponents. The environmental factor weight is replaced with the estimated from the opponent model weight of the opponent, in the particular negotiating issue at the time. Since our agent negotiates with two other agents, this new concession rate had to be estimated regarding both of them, so the average opponents' weight per issue was estimated and incorporated in the concession rate function.

Thus, beliefs about the opponents are taken into consideration and Conan strategy maintains its aspects but stops being only self and environment oriented.

5.7.2 Results and Conclusions

The Conan Strategy agents (or, ConanGah and ConanBayes) got the 3d place in the small tournament and the 4th place in the big tournament (Tables 20). In more detail, both new Conan Agents did not manage to win in any domain in both tournaments; the agent with the Gahboninho opponent model gained overall utility 7.32 and 6.98 (Table 20 (a), 21 (a) & 22 (a)) and the Bayesian one, 7,35 and 6.97 respectively (Table 20 (b), 21 (b) & 22 (b)).

The two implementations of this agent have almost identical results, so we cannot say that this strategy performed better with a particular one.

DOMAIN	BIG TOURNAMENT		SMALL TOURNAMENT	
	Score	Rank	Score	Rank
1	0.6027	6	0.7021	4
2	0.7004	3	0.6006	3
3	0.5701	6	0.6647	4
4	0.8582	3	0.8887	3
5	0.9085	2	0.9000	4
6	0.5427	3	0.5192	4
7	0.6198	5	0.6589	3
8	0.7554	7	0.8147	4
9	0.6787	3	0.7723	3
10	0.7440	5	0.7998	4
TOTAL	6.9810	4	7.3214	3

Table 20(a): Tournament results per domain for ConanGah agent

DOMAIN	BIG TOURNAMENT		SMALL TOURNAMENT	
	Score	Rank	Score	Rank
1	0.5964	6	0.7057	4
2	0.6997	3	0.6053	3
3	0.5679	6	0.6689	4
4	0.8567	3	0.8536	3
5	0.9084	2	0.8999	3
6	0.5463	2	0.5179	4
7	0.6203	5	0.6522	3
8	0.7543	7	0.8765	3
9	0.6806	3	0.7802	3
10	0.7447	4	0.7930	4
TOTAL	6.9758	4	7.3536	3

Table 20(b): Tournament results per domain for ConanBayes agent

Agent	*Score*
ParsAgent	8.436776
RandomDance	7.834595
ConanGahb	7.321495
Atlas3	7.331629

Table 21(a): Agents' ranking for small tournament – ConanGah agent

Agent	*Score*
ParsAgent	8.237052
RandomDance	7.801234
ConanBayes	7.353666
Atlas3	7.210370

Table 21(b): Agents' ranking for small tournament – ConanBayes agent

Agent	*Score*
ParsAgent	7.341698
PokerFace	7.247328
RandomDance	7.234683
ConanGahb	6.981038
Sengoku	6.886076
Atlas3	6.711079
AgentH	6.501046
AresParty	6.490674
Mercury	6.021149

Table 22(a): Agents' ranking for big tournament – ConanGah agent

Agent	*Score*
ParsAgent	7.319800
PokerFace	7.248236
RandomDance	7.209815
ConanBayes	6.975834
Sengoku	6.897981
Atlas3	6.685407
AgentH	6.505297
AresParty	6.470708
Mercury	6.026784

Table 22(b): Agents' ranking for big tournament – ConanBayes agent

The agents, however, have outstanding results in both tournament regarding the utility outcome and social welfare. Actually, this agent overcomes the agent of Section 5.5 who had the best utility outcome so far. Still, the agents did not manage to climb in rankings, since apparently (judging from the results) this strategy have also favored a lot the opponents. Intuitively, the addition of opponent modeling helped everyone reach arguments faster. Thus, the opponents did not suffer a decrease in their utility. Regardless, our assumption that an opponent model incorporated in Conan strategy would lead to better results, was right. The new altered strategy shows much improvement compared to the one in Section 5.6, both in terms of ranking and utility-wise, although it did not manage to overcome previously implemented agents in this thesis.

5.8 Conan Strategy incorporated in Hybrid MGT - Smart Meta-Strategy

For our last implementation, we decided to take the Conan Agent one step further, since the evaluation process in Section 5.7 led in very promising results regarding outcome. The concession rate in Section 5.7 was altered, in order to take opponents preference into account, but handled poorly, by estimating the average of the two opponents' weigh in every issue. A new concession rate and negotiation status management is proposed in this implementation, by incorporating the Hybrid MGT-Smart Meta-Strategy proposed in Section 5.5.

5.8.1 The Strategy

From the greedy ratio estimation in Section 5.5, it is easy to discover the opponents with whom the agent has similar weights in issues (i.e. same preferences). Thus, the opponent with the biggest impact in our agent's performance can be found; it will be the one who is strongly opposed to our agent in issues with high importance for both of them.

The previous strategy (i.e. the one in Section 5.7) is modeled for both opponents, but is weighted, with bigger weight to the negotiating status of the agent whose preferences are similar to the agent as the exploration goes on. The weight of NS, is actually the probability of similarity. So, we propose the following formula (Eq. 30), instead of the one used in Section 5.6 (Eq. 29):

$$NS = \frac{P(s)NS_1 + (1-P(s))NS_2}{2} \qquad (30)$$

where P(s): the probability of similarity of the agent with the most similar preference profile.

Moreover, the same concept applies on concession rate; the environmental weight is not replaced with the average weight of the opponents on an issue as in Section 5.7. Considering in which issue the agents negotiated on and the MGT algorithm results about greedy ratio, the agent's estimated concession rate (described in Eq. 25) depends on the preference (i.e. weight) of the stronger opposing agent on the particular issue in the domain it negotiates in at the time.

5.8.2 Results and Conclusions

Both "Conan with Hybrid MGT-Smart Meta-Strategy" agents (or, HybridConanGahb and HybridConanBayes) won in the small tournament, the Gahboninho agent got the 4^{th} place and the Bayesian one the 3d in the big tournament (Tables 20). In more detail, Gahboninho Hybrid Conan agent won in 8/10 domains in the small tournament while achieving utilities of 7.27 utilities and none in the big one, where it achieved a utility of 6.41 (Table 23 (a), 24 (a) & 25 (a)). Respectively, the Bayesian Hybrid Conan agent won in 10/10 domains n the small tournament with 6.95 utility and 4/10 in the big one with 6.63 utility (Table 23 (b), 24 (b) & 25 (b)).

The two implementations of this agent achieved similar results regarding utility, but the Bayesian agent managed to achieve a better ranking: actually, the best ranking observed so far among all implemented agents in this thesis.

DOMAIN	BIG TOURNAMENT		SMALL TOURNAMENT	
	Score	Rank	Score	Rank
1	0.4675	7	0.6490	1
2	0.6495	3	0.6095	2
3	0.4842	6	0.6254	1
4	0.7481	7	0.9008	1
5	0.9046	2	0.9000	1
6	0.5060	2	0.5385	2
7	0.5651	3	0.6201	1
8	0.7045	8	0.7901	2
9	0.6898	2	0.8101	1
10	0.6911	5	0.8263	1
TOTAL	6.4108	4	7.2703	1

Table 23(a): Tournament results per domain for HybridConanGahb agent

DOMAIN	BIG TOURNAMENT		SMALL TOURNAMENT	
	Score	Rank	Score	Rank
1	0.5489	5	0.5757	1
2	0.6882	1	0.5571	1
3	0.5368	5	0.5920	1
4	0.7142	8	0.8462	1
5	0.9122	1	0.9000	1
6	0.5576	1	0.5326	1
7	0.5605	6	0.5703	1
8	0.7105	7	0.7504	1
9	0.7034	1	0.8759	1
10	0.7025	3	0.7543	1
TOTAL	6.6352	3	6.9548	1

Table 23(b): Tournament results per domain for HybridConanBayes agent

Agent	*Score*
HybridConanGahb	7.270371
ParsAgent	7.151332
RandomDance	6.738120
Atlas3	6.535179

Agent	*Score*
HybridConanBayes	6.954831
ParsAgent	6.279284
RandomDance	6.092185
Atlas3	6.052871

Table 24(a): Agents' ranking for small tournament – HybridConanGahb agent

Table 24(b): Agents' ranking for small tournament – HybridConanBayes agent

Agent	***Score***
RandomDance	6.635031
ParsAgent	6.620279
PokerFace	6.608827
HybridConanGahb	6.410856
Sengoku	6.410779
Atlas3	6.285261
AgentH	6.108981
AresParty	5.676200
Mercury	5.457333

Table 25(a): Agents' ranking for big tournament – HybridConanGahb agent

Agent	***Score***
RandomDance	6.645572
PokerFace	6.636945
HybridConanBayes	6.635278
ParsAgent	6.635185
Sengoku	6.526289
Atlas3	6.342931
AgentH	6.112963
AresParty	5.777456
Mercury	5.641552

Table 25(b): Agents' ranking for big tournament – HybridConanBayes agent

Comparing this strategy to that of Section 5.7 we definitely notice an improvement, especially regarding the ranking in the small tournament, since the agent managed to jump from the 3d to 1st place in both opponent model implementations. The main contribution of this version of Conan strategy, is that for the first time in this thesis an agent manages to reach such a high ranking in both tournaments, as agents' Bayesian version here did. It seems that the alterations made compared to the implementations of Sections 5.6 and 5.7 benefited the agent, and the concession and negotiating status factors were handled properly, definitely much better than previous versions. The only weak result of this strategy was in terms of utility gain; yet that did not affect ranking, it even elevated it, since the social welfare decreased drastically.

6. Further Experimentation

In order to further evaluate our agents, we organized a new tournament, where the implemented agents were pitted against each other, without any opponents from previous ANAC competitions. The domains were exactly the same as the ones used before.

The results of the tournament are presented in Table 26.

Agent	Score	StD
ConanGahb	4.273	0.523
ConanBayes	4.261	0.511
ConanAgent	4.147	0.397
BoltzMutModel	3.886	0.136
BoltzmannAgent	3.833	0.083
BoltzBayesOpModel	3.822	0.072
SmartMetaGahb	3.640	0.11
SmartMetaBayes	3.633	0.117
HybridMgtBayes	3.558	0.192
HybridMgtGahb	3.556	0.194
HybridConanGahb	3.405	0.345
BoltzGahbOpModel	3.378	0.372
HybridConanBayes	3.370	0.38

Table 26: Results of the Tournament with participant agents being our developed agents. Standard Deviation (StD) is also included.

By observing the results, we notice that the agents' relative performance was not entirely consistent to that in the tournaments held in Chapter 5. This deviation was expected, since an agent's performance may drastically change given its opponents. This is an entirely different environment, since the whole interaction process changes because of the different strategies from agent to agent.

Actually, the best performing agent, (Section 5.8 – Conan with Hybrid MGT-Smart-Meta strategy), appears to be the weakest one here. The top two agents are the ones from Section 5.7 – Conan Strategy with use of opponents' weights, with the first being the one using Gahboninho opponent modeling. Those agents had seemed to be promising because of their high utility gain – the higher in the whole evaluation process in Chapter 5. In the third place comes the Conan strategy of Section 5.6: this is a self-centered agent, with no opponent model. We notice that actually the agents with the worst performance in the tournaments referred in Chapter 5, have managed to outperform the ones here, by a significant utility difference.

Boltzmann Strategy agents (i.e. those in Sections 5.1 and 5.2) seem to perform well here also, as they reached 4^{th} and 5^{th} place respectively. This is quite interesting, since these were relatively simple strategies, especially the one in the Section 5.1 which does not even employ an opponent model.

7. Our ANAC submission

As already mentioned, one of the purposes of this thesis was to select one of the developed agents to participate in the 2017 international ANAC competition.

In this point, some changes in the rules of the ANAC tournament in 2017 must be noted, since they led us to face certain difficulties regarding the agents' implementations. ANAC '16 and ANAC '17 were the only competitions where learning from previous negotiations and storing information were allowed. When this thesis started, only ANAC '15 agents and the negotiating platform were available at the time. That is why in our tournaments of Chapter 5, only ANAC '15 agents were used as opponents, as mentioned in Section 5.1. So, our agents had to be originally developed following the ANAC '15 rules and coding format to be able to interfere and negotiate in the Genius platform against those opponents. Last years' agents along with the Genius version that were about to be used in this years' competition, were available only around one month before the submission.

By this time, our agents had already been developed and evaluated, following the process outlined in Chapter 5. Considering those results, we selected to submit to the agent from Section 5.8 – Conan agent with Hybrid MGT Smart-Meta-strategy along with Bayesian opponent model, since it appeared to be the most efficient among the others. Note that the internal tournament described in Chapter 6, had not been held at the time, or else it would have been taken into account.

Since our agent for submission had been chosen, certain changes had to be made, in order to satisfy the new competition rules and match the newly introduced setting of ANAC '17. We incorporated the learning from previous negotiations aspect in a simple way: the agent was saving the offers that led to an agreement for each opponent in every domain, in order to use them in the following negotiations; this is expected to benefit the agent, since an agreement is guaranteed and less time is reclaimed for a negotiation. Regarding saving information, other sharper ideas were explored (e.g. storing all the preference profiles of the opponents, or even recognizing information about an opponent's identity and strategy) but unfortunately the way the strategy itself had been developed, did not allow further actions to be implemented in this specific timeline.

Regarding the actual competition process, the 2017 qualification tournaments were repeated 5 times in 8 different negotiating scenarios. The agents were split at two different pools, each of one consisted of 9 autonomous negotiating agents; thus 18 agents in total participated in the qualification tournament and the top 4 from each pool participated in the finals. Our agent participated in one of those pools facing the agents showing in Table 29.

We should note, that apart from the agent itself, each participant in the tournament should also submit a negotiating scenario. So, in total 18 scenarios were submitted, but only 8 of them were selected for the actual tournament. Our negotiating scenario submitted, Movie Time, was actually one of the chosen ones (Table 27). Movie Time Domain consisted of three different preference profiles. It is attributes are separately presented in Table 28.

Name	# of Issue	# of Values (1)	# of Values (2)	# of Values (3)	# of Values (4)	# of Values (5)	Scenario Size
GeneJack	4	10	10	10	10		10000
MyDomain	5	4	10	8	14	9	40320
Music	4	9	4	7	6		1512
SuperMarket	5	8	6	5	6	4	5760
Movietime	4	6	2	3	4		144
Lunch Time	3	5	4	4			80
Taxung	4	4	5	3	6		360
SmartGrid	3	5	4	4			80

Table 27: The negotiating scenarios that have been selected to be used in the tournament. Among those, our submission Movie Time domain.

Movie Time Domain

Film Genre	Place	Transportation	Snack
Thriller	*Home*	*Car*	*Pop Corn*
Comedy	*Cinema*	*Bus*	*Nachos*
Adventure		*Taxi*	*Ice cream*
Documentary			*Milkshake*
Film Noir			
Animation			

Table 28: Movie Time Domain, different issues and issue values are presented

The tournament results of the pool our agent participated, are shown in Table 29, as presented in the *IJCAI*[4] '17 conference.

[4] https://ijcai-17.org/

Agent Name	***Individual Utility***
Rubick	0.742502203
AgentF	0.722600455
ParsAgent3	0.720973580
CaduceusDC16	0.709912278
Mosa	0.700470270
Gin	0.696220696
GeneKing	0.683409649
TucAgent	0.675038202
Group3	0.623009688

Table 29: Qualification tournament results. Our agent is TucAgent. No information was provided regarding the agents' performance in individual domains.

Unfortunately, our agent did not manage to reach the top four places and participate in the finals. One of the aspects that lead to our agent's failure is the fact that the new added rules of ANAC as they have been explained in the begging of this chapter had not been taken into account from the beginning of the agents' development, but only handled nearly before the submission deadline. Moreover, the further experimentation results (i.e. those of Chapter 6), probably would have changed our submission choice, if this stages tournament had been held earlier. In any case, the choice and number of opponents may be crucial to an agent's performance, as was demonstrated in our experiments in Chapter 5 (e.g. Atlas3 agent was the winner of ANAC '15, but in our tournaments, was hardly in the top three performing agents).

8. Conclusions

In this thesis we explored alternative negotiation strategies, along with different opponent models, for use by agents capable of participating in the international ANAC competition. We chose to implement, evaluate, and test several strategies found in the literature, along with entirely novel concepts and strategies. These were all systematically and extensively evaluated, as detailed in our thesis. Our results show that, in the ANAC setting, frequency based opponent models are slightly more beneficial to most agent strategies than Bayesian ones. Still, combining a Bayesian opponent model with a specific agent strategy ("Conan" in Section 5.8) resulted to the strongest agent among the twelve we developed – i.e., the agent that outperformed all the rest when pitted against them in an ANAC competition simulation.

We observed in our experiments that when accurate opponent modeling is achieved in an early stage, agreements are reached far before the deadline. As such, when this occurs, a higher utility outcome is guaranteed to all agents in discounted domains.

Now, results in Chapter 5 showed that the "joint" modeling of the opponents (modeling all opponents as one) did not appear to perform as well as modeling opponents separately. However, there was no significant difference in performance among the two. It is interesting that in the tournament described in Chapter 6, the Boltzmann agent with the Gahbonihno "joint" opponent modeling did manage to reach the 4th place, and achieved better results than the one employing separate modeling of the opponents.

Regarding the implemented strategies themselves, the results are rather conflicting, since a slight change in the negotiation setting (such as replacing an agent with another, or changing the number of the negotiating parties) can alter the negotiation outcome. Still, some general observations can be made.

First of all, employing a complicated or "sophisticated" strategy does not actually guarantee good results: we observed that relatively simple strategies, such as the Boltzmann one – used without even employing an opponent model - performed better than more elaborate ones. Moreover, we observed that the idea of focusing on conducting separate negotiations per issue, was found to be helpful in many cases. By examining all the different aspects, we could say that if the best strategy among those developed in this thesis had to be chosen, it would be the Conan strategy enhanced as we proposed in Section 5.7, along with the Gahbonihno or Bayesian opponent model, since these were the ones outperforming all of our implemented agents and still performed decently in the tournaments in Section 5.

Such "lessons learned" from the work conducted in this thesis, can be used in future, in order to search deeper in the field of negotiations. For instance, it seems that an agent's performance dramatically changes each time one more agent participates in the negotiation. It can be noticed in our results that many of the implemented agents won their "small" tournaments, but did not fare well when more agents were added in the setting. Given this, a question arising is what is the number of opponent agents that delimits the effectiveness of each single agent strategy. Answering this and similar questions is interesting future work. Moreover, additional strategies and techniques could be explored, while the ones presented here could be employed and tested in different negotiation settings. Finally, it would be very interesting if multiagent negotiation ideas and algorithms such as the ones developed here could be in some way employed in the conceptually related subfield of multi-agent argumentation [54][56].

References

1. M. Wooldridge, N. Jennings, (1995). Intelligent agents: Theory and practice. *The Knowledge Engineering Review, Vol. 10:2, 115-152.*

2. M. Roderick, D. Messick. Negotiation as a Social Process. *Sage Publications.*

3. N. R. Jennings, T. J. Norman, P. Faratin, P. O'Brien & B. Odgers, (2000). Autonomous agents for business process. *Applied Artificial Intelligence, Vol. 14, Issue 2*.

4. G. E. Kersten, L. Gordon, (2003). Aspire: an integrated negotiation support system and software agents for e-business negotiation. *International Journal of Internet and Enterprise Management, Vol. 1, Issue 3.*

5. S. S. Fatima, M. Wooldridge, N. R. Jennings, (2004). An agenda-based framework for multi-issue negotiation. *Artificial Intelligence, Vol. 152, Issue 1, 1-45.*

6. T. Baarslag, K. V. Hindriks, (2013). Accepting optimally in automated negotiation with incomplete information. *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems, 715-722.*

7. C. M. Jonker, V. Robu, (2007). An agent architecture for multi-attribute negotiation using incomplete preference information. *Autonomous Agents and multi-agent systems, Vol. 15, Number 2, 221*.

8. S. S. Fatima, M. Wooldridge and N. R. Jennings, (2006). Multi-Issue Negotiation with Deadlines. *Journal of Artificial Intelligence Research Vol. 27, 381-417.*

9. D. L. Smith, D. G. Pruitt & P. J. D Carnevale, (1982). Matching and mismatching: The effect of own limit, other's toughness, and time pressure on concession rate in negotiation. *Journal of Personality and Social Psychology, 42(5), 876-883.*

10. S. Kraus, J. Wilkenfeld, G. Zlotkin, (1995). Multiagent negotiation under time constraints. *Artificial Intelligence, Vol. 75, Issue 2, 297-345*.

11. Roth, E. Alvin, (1985). Game-Theoretic Models of Bargaining. *Cambridge University Press*

12. S. S. Fatima, M. Wooldridge, N. R. Jenning, (2005). Bargaining with incomplete information. *Annals of Mathematics and Artificial Intelligence, Vol. 44, Number 3, 207*.

13. C. Li, J. Giampapa, and K. Sycara, (2003). A Review of Research Literature on Bilateral Negotiations. *CMU research showcase, Carnegie Mellon University*.

14. G. Chalkiadakis, E. Elkind, M. Wooldridge, (2011). Computational Aspects of Cooperative Game Theory. *Synthesis Lectures on Artificial Intelligence and Machine Learning, 1-168.*

15. J. F. Nash, (1950). The bargaining problem. *Econometrica, 28:155-162*.

16. M. Smorodinsky, E. Kalai, (1975). Other solutions to Nash's bargaining problem. *Econometrica, 43:513-518.*

17. Rubinstein, M. J. Osborne and Ariel, (1994). A Course in Game Theory. *MIT Press*.

18. A. Muthoo, (1999). Bargaining theory with applications. *Cambridge University Press*.

19. L. A. Busch and Q. Wen, (1995). Perfect Equilibria in a Negotiation Model. *Econometrica, Vol. 63. 545-565.*

20. F. G. Takako, (2015). Bayesian Nash Equilibrium. *Non-Cooperative Game Theory, 133-151, Springer.*

21. Y. Shoham, K. Leyton-Brown, (2008*)*. Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations. *Cambridge University Press.*

22. J. Tirole, D. Fudenberg, (1991). Game Theory. *MIT Press.*

23. A. Rubinstein, M. J. Osborne, (1994). A Course in Game Theory. *MIT Press*.

24. Z. I. William. Conflict Resolution. *Multilateral Negotiation Vol. 2, The school of Advanced International Studies, Washington D.C., USA.*

25. H. Raiffa, (1982). The Art and Science of Negotiation. *Cambridge: Harvard University Press*.

26. G. Lai, C. Li, K. Sycara, J. Giampapa, (2004). Literature Review on Multi-Attribute Negotiation. *Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-04-66.*

27. D. Kahneman, (2003). Maps of Bounded Rationality: Psychology for Behavioral Economics. *The American economic review, 93(5), 1449-1475*.

28. P. Faratin, (2000). Using similarity criteria to make negotiation trade-offs. *Boston, USA: 4th*

International Conference on Multi-Agent Systems. 119-126.

29. A. Dirkzwager, (2011). An investigation of autonomous negotiating agents dealing with time pressure.

30. T. Baarslag, K. Hindriks, C. Jonker, S. Kraus and R. Lin. The First Automated Negotiating Agents Competition (ANAC 2010). *New trends in Agent-Based automated negotiations, 113-135, part of the Studies in Computational Intelligence book series, SCI, Vol. 383*

31. T. Baarslag, W. Pasman, K. Hindriks, D. Tykhonov, W. Visser, M. Hendrikx, D. Feirstein, (2017). Negotiation User Guide.

32. R. Aydogan, D. Festen, K. V. Hindriks, C. M. Jonker. Alternating Offers Protocols for Multilateral Negotiation. *Modern Aproaches to Agent-based Complex Automated Negotiations, 153-167.*

33. T. Baarslag, M. J. C. Hendrikx, K. V. Hindriks, C. M. Jonker, (2016). Learning about the opponent in automated bilateral negotiation: a comprehensive survey of opponent modeling techniques. *Autonomous agents and Multi-agent systems, Vol. 30, Issue 5, 849-898.*

34. T. Bosse, C. M. Jonker & J. Treur, (2008). Automated formal analysis of human multi-issue negotiation processes. *Multiagent and Grid Systems. 213–233.*

35. R. Lin, S. Kraus, J. Wilkenfeld & J. Barry, (2008) Negotiating with bounded rational agents in environments with incomplete information using an automated agent. *Artificial Intelligence, Vol. 172, Issues 6-7, 823–851*.

36. L. Bolc, A. Bundy, P. Hayes, J. Siekmann, (1983). Machine Learning: An artificial Intelligence Approach. *Springer.*

37. A. G. Barto, R. S. Sutton and, (2012). Reinforcement Learning: An Introduction. *The MIT Press*

38. T. Baarslag, M. Hendrikx, K. Hindriks, and C. Jonker, (2012). Measuring the Performance of Online Opponent Models in Automated Bilateral Negotiation. *Advances in Artificial intelligence, 1-14.*

39. T. Baarslag, M. J. C. Hendrikx, K. V. Hindriks, (2016). A Survey of Opponent Modeling Techniques in Automated Negotiation. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent systems, 575-576.*

40. K. Hindriks & D. Tykhonov, (2008). Opponent Modelling in Automated Multi-Issue Negotiation Using Bayesian Learning. In *Proceedings of the 7th International joint Conference on Autonomous Agents & Multiagent systems, Vol. 1, 331-338.*

41. M. B. Adar, S. Nadav, and \E. Avshalom, (2013). Gahboninho: Strategy for Balancing Pressure and Compromise in Automated Negotiation. *Studies in Computational intelligence, Complex Automated Negotiations: Theories, Models and Software Competitions, 205-208, SCI, Vol. 435.*

42. M. Coggan, Research supervised by Prof. D. Precup, (2004). Exploration and Exploitation in Reinforcement Learning. *CRA-W DMP Project at McGll university.*

43. H. Jazayeriy, (2012). Generating pareto-optimal offers in bilateral negotiation with one-side uncertain importance weights. *Computing and Informatics, Vol. 31, 1235–1253.*

44. B. Alrayes, Ö. Kafalı, K. Stathis, (2014). CONAN: a heuristic strategy for Concurrent Negotiating Agents. *AAMAS '14.*

45. S. Fatima, S. Kraus, M. Wooldridge, (2014). Principles of Automated Negotiation. *Cambridge University Press*.

46. T. Baarslag, (2016). Exploring the strategy space on negotiating agents: a framework for bidding, learning and accepting in automated negotiation. *Springer thesis.*

47. W. Zartman. Multilateral Negotiation. *Conflict resolution, Vol. 2, The school of Advanced International Studies, Washington, D.C., USA.*

48. W. Zartman, J. Blaustein, (2007). Negotiation and conflict management: Essays on theory and practice. *Routledge.*

49. D. Zeng, K. Sycara. How can an agent learn to negotiate? *Intelligent Agents, Agent theories, Architectures and Languages, 233-244.*

50. R. B. Myerson. Game Theory. Analysis of Conflict. *Harvard University Press*.

51. K. Leyton-Brown & Y. Shoham, (2009). Multiagent Systems, Game-Theoretic, and Logical

Foundations. *Cambridge University Press*
52. F. Lopes, M. Wooldridge, A. Q. Novais, (2008). Negotiation among autonomous computational agents: principles, analysis and challenges. *Artificial Intelligence, 29: 1-44, Springer.*
53. P. Emerson, (2013). The original Borda count and partial voting. *Social choice and welfare, Vol. 40, Issue 2, 353-358.*
54. A. Kakas and P. Moraitis. Argumentation based decision making for autonomous agents. In *Proceedings of the second international Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '03), 883-890.*
55. T. Baarslag, (2013). Predicting the Performance of Opponent Models n Automated Negotiation. *Web Intelligence and Intelligent Agent Technologies. 2013 IEEE/WIC/ACM International Joint Conferences on. Vol. 2. IEEE.*
56. N. Spanoudakis, A. Kakas, P. Moraitis. Conflicts Resolution with the SoDA Methodology. *2nd International Workshop on Conflict Resolution in Decision Making (COREDEMA 2016), Post-Proceedings, Lecture Notes in Artificial Intelligence (LNAI), vol. 10238, Springer-Verlag.*

Appendix with experiments' analytics

9.1 Boltzmann Strategy without opponent model

Party Domain (DF = 1, preference profiles: 4)

Agent	Score
RandomDance	0,6447112721
MyAgent	0,6265700798
ParsAgent	0,5684692331
Atlas3	0,5627734568

Agent	Score
RandomDance	0,6524916755
Sengoku	0,6030768557
PokerFace	0,6025279824
Atlas3	0,5969270935
ParsAgent	0,5544754478
AgentH	0,5436265519
MyAgent	0,5250604278
Mercury	0,5043938713
AresParty	0,4766601999

Triangular Fight Domain (DF = 0.1, preference profiles: 3)

Agent	Score
ParsAgent	0,6423611111
Atlas3	0,6213333333
MyAgent	0,5680272109
RandomDance	0,5575163399

Agent	Score
PokerFace	0,716998672
ParsAgent	0,7166666667
RandomDance	0,6662674651
MyAgent	0,6588510354
Sengoku	0,6515761234
AresParty	0,6233532934
Atlas3	0,5674682699
AgentH	0,5648702595
Mercury	0,5507936508

Smart Grid Domain (DF = 1, preference profiles: 3)

Agent	Score
MyAgent	0,6225200088
ParsAgent	0,5570703567
RandomDance	0,5456432653
Atlas3	0,5394543388

Agent	Score
RandomDance	0,5832344305
Atlas3	0,5736562548
Sengoku	0,5700490102
PokerFace	0,5512203632
ParsAgent	0,5278002948
MyAgent	0,4957611736
AgentH	0,4934087715
Mercury	0,4645196662
AresParty	0,3988095238

University Domain (DF = 1, preference profiles: 4)

Agent	Score
MyAgent	0,8993299196
Atlas3	0,8736977594
RandomDance	0,8665651478
ParsAgent	0,8520788574

Agent	Score
RandomDance	0,8492148728
Atlas3	0,82197387
PokerFace	0,8176436778
Sengoku	0,8125712588
ParsAgent	0,7922099823
AgentH	0,7900017346
MyAgent	0,7619212974
Mercury	0,7535630152
AresParty	0,6891757696

Japan trip Domain (DF = 0.9, preference profiles: 3)

Agent	Score
MyAgent	0,9
Atlas3	0,9
RandomDance	0,9
ParsAgent	0,9

Agent	Score
AresParty	0,9126984127
MyAgent	0,9051984127
Sengoku	0,9015972222
Atlas3	0,9013789683
ParsAgent	0,8960714286
RandomDance	0,8942460317
PokerFace	0,8799900794
AgentH	0,8271130952
Mercury	0,8175496032

Domain Ace (DF = 0.2, preference profiles: 3)

Agent	Score
ParsAgent	0,5846956433
MyAgent	0,5489195504
Atlas3	0,46516343
RandomDance	0,433962469

Agent	Score
ParsAgent	0,5721016362
MyAgent	0,5299762328
PokerFace	0,5170597915
RandomDance	0,5129731617
Sengoku	0,5078215613
AresParty	0,4599124024
AgentH	0,4189175121
Atlas3	0,4129597147
Mercury	0,3984358949

KDomain (DF = 0.7, preference profiles: 3)

Agent	Score
ParsAgent	0,6634259259
MyAgent	0,6108333333
RandomDance	0,5584259259
Atlas3	0,4862037037

Agent	Score
ParsAgent	0,6328472222
AresParty	0,6225694444
PokerFace	0,6197222222
MyAgent	0,5996130952
RandomDance	0,5980853175
Sengoku	0,577906746
AgentH	0,5272718254
Atlas3	0,5223214286
Mercury	0,4526984127

Symposium Domain (DF = 1, preference profiles: 6)

Agent	Score
Atlas3	0,7496993446
MyAgent	0,7402329201
RandomDance	0,7213321777
ParsAgent	0,7158731991

Agent	Score
RandomDance	0,7578328124
Atlas3	0,7507716233
Sengoku	0,7481736563
PokerFace	0,7479756398
ParsAgent	0,7421319176
AgentH	0,7160473307
MyAgent	0,7118798214
Mercury	0,6927757372
AresParty	0,6418939471

Electric Vehicle Domain (DF = 0.3, preference profiles: 3)

Agent	Score
ParsAgent	0,7845679012
MyAgent	0,7759259259
Atlas3	0,6733024691
RandomDance	0,650617284

Agent	Score
MyAgent	0,7171957672
ParsAgent	0,6993386243
PokerFace	0,6726851852
Sengoku	0,6453207672
Atlas3	0,6200396825
RandomDance	0,619212963
AgentH	0,5948082011
Mercury	0,5347056878
AresParty	0,5244708995

Bank Robbery Domain (DF = 1, preference profiles: 5)

Agent	Score
MyAgent	0,8186196846
ParsAgent	0,7641846226
RandomDance	0,7459356934
Atlas3	0,7204101644

Agent	Score
ParsAgent	0,7275094092
PokerFace	0,7210516445
Atlas3	0,7199586086
RandomDance	0,7137750398
Sengoku	0,7062297879
MyAgent	0,7057799787
AgentH	0,6697497679
Mercury	0,6623525158
AresParty	0,6538499057

9.2 Boltzmann Strategy with Gahboninho opponent model, modeling opponents as one

Party Domain (DF = 1, preference profiles: 4)

Agent	Score
RandomDance	0,7105572214
MyAgent	0,6937360306
ParsAgent	0,6570497567
Atlas3	0,6143086265

Agent	Score
RandomDance	0,6742049071
PokerFace	0,6226866776
Sengoku	0,6181761618
Atlas3	0,6118642647
ParsAgent	0,5875746886
MyAgent	0,5659392745
AgentH	0,5604163392
Mercury	0,520342775
AresParty	0,5104693477

Triangular Fight Domain (DF = 0.1, preference profiles: 3)

Agent	Score
ParsAgent	0,6974358974
RandomDance	0,6125786164
MyAgent	0,6076923077
Atlas3	0,5685534591

Agent	Score
ParsAgent	0,7516235918
PokerFace	0,7347883598
RandomDance	0,6798280423
Sengoku	0,6646785951
MyAgent	0,662027833
AresParty	0,6357142857
AgentH	0,5943121693
Atlas3	0,5687169312
Mercury	0,5552910053

Smart Grid Domain (DF = 1, preference profiles: 3)

Agent	Score
MyAgent	0,5829951357
Atlas3	0,5576058397
RandomDance	0,5533956294
ParsAgent	0,5520742023

Agent	Score
Sengoku	0,5814768548
Atlas3	0,5789263647
ParsAgent	0,575627353
RandomDance	0,5746958002
PokerFace	0,5586528248
MyAgent	0,5078675463
AgentH	0,5038582186
Mercury	0,4562649525
AresParty	0,4126984127

University Domain (DF = 1, preference profiles: 4)

Agent	Score
MyAgent	0,8953650267
Atlas3	0,8932632163
RandomDance	0,8826866049
ParsAgent	0,8660434453

Agent	Score
RandomDance	0,8581566396
PokerFace	0,8319540718
Atlas3	0,8294901843
Sengoku	0,8201363911
ParsAgent	0,8079347721
AgentH	0,7947648428
MyAgent	0,7804465964
Mercury	0,7570003264
AresParty	0,7138599106

Japan trip Domain (DF = 0.9, preference profiles: 3)

Agent	Score
MyAgent	0,9
Atlas3	0,9
RandomDance	0,9
ParsAgent	0,9

Agent	Score
AresParty	0,9150793651
Sengoku	0,9023710317
Atlas3	0,9010813492
ParsAgent	0,8975694444
RandomDance	0,8961607143
MyAgent	0,8953769841
PokerFace	0,8803174603
AgentH	0,8208035714
Mercury	0,8100297619

Domain Ace (DF = 0.2, preference profiles: 3)

Agent	Score
ParsAgent	0,7412259075
RandomDance	0,5665500086
MyAgent	0,5149680758
Atlas3	0,430540729

Agent	Score
ParsAgent	0,6024695078
PokerFace	0,5426255592
RandomDance	0,5258574476
Sengoku	0,5179974183
MyAgent	0,4918454661
AresParty	0,4715188677
AgentH	0,4327167352
Atlas3	0,4065235735
Mercury	0,4047250976

KDomain (DF = 0.7, preference profiles: 3)

Agent	Score
ParsAgent	0,875
RandomDance	0,7255555556
MyAgent	0,6245488888
Atlas3	0,5169444444

Agent	Score
ParsAgent	0,6980853175
PokerFace	0,670625
AresParty	0,664781746
RandomDance	0,6337896825
Sengoku	0,5946924603
MyAgent	0,5696329365
AgentH	0,5447519841
Atlas3	0,5269444444
Mercury	0,4538293651

Symposium Domain (DF = 1, preference profiles: 6)

Agent	Score
Atlas3	0,7573995726
MyAgent	0,7407625073
RandomDance	0,7307522158
ParsAgent	0,7226311663

Agent	Score
RandomDance	0,7608799373
Atlas3	0,7544679105
PokerFace	0,7536152789
Sengoku	0,7533888806
ParsAgent	0,7437390087
MyAgent	0,7262773689
AgentH	0,7227859625
Mercury	0,6958998751
AresParty	0,6551724138

Electric Vehicle Domain (DF = 0.3, preference profiles: 3)

Agent	Score
ParsAgent	0,9425925926
RandomDance	0,8475308642
MyAgent	0,724537037
Atlas3	0,6148148148

Agent	Score
ParsAgent	0,7418650794
PokerFace	0,7050595238
MyAgent	0,6826554233
RandomDance	0,6713789683
Sengoku	0,6688657407
Atlas3	0,619130291
AgentH	0,5882771164
Mercury	0,5344742063
AresParty	0,524537037

Bank Robbery Domain (DF = 1, preference profiles: 5)

Agent	Score
MyAgent	0,8185994096
ParsAgent	0,7869197004
RandomDance	0,7511331699
Atlas3	0,7477960207

Agent	Score
ParsAgent	0,7424828579
PokerFace	0,7358909886
Atlas3	0,730166104
MyAgent	0,7284394284
Sengoku	0,7171362239
RandomDance	0,7170460157
Mercury	0,6831245316
AgentH	0,6807253048
AresParty	0,6795001473

9.3 Boltzmann Strategy with Gahboninho opponent model

Party Domain (DF = 1, preference profiles: 4)

Agent	Score
MyAgent	0,5718693344
RandomDance	0,5358788728
Atlas3	0,4954880573
ParsAgent	0,4630901339

Agent	Score
RandomDance	0,6201093905
Sengoku	0,5771426245
Atlas3	0,5731219044
PokerFace	0,5649086291
AgentH	0,5297995922
ParsAgent	0,5212238288
Mercury	0,4660487368
AresParty	0,4496016304
MyAgent	0,4343462409

Triangular Fight Domain (DF = 0.1, preference profiles: 3)

Agent	Score
MyAgent	0,5423076923
RandomDance	0,5044871795
Atlas3	0,4993333333
ParsAgent	0,474

Agent	Score
MyAgent	0,685915493
PokerFace	0,6764940239
ParsAgent	0,6677640604
RandomDance	0,6207422134
Sengoku	0,6152129817
AresParty	0,5831004657
AgentH	0,5731204258
Atlas3	0,5393145161
Mercury	0,5315440689

Smart Grid Domain (DF = 1, preference profiles: 3)

Agent	Score
MyAgent	0,5902929278
ParsAgent	0,5060897697
Atlas3	0,504213555
RandomDance	0,442768399

Agent	Score
RandomDance	0,5612231
Atlas3	0,5508516413
Sengoku	0,5503452241
PokerFace	0,5298440179
ParsAgent	0,5053273378
AgentH	0,4989624236
MyAgent	0,4525570948
Mercury	0,4339558281
AresParty	0,371031746

University Domain (DF = 1, preference profiles: 4)

Agent	Score
MyAgent	0,8562785433
Atlas3	0,8093541865
RandomDance	0,8046295149
ParsAgent	0,7674592797

Agent	Score
RandomDance	0,8204452163
Atlas3	0,8050697331
PokerFace	0,8014428033
Sengoku	0,7857271682
AgentH	0,7827941519
ParsAgent	0,7606582929
Mercury	0,7170729818
MyAgent	0,6898016967
AresParty	0,6446650124

Japan trip Domain (DF = 0.9, preference profiles: 3)

Agent	Score
MyAgent	0,9
Atlas3	0,9
RandomDance	0,9
ParsAgent	0,9

Agent	Score
MyAgent	0,9099702381
AresParty	0,9099206349
Sengoku	0,9015079365
Atlas3	0,9005853175
RandomDance	0,8978174603
PokerFace	0,8925595238
ParsAgent	0,8917857143
AgentH	0,834702381
Mercury	0,823452381

Domain Ace (DF = 0.2, preference profiles: 3)

Agent	Score
MyAgent	0,4937249133
ParsAgent	0,4811157895
RandomDance	0,4105066008
Atlas3	0,3910230933

Agent	Score
MyAgent	0,5548749198
ParsAgent	0,5448065887
PokerFace	0,4914638505
RandomDance	0,4881042728
Sengoku	0,4688627079
AresParty	0,4400024
AgentH	0,4157360333
Atlas3	0,3981351091
Mercury	0,3836807935

KDomain (DF = 0.7, preference profiles: 3)

Agent	Score
MyAgent	0,6444444444
ParsAgent	0,5642592593
RandomDance	0,4853703704
Atlas3	0,4335185185

Agent	Score
ParsAgent	0,6261309524
PokerFace	0,6030753968
RandomDance	0,5783730159
AresParty	0,5711904762
MyAgent	0,5629960317
Sengoku	0,5609920635
AgentH	0,5112301587
Atlas3	0,5030952381
Mercury	0,4295039683

Symposium Domain (DF = 1, preference profiles: 6)

Agent	Score
MyAgent	0,747051783
Atlas3	0,720653618
RandomDance	0,7052912271
ParsAgent	0,6962197855

Agent	Score
RandomDance	0,7373483075
Atlas3	0,7363147319
Sengoku	0,7309725032
ParsAgent	0,730747299
PokerFace	0,7280485331
AgentH	0,7124003988
Mercury	0,6717074753
MyAgent	0,6630596642
AresParty	0,6243659972

Electric Vehicle Domain (DF = 0.3, preference profiles: 3)

Agent	Score
MyAgent	0,85
ParsAgent	0,8055555556
RandomDance	0,6384259259
Atlas3	0,6307098765

Agent	Score
MyAgent	0,7033730159
ParsAgent	0,6983465608
PokerFace	0,6562169312
Sengoku	0,6387070106
Atlas3	0,6211970899
RandomDance	0,6151124339
AgentH	0,5869543651
AresParty	0,523478836
Mercury	0,5226521164

Bank Robbery Domain (DF = 1, preference profiles: 5)

Agent	Score
MyAgent	0,7576746945
ParsAgent	0,6927939374
RandomDance	0,6657591345
Atlas3	0,6636153628

Agent	Score
ParsAgent	0,7076570327
PokerFace	0,7060095073
Atlas3	0,7021322904
Sengoku	0,6900160388
RandomDance	0,6882200002
AgentH	0,6761710109
MyAgent	0,6610350855
Mercury	0,65546801
AresParty	0,6376609987

9.4 Boltzmann Strategy with Bayesian opponent model

Party Domain (DF = 1, preference profiles: 4)

Agent	Score
MyAgent	0,6616843331
RandomDance	0,661324557
ParsAgent	0,5991312028
Atlas3	0,5846974034

Agent	Score
RandomDance	0,6476649043
Sengoku	0,6013026816
PokerFace	0,5977185965
Atlas3	0,5934056885
ParsAgent	0,554393472
AgentH	0,5380894029
MyAgent	0,5015611239
Mercury	0,4906251957
AresParty	0,4651928925

Triangular Fight Domain (DF = 0.1, preference profiles: 3)

Agent	Score
RandomDance	0,608974359
ParsAgent	0,60625
Atlas3	0,5986666667
MyAgent	0,5541666667

Agent	Score
ParsAgent	0,7072372999
PokerFace	0,7054108216
RandomDance	0,6654050465
Sengoku	0,6481854839
MyAgent	0,6412121212
AresParty	0,5960159363
Atlas3	0,5712431694
AgentH	0,5648666667
Mercury	0,5582666667

Smart Grid Domain (DF = 1, preference profiles: 3)

Agent	Score	Agent	Score
MyAgent	0,620852766	Atlas3	0,5819344565
ParsAgent	0,577298617	RandomDance	0,5774969358
Atlas3	0,5554557239	Sengoku	0,5758347319
RandomDance	0,5207784034	PokerFace	0,5537471001
		ParsAgent	0,5318109795
		MyAgent	0,4984624762
		AgentH	0,4936132674
		Mercury	0,4468553411
		AresParty	0,3670634921

University Domain (DF = 1, preference profiles: 4)

Agent	Score	Agent	Score
MyAgent	0,8999102983	RandomDance	0,8452475022
Atlas3	0,8732056042	Atlas3	0,8178246532
RandomDance	0,866842779	PokerFace	0,81382214
ParsAgent	0,846446548	Sengoku	0,8103126438
		ParsAgent	0,7876614305
		AgentH	0,7863708455
		Mercury	0,7462550026
		MyAgent	0,7424445189
		AresParty	0,6688182721

Japan trip Domain (DF = 0.9, preference profiles: 3)

Agent	Score	Agent	Score
MyAgent	0,9	AresParty	0,9115079365
Atlas3	0,9	MyAgent	0,9065674603
RandomDance	0,9	Sengoku	0,9016666667
ParsAgent	0,9	Atlas3	0,9011011905
		ParsAgent	0,8964087302
		RandomDance	0,8960019841
		PokerFace	0,8800694444
		AgentH	0,8357638889
		Mercury	0,8233035714

Domain Ace (DF = 0.2, preference profiles: 3)

Agent	Score	Agent	Score
ParsAgent	0,5845330655	ParsAgent	0,573625765
MyAgent	0,5649118349	MyAgent	0,5257072376
Atlas3	0,4647045274	PokerFace	0,5077459792
RandomDance	0,4566159712	RandomDance	0,5052572911
		Sengoku	0,4916178812
		AresParty	0,4514885915
		AgentH	0,4219791589
		Atlas3	0,4195259011
		Mercury	0,3980712773

KDomain (DF = 0.7, preference profiles: 3)

Agent	Score
RandomDance	0,542037037
ParsAgent	0,5075925926
Atlas3	0,4906481481
MyAgent	0,4672222222

Agent	Score
ParsAgent	0,6108630952
PokerFace	0,5998214286
AresParty	0,5896924603
RandomDance	0,5896428571
Sengoku	0,5652579365
Atlas3	0,520218254
MyAgent	0,5101388889
AgentH	0,5069444444
Mercury	0,4287996032

Symposium Domain (DF = 1, preference profiles: 6)

Agent	Score
MyAgent	0,7494016448
Atlas3	0,7437195658
RandomDance	0,7393544876
ParsAgent	0,7240019636

Agent	Score
RandomDance	0,7553891578
Sengoku	0,7489566587
PokerFace	0,7479234764
Atlas3	0,7476991542
ParsAgent	0,7408158049
AgentH	0,7152656099
MyAgent	0,7026284207
Mercury	0,6876122401
AresParty	0,6387168815

Electric Vehicle Domain (DF = 0.3, preference profiles: 3)

Agent	Score
ParsAgent	0,7907407407
MyAgent	0,774691358
RandomDance	0,6703703704
Atlas3	0,6490740741

Agent	Score
ParsAgent	0,7038359788
MyAgent	0,6966269841
PokerFace	0,6666005291
Sengoku	0,6457506614
Atlas3	0,6290013228
RandomDance	0,6143518519
AgentH	0,5868882275
Mercury	0,5337136243
AresParty	0,5213624339

Bank Robbery Domain (DF = 1, preference profiles: 5)

Agent	Score
MyAgent	0,8104959634
ParsAgent	0,7596350812
RandomDance	0,7425750783
Atlas3	0,7207102788

Agent	Score
ParsAgent	0,7263668654
PokerFace	0,722756779
Atlas3	0,719089242
RandomDance	0,7116795311
Sengoku	0,708638573
MyAgent	0,6969975745
AgentH	0,6720744957
Mercury	0,6720223388
AresParty	0,6539616213

9.5 Smart Meta - Strategy with Gahboninho opponent model

Party Domain (DF = 1, preference profiles: 4)

Agent	Score
myAgent	0,6239564344
Atlas3	0,5751328198
RandomDance	0,5689395869
ParsAgent	0,5478274856

Agent	Score
RandomDance	0,6385488104
Sengoku	0,5932439932
Atlas3	0,5857963768
PokerFace	0,5840963883
AgentH	0,5516730462
ParsAgent	0,5374722832
myAgent	0,484205333
Mercury	0,4778285542
AresParty	0,4647623365

Triangular Fight Domain (DF = 0.1, preference profiles: 3)

Agent	Score
myAgent	0,552
RandomDance	0,525
ParsAgent	0,502
Atlas3	0,4801282051

Agent	Score
myAgent	0,6824596774
PokerFace	0,6765873016
ParsAgent	0,6638545953
RandomDance	0,6320053121
Sengoku	0,609127789
AresParty	0,5756
AgentH	0,5634130146
Atlas3	0,5348058902
Mercury	0,5260146374

Smart Grid Domain (DF = 1, preference profiles: 3)

Agent	Score
myAgent	0,6326275303
ParsAgent	0,5903403287
Atlas3	0,5384024352
RandomDance	0,5026327553

Agent	Score
RandomDance	0,5839219873
Sengoku	0,5788051374
Atlas3	0,5718213412
PokerFace	0,5710793954
AgentH	0,5324574881
ParsAgent	0,529516522
myAgent	0,5134834776
Mercury	0,4566423432
AresParty	0,380952381

University Domain (DF = 1, preference profiles: 4)

Agent	Score
myAgent	0,925903153
Atlas3	0,8836567631
RandomDance	0,8562958306
ParsAgent	0,8555734799

Agent	Score
RandomDance	0,8391677546
Atlas3	0,818665017
PokerFace	0,8156049972
Sengoku	0,8091947084
ParsAgent	0,794780944
AgentH	0,7897129986
myAgent	0,7480968918
Mercury	0,7402804506
AresParty	0,6950248756

Japan trip Domain (DF = 0.9, preference profiles: 3)

Agent	Score	Agent	Score
myAgent	0,9	AresParty	0,9123015873
Atlas3	0,9	Sengoku	0,9016765873
RandomDance	0,9	Atlas3	0,9008531746
ParsAgent	0,9	myAgent	0,8996031746
		RandomDance	0,8947222222
		ParsAgent	0,8934821429
		PokerFace	0,8798313492
		AgentH	0,8267757937
		Mercury	0,8191468254

Domain Ace (DF = 0.2, preference profiles: 3)

Agent	Score	Agent	Score
ParsAgent	0,4787539377	ParsAgent	0,5458873647
myAgent	0,458289498	myAgent	0,5184432139
RandomDance	0,4088190002	RandomDance	0,4981757447
Atlas3	0,4045657046	PokerFace	0,4959742586
		Sengoku	0,4717529971
		AresParty	0,4409156112
		AgentH	0,4243035243
		Atlas3	0,3964119081
		Mercury	0,3787355602

KDomain (DF = 0.7, preference profiles: 3)

Agent	Score	Agent	Score
myAgent	0,6530555556	ParsAgent	0,6261607143
ParsAgent	0,6139814815	PokerFace	0,5934623016
RandomDance	0,4978703704	AresParty	0,5899801587
Atlas3	0,4563888889	RandomDance	0,5838492063
		Sengoku	0,5787896825
		myAgent	0,5624007937
		AgentH	0,5337599206
		Atlas3	0,508234127
		Mercury	0,4318948413

Symposium Domain (DF = 1, preference profiles: 6)

Agent	Score	Agent	Score
myAgent	0,8473533634	RandomDance	0,7660906037
Atlas3	0,8034826126	ParsAgent	0,760034575
ParsAgent	0,8024055011	Atlas3	0,7564559178
RandomDance	0,8005671787	PokerFace	0,7555730872
		Sengoku	0,7530553565
		AgentH	0,738781916
		myAgent	0,7255401908
		Mercury	0,6794130657
		AresParty	0,6337185726

Electric Vehicle Domain (DF = 0.3, preference profiles: 3)

Agent	Score
myAgent	0,837654321
ParsAgent	0,8333333333
RandomDance	0,6910493827
Atlas3	0,6206790123

Agent	Score
myAgent	0,7291005291
ParsAgent	0,7198412698
PokerFace	0,6827050265
Sengoku	0,6525793651
RandomDance	0,6356150794
Atlas3	0,6268849206
AgentH	0,5952380952
Mercury	0,5330026455
AresParty	0,5215608466

Bank Robbery Domain (DF = 1, preference profiles: 5)

Agent	Score
myAgent	0,819562567
ParsAgent	0,7402965367
Atlas3	0,7188670177
RandomDance	0,7057642022

Agent	Score
ParsAgent	0,721887129
PokerFace	0,7211149239
Atlas3	0,7177788452
myAgent	0,7095269556
Sengoku	0,7066351868
RandomDance	0,7019864104
AgentH	0,6892264078
Mercury	0,6611749528
AresParty	0,635405313

9.6 Smart Meta - Strategy with Bayesian opponent model

Party Domain (DF = 1, preference profiles: 4)

Agent	Score
BayesianAgent	0,602248694
RandomDance	0,5829419845
Atlas3	0,5521754283
ParsAgent	0,5313371723

Agent	Score
RandomDance	0,6355457039
Sengoku	0,5957655128
PokerFace	0,591496632
Atlas3	0,588445721
AgentH	0,5484194595
ParsAgent	0,5443716893
BayesianAgent	0,4783794497
Mercury	0,4750701998
AresParty	0,4585144152

Triangular Fight Domain (DF = 0.1, preference profiles: 3)

Agent	Score
BayesianAgent	0,5470588235
RandomDance	0,488
Atlas3	0,4782312925
ParsAgent	0,4694444444

Agent	Score
BayesianAgent	0,6854251012
PokerFace	0,6768463074
ParsAgent	0,6668049793
RandomDance	0,6272666667
Sengoku	0,6126436782
AresParty	0,580904857
AgentH	0,5666666667
Atlas3	0,5410774411
Mercury	0,5365205843

Smart Grid Domain (DF = 1, preference profiles: 3)

Agent	Score
BayesianAgent	0,6513707479
ParsAgent	0,5841006121
RandomDance	0,5395758185
Atlas3	0,5380968685

Agent	Score
RandomDance	0,5879273881
Atlas3	0,570025377
Sengoku	0,5614483862
PokerFace	0,5569128602
AgentH	0,5218659227
ParsAgent	0,5118875281
BayesianAgent	0,4877312797
Mercury	0,440571219
AresParty	0,3908730159

University Domain (DF = 1, preference profiles: 4)

Agent	Score
BayesianAgent	0,8828390047
Atlas3	0,8747333597
RandomDance	0,8250622882
ParsAgent	0,8206890196

Agent	Score
RandomDance	0,8343999999
Atlas3	0,8164685931
PokerFace	0,8154891214
Sengoku	0,8068018363
ParsAgent	0,7905588067
AgentH	0,787274103
BayesianAgent	0,7417886898
Mercury	0,7407558656
AresParty	0,6911984088

Japan trip Domain (DF = 0.9, preference profiles: 3)

Agent	Score
Atlas3	0,9
BayesianAgent	0,8990740741
ParsAgent	0,8975925926
RandomDance	0,89

Agent	Score
AresParty	0,9125
Sengoku	0,9018452381
Atlas3	0,900952381
BayesianAgent	0,8993452381
RandomDance	0,8963888889
ParsAgent	0,894593254
PokerFace	0,8768452381
AgentH	0,8263095238
Mercury	0,8145238095

Domain Ace (DF = 0.2, preference profiles: 3)

Agent	Score
ParsAgent	0,4798752913
BayesianAgent	0,4687716125
RandomDance	0,4263872953
Atlas3	0,3894257759

Agent	Score
ParsAgent	0,5362931032
BayesianAgent	0,519349727
PokerFace	0,4867987898
RandomDance	0,476318897
Sengoku	0,4709570595
AresParty	0,4410494602
AgentH	0,4150933136
Atlas3	0,3971359912
Mercury	0,3878834697

KDomain (DF = 0.7, preference profiles: 3)

Agent	Score
BayesianAgent	0,6275925926
ParsAgent	0,5972222222
RandomDance	0,5314814815
Atlas3	0,4557407407

Agent	Score
ParsAgent	0,6212896825
PokerFace	0,5936309524
AresParty	0,5860912698
RandomDance	0,581140873
Sengoku	0,5780952381
BayesianAgent	0,5738392857
AgentH	0,5297420635
Atlas3	0,5103174603
Mercury	0,4394246032

Symposium Domain (DF = 1, preference profiles: 6)

Agent	Score
BayesianAgent	0,8467889067
Atlas3	0,8078267968
RandomDance	0,7985677385
ParsAgent	0,600617284

Agent	Score
RandomDance	0,7674984574
ParsAgent	0,7601665572
Atlas3	0,756032281
PokerFace	0,7541493626
Sengoku	0,7534261067
AgentH	0,7385877742
BayesianAgent	0,7243837339
Mercury	0,6775799158
AresParty	0,6335428919

Electric Vehicle Domain (DF = 0.3, preference profiles: 3)

Agent	Score
ParsAgent	0,600617284
BayesianAgent	0,5956790123
RandomDance	0,5046296296
Atlas3	0,5

Agent	Score
BayesianAgent	0,7265873016
ParsAgent	0,7208994709
PokerFace	0,678505291
Sengoku	0,6476521164
RandomDance	0,6368220899
Atlas3	0,6276785714
AgentH	0,5998015873
Mercury	0,5253141534
AresParty	0,5185846561

Bank Robbery Domain (DF = 1, preference profiles: 5)

Agent	Score
BayesianAgent	0,8173553881
ParsAgent	0,7402984987
Atlas3	0,7147459204
RandomDance	0,6952536563

Agent	Score
ParsAgent	0,7244480466
PokerFace	0,7226951455
Atlas3	0,7183678079
BayesianAgent	0,7095649075
Sengoku	0,7060294528
RandomDance	0,7034515323
AgentH	0,6908758581
Mercury	0,6585368606
AresParty	0,638336238

9.7 Hybrid MGT – Smart Meta - Strategy with Gahboninho opponent model

Party Domain (DF = 1, preference profiles: 4)

Agent	Score
myAgent	0,6530314399
RandomDance	0,564895729
Atlas3	0,546223743
ParsAgent	0,543067356

Agent	Score
RandomDance	0,620291272
Sengoku	0,5952467958
Atlas3	0,580360644
PokerFace	0,5674971739
ParsAgent	0,5446729223
AgentH	0,5390057717
myAgent	0,476188276
Mercury	0,4675982309
AresParty	0,4591675093

Triangular Fight Domain (DF = 0.1, preference profiles: 3)

Agent	Score
myAgent	0,5470588235
RandomDance	0,5044871795
Atlas3	0,4829931973
ParsAgent	0,4775510204

Agent	Score
myAgent	0,6816901408
PokerFace	0,6719123506
ParsAgent	0,6710835059
RandomDance	0,6256461233
Sengoku	0,6146801347
AresParty	0,5783765802
AgentH	0,5709163347
Atlas3	0,5355782313
Mercury	0,527

Smart Grid Domain (DF = 1, preference profiles: 3)

Agent	Score
myAgent	0,6123537757
ParsAgent	0,5515462613
Atlas3	0,5124074911
RandomDance	0,4548844401

Agent	Score
RandomDance	0,5823373366
Atlas3	0,5670898599
Sengoku	0,558791985
ParsAgent	0,5282455094
PokerFace	0,515709488
AgentH	0,4876871271
myAgent	0,4543073788
Mercury	0,4349271298
AresParty	0,3996023857

University Domain (DF = 1, preference profiles: 4)

Agent	Score
myAgent	0,9430480298
RandomDance	0,8791876055
Atlas3	0,8740927202
ParsAgent	0,8674003421

Agent	Score
RandomDance	0,841680846
Atlas3	0,8190654746
PokerFace	0,8189350049
Sengoku	0,8034242786
ParsAgent	0,8014592823
AgentH	0,791337959
myAgent	0,7655462984
Mercury	0,7347220039
AresParty	0,6893300248

Japan trip Domain (DF = 0.9, preference profiles: 3)

Agent	Score
myAgent	0,9007407407
Atlas3	0,9
ParsAgent	0,8998148148
RandomDance	0,8946296296

Agent	Score
AresParty	0,9103174603
myAgent	0,9084027778
Sengoku	0,9013888889
Atlas3	0,9012698413
RandomDance	0,8983333333
ParsAgent	0,8925694444
PokerFace	0,8854563492
AgentH	0,832172619
Mercury	0,8220535714

Domain Ace (DF = 0.2, preference profiles: 3)

Agent	Score
myAgent	0,5165967882
ParsAgent	0,4850650408
RandomDance	0,4077604507
Atlas3	0,3904159623

Agent	Score
myAgent	0,5620055705
ParsAgent	0,5324748495
PokerFace	0,4910552241
RandomDance	0,4878624205
Sengoku	0,468698791
AresParty	0,4377178241
AgentH	0,4221999008
Atlas3	0,3955118902
Mercury	0,3875715668

KDomain (DF = 0.7, preference profiles: 3)

Agent	Score
myAgent	0,6935185185
ParsAgent	0,6115740741
RandomDance	0,515
Atlas3	0,432037037

Agent	Score
ParsAgent	0,6220039683
PokerFace	0,5901686508
AresParty	0,5814285714
RandomDance	0,5708730159
myAgent	0,5699404762
Sengoku	0,5573809524
AgentH	0,5219146825
Atlas3	0,5125099206
Mercury	0,4263392857

Symposium Domain (DF = 1, preference profiles: 6)

Agent	Score
myAgent	0,7460122318
Atlas3	0,7346594744
RandomDance	0,7171663011
ParsAgent	0,7022963685

Agent	Score
RandomDance	0,7460205137
Atlas3	0,7430850879
Sengoku	0,7405926262
ParsAgent	0,7365852411
PokerFace	0,7356019116
AgentH	0,7171368876
myAgent	0,686984596
Mercury	0,679369726
AresParty	0,6436256598

Electric Vehicle Domain (DF = 0.3, preference profiles: 3)

Agent	Score
myAgent	0,8271604938
ParsAgent	0,8160493827
RandomDance	0,6919753086
Atlas3	0,6145061728

Agent	Score
ParsAgent	0,7091931217
myAgent	0,7063492063
PokerFace	0,6674768519
Sengoku	0,6428571429
RandomDance	0,6259920635
Atlas3	0,6237268519
AgentH	0,5962632275
Mercury	0,5208167989
AresParty	0,5186507937

Bank Robbery Domain (DF = 1, preference profiles: 5)

Agent	Score
myAgent	0,8218078833
ParsAgent	0,7350161761
Atlas3	0,697673011
RandomDance	0,686152326

Agent	Score
ParsAgent	0,7190846115
Atlas3	0,7084936992
PokerFace	0,7075429407
Sengoku	0,6969637664
RandomDance	0,6934966401
myAgent	0,6844085529
AgentH	0,6779137654
Mercury	0,6530566998
AresParty	0,6356261677

9.8 Hybrid MGT – Smart Meta - Strategy with Bayesian opponent model

Party Domain (DF = 1, preference profiles: 4)

Agent	Score
myAgent	0,647564136
RandomDance	0,5626580742
ParsAgent	0,5393912283
Atlas3	0,5292843982

Agent	Score
RandomDance	0,6235860011
Sengoku	0,5913130397
Atlas3	0,5800466191
PokerFace	0,5710185328
ParsAgent	0,544208209
AgentH	0,530341731
myAgent	0,4770812784
Mercury	0,4689885
AresParty	0,4572875582

Triangular Fight Domain (DF = 0.1, preference profiles: 3)

Agent	Score
myAgent	0,552
RandomDance	0,5189542484
ParsAgent	0,4952380952
Atlas3	0,4928104575

Agent	Score
myAgent	0,6851405622
PokerFace	0,6810756972
ParsAgent	0,6640657084
RandomDance	0,624
Sengoku	0,6167672703
AresParty	0,5799468792
AgentH	0,5666001331
Atlas3	0,5354795439
Mercury	0,5306719894

Smart Grid Domain (DF = 1, preference profiles: 3)

Agent	Score
myAgent	0,6495832642
ParsAgent	0,5616738776
Atlas3	0,5212724151
RandomDance	0,4970876516

Agent	Score
RandomDance	0,5707881378
Atlas3	0,5573161668
Sengoku	0,5538121028
PokerFace	0,5171985991
ParsAgent	0,5121967192
AgentH	0,5038591886
myAgent	0,4448926851
Mercury	0,4245944659
AresParty	0,3888888889

University Domain (DF = 1, preference profiles: 4)

Agent	Score
myAgent	0,9331681322
RandomDance	0,8758523605
Atlas3	0,8749996742
ParsAgent	0,8613723413

Agent	Score
RandomDance	0,8438250461
PokerFace	0,8206108389
Atlas3	0,8194871847
Sengoku	0,8057158674
ParsAgent	0,8016445804
AgentH	0,7938547754
myAgent	0,766482321
Mercury	0,7405230451
AresParty	0,6898263027

Japan trip Domain (DF = 0.9, preference profiles: 3)

Agent	Score
myAgent	0,9
RandomDance	0,9
Atlas3	0,9
ParsAgent	0,9

Agent	Score
AresParty	0,9103174603
myAgent	0,9082242063
Sengoku	0,9018452381
Atlas3	0,9013293651
RandomDance	0,9006845238
ParsAgent	0,8927777778
PokerFace	0,8841865079
AgentH	0,8358035714
Mercury	0,8267162698

Domain Ace (DF = 0.2, preference profiles: 3)

Agent	Score
myAgent	0,4913903934
ParsAgent	0,4838560806
RandomDance	0,4087856273
Atlas3	0,3908668838

Agent	Score
myAgent	0,5629978126
ParsAgent	0,5416528908
PokerFace	0,4904401067
RandomDance	0,4861330854
Sengoku	0,4679524983
AresParty	0,4404441175
AgentH	0,4152064528
Atlas3	0,394597688
Mercury	0,3802957742

KDomain (DF = 0.7, preference profiles: 3)

Agent	Score
myAgent	0,6367592593
ParsAgent	0,5576851852
RandomDance	0,52
Atlas3	0,4191666667

Agent	Score
ParsAgent	0,6175198413
PokerFace	0,5949702381
AresParty	0,5872420635
myAgent	0,5777777778
RandomDance	0,5672123016
Sengoku	0,5639583333
AgentH	0,5263194444
Atlas3	0,5127579365
Mercury	0,4329761905

Symposium Domain (DF = 1, preference profiles: 6)

Agent	Score
myAgent	0,7488798034
Atlas3	0,736914488
RandomDance	0,7113124795
ParsAgent	0,7078256452

Agent	Score
Atlas3	0,7434841373
RandomDance	0,7431976877
Sengoku	0,7398801056
ParsAgent	0,7367134371
PokerFace	0,7345320874
AgentH	0,7182335784
myAgent	0,6848797841
Mercury	0,6767429563
AresParty	0,643560337

Electric Vehicle Domain (DF = 0.3, preference profiles: 3)

Agent	Score
myAgent	0,8179012346
ParsAgent	0,8111111111
RandomDance	0,7032407407
Atlas3	0,6212962963

Agent	Score
ParsAgent	0,7099867725
myAgent	0,708994709
PokerFace	0,6639219577
Sengoku	0,6511904762
RandomDance	0,6291005291
Atlas3	0,615922619
AgentH	0,5861937831
Mercury	0,5304232804
AresParty	0,519510582

Bank Robbery Domain (DF = 1, preference profiles: 5)

Agent	Score
myAgent	0,8320297842
ParsAgent	0,7403001912
Atlas3	0,6980494149
RandomDance	0,6869110515

Agent	Score
ParsAgent	0,7208419301
PokerFace	0,7087337266
Atlas3	0,7086787989
Sengoku	0,6977193411
RandomDance	0,6917264855
myAgent	0,6842885618
AgentH	0,6793841071
Mercury	0,6545160684
AresParty	0,6368354049

9.9 Conan Strategy without opponent model

Party Domain (DF = 1, preference profiles: 4)

Agent	Score
RandomDance	0,8121671476
ParsAgent	0,8052537348
Atlas3	0,8037255517
myAgent	0,6026208753

Agent	Score
RandomDance	0,73260527
PokerFace	0,6973278275
Sengoku	0,6769234692
Atlas3	0,6741756922
ParsAgent	0,6660338604
AresParty	0,6205174222
AgentH	0,6092898025
Mercury	0,5689031701
myAgent	0,5547036684

Triangular Fight Domain (DF = 0.1, preference profiles: 3)

Agent	Score
ParsAgent	0,8666666667
RandomDance	0,7685534591
myAgent	0,5641509434
Atlas3	0,5608974359

Agent	Score
RandomDance	0,6709029846
Atlas3	0,6674143229
PokerFace	0,651677672
Sengoku	0,6483737566
ParsAgent	0,6482929591
AgentH	0,5712758207
AresParty	0,5416666667
Mercury	0,5366515416
myAgent	0,535014691

Smart Grid Domain (DF = 1, preference profiles: 3)

Agent	Score
ParsAgent	0,8701513801
Atlas3	0,8340662675
RandomDance	0,7823717866
myAgent	0,5881271805

Agent	Score
RandomDance	0,6709029846
Atlas3	0,6674143229
PokerFace	0,651677672
Sengoku	0,6483737566
ParsAgent	0,6482929591
AgentH	0,5712758207
AresParty	0,5416666667
Mercury	0,5366515416
myAgent	0,535014691

University Domain (DF = 1, preference profiles: 4)

Agent	Score
Atlas3	0,9479415431
RandomDance	0,9177600247
ParsAgent	0,8940547361
myAgent	0,820587839

Agent	Score
RandomDance	0,8863513601
PokerFace	0,860369705
Atlas3	0,8594088244
Sengoku	0,8478529345
ParsAgent	0,8353034991
AresParty	0,8221559861
AgentH	0,8128058967
myAgent	0,8030690578
Mercury	0,7802531389

Japan trip Domain (DF = 0.9, preference profiles: 3)

Agent	Score
RandomDance	0,9055555556
Atlas3	0,9
ParsAgent	0,8972222222
myAgent	0,8808333333

Agent	Score
AresParty	0,9176587302
Sengoku	0,9046527778
Atlas3	0,9010218254
ParsAgent	0,9000396825
RandomDance	0,8991865079
PokerFace	0,8846130952
AgentH	0,8168353175
myAgent	0,8092361111
Mercury	0,8069246032

Domain Ace (DF = 0.2, preference profiles: 3)

Agent	Score
ParsAgent	0,8651675854
RandomDance	0,6436550832
myAgent	0,4267720609
Atlas3	0,4242564966

Agent	Score
ParsAgent	0,6731771195
PokerFace	0,5953625626
RandomDance	0,5753334255
Sengoku	0,5543892168
AresParty	0,4940560316
AgentH	0,4584816754
myAgent	0,4519384248
Atlas3	0,4143126867
Mercury	0,4077539465

KDomain (DF = 0.7, preference profiles: 3)

Agent	Score
ParsAgent	0,9138888889
RandomDance	0,8490740741
Atlas3	0,662037037
myAgent	0,4283333333

Agent	Score
ParsAgent	0,7453571429
AresParty	0,7202380952
PokerFace	0,7077480159
RandomDance	0,6896924603
Sengoku	0,6564583333
Atlas3	0,5867559524
AgentH	0,5587698413
Mercury	0,4924206349
myAgent	0,4525

Symposium Domain (DF = 1, preference profiles: 6)

Agent	Score
RandomDance	0,9101472836
ParsAgent	0,90480444
Atlas3	0,889371507
myAgent	0,7408318439

Agent	Score
ParsAgent	0,8123621671
RandomDance	0,8091450874
PokerFace	0,7992015826
Atlas3	0,7962354655
Sengoku	0,7913001963
AgentH	0,7584203585
Mercury	0,7253711414
AresParty	0,7177353896
myAgent	0,708446989

Electric Vehicle Domain (DF = 0.3, preference profiles: 3)

Agent	Score
ParsAgent	0,9240740741
RandomDance	0,8820987654
Atlas3	0,5907407407
myAgent	0,5856481481

Agent	Score
ParsAgent	0,7458994709
PokerFace	0,7091765873
RandomDance	0,6778935185
Sengoku	0,6700066138
Atlas3	0,6187996032
AgentH	0,6041501323
myAgent	0,6031911376
Mercury	0,5247850529
AresParty	0,5244047619

Bank Robbery Domain (DF = 1, preference profiles: 5)

Agent	Score
ParsAgent	0,8541494922
RandomDance	0,8500507954
Atlas3	0,8324096193
myAgent	0,694014931

Agent	Score
ParsAgent	0,7813201444
PokerFace	0,778682905
Atlas3	0,7708995005
RandomDance	0,7546924969
Sengoku	0,751448716
AresParty	0,7320061466
AgentH	0,7132144879
Mercury	0,7095305371
myAgent	0,6732679368

9.10 Conan Strategy with use of opponents' weights with Gahboninho opponent model

Party Domain (DF = 1, preference profiles: 4)

Agent	Score
RandomDance	0,8164269927
ParsAgent	0,7729518866
Atlas3	0,7257010907
myAgent	0,702184387

Agent	Score
RandomDance	0,7176946639
PokerFace	0,6579412635
Atlas3	0,6450865244
Sengoku	0,6429449122
ParsAgent	0,6281706005
myAgent	0,6027239755
AgentH	0,5942185888
AresParty	0,5842752944
Mercury	0,5492207953

Triangular Fight Domain (DF = 0.1, preference profiles: 3)

Agent	Score
ParsAgent	0,7849056604
RandomDance	0,6754716981
myAgent	0,600617284
Atlas3	0,5685534591

Agent	Score
ParsAgent	0,7483062331
PokerFace	0,7374338624
myAgent	0,7004676019
RandomDance	0,6872666667
Sengoku	0,6376
AresParty	0,633994709
AgentH	0,5974767596
Atlas3	0,5505678023
Mercury	0,5435333333

Smart Grid Domain (DF = 1, preference profiles: 3)

Agent	Score
ParsAgent	0,8683539117
Atlas3	0,7531016522
RandomDance	0,7229558691
myAgent	0,6647668353

Agent	Score
RandomDance	0,6624432761
PokerFace	0,6345123018
Atlas3	0,6287032858
ParsAgent	0,6234494994
Sengoku	0,6101093739
myAgent	0,5701608307
AgentH	0,5626072518
Mercury	0,5081659426
AresParty	0,4900793651

University Domain (DF = 1, preference profiles: 4)

Agent	Score
Atlas3	0,9215960364
RandomDance	0,9033257973
myAgent	0,8887175579
ParsAgent	0,8772557951

Agent	Score
RandomDance	0,8810747996
PokerFace	0,8662586034
myAgent	0,8582745589
Atlas3	0,854858779
Sengoku	0,8402024439
ParsAgent	0,8225296266
AresParty	0,8221559861
AgentH	0,8153816818
Mercury	0,786038442

Japan trip Domain (DF = 0.9, preference profiles: 3)

Agent	Score
ParsAgent	0,9
Atlas3	0,9
RandomDance	0,9
myAgent	0,9

Agent	Score
AresParty	0,9095238095
myAgent	0,9085615079
Sengoku	0,9085615079
Atlas3	0,9011111111
RandomDance	0,8970734127
ParsAgent	0,8916269841
PokerFace	0,8876388889
AgentH	0,8282539683
Mercury	0,8252876984

Domain Ace Domain (DF = 0.2, preference profiles: 3)

Agent	Score
ParsAgent	0,7418607912
Atlas3	0,5269124107
RandomDance	0,5192662122
myAgent	0,5192422267

Agent	Score
ParsAgent	0,6113121759
PokerFace	0,5444632042
myAgent	0,5427547356
RandomDance	0,5278263678
Sengoku	0,4983490477
AresParty	0,4567506
AgentH	0,453657107
Atlas3	0,4215736073
Mercury	0,4045956216

KDomain (DF = 0.7, preference profiles: 3)

Agent	Score
ParsAgent	0,8391666667
RandomDance	0,7326851852
myAgent	0,6589814815
Atlas3	0,5723148148

Agent	Score
ParsAgent	0,713452381
AresParty	0,6789285714
PokerFace	0,6573412698
RandomDance	0,6490277778
myAgent	0,6198710317
Sengoku	0,593452381
AgentH	0,5671428571
Atlas3	0,5526884921
Mercury	0,4574603175

Symposium Domain (DF = 1, preference profiles: 6)

Agent	Score
RandomDance	0,8849298038
ParsAgent	0,8748659497
Atlas3	0,8530151628
myAgent	0,8147768664

Agent	Score
RandomDance	0,7965203237
ParsAgent	0,7903277706
PokerFace	0,7795674337
Atlas3	0,7750214219
Sengoku	0,7699278148
AgentH	0,7565978893
myAgent	0,7554175232
Mercury	0,7116358541
AresParty	0,6813889735

Electric Vehicle Domain (DF = 0.3, preference profiles: 3)

Agent	Score
ParsAgent	0,924691358
RandomDance	0,8396604938
myAgent	0,7723765432
Atlas3	0,674691358

Agent	Score
ParsAgent	0,7431878307
PokerFace	0,712037037
myAgent	0,6787367725
RandomDance	0,670651455
Sengoku	0,6516699735
Atlas3	0,6264550265
AgentH	0,6080853175
Mercury	0,5313988095
AresParty	0,5244047619

Bank Robbery Domain (DF = 1, preference profiles: 5)

Agent	Score
ParsAgent	0,8527245518
RandomDance	0,8398732112
Atlas3	0,8157435509
myAgent	0,7998323168

Agent	Score
PokerFace	0,7701346896
ParsAgent	0,7693357046
Atlas3	0,7550129697
RandomDance	0,7451044572
myAgent	0,7440697935
Sengoku	0,7332593242
AgentH	0,7176254815
AresParty	0,7091720261
Mercury	0,7038126775

9.11 Conan Strategy with use of opponents' weights with Bayesian opponent model

Party Domain (DF = 1, preference profiles: 4)

Agent	Score
RandomDance	0,8169297303
ParsAgent	0,7772028161
Atlas3	0,7250266211
myAgent	0,7057140713

Agent	Score
RandomDance	0,7144141431
PokerFace	0,6553884973
Atlas3	0,6399810531
Sengoku	0,6382410645
ParsAgent	0,6250826936
myAgent	0,5964472225
AgentH	0,5893988445
AresParty	0,5768056672
Mercury	0,5470544786

Triangular Fight Domain (DF = 0.1, preference profiles: 3)

Agent	Score
ParsAgent	0,7639455782
RandomDance	0,6691823899
myAgent	0,6053333333
Atlas3	0,5537414966

Agent	Score
ParsAgent	0,7505175983
PokerFace	0,736005344
myAgent	0,6997306397
RandomDance	0,6777333333
Sengoku	0,637156271
AresParty	0,6323392975
AgentH	0,5946215139
Atlas3	0,5488529015
Mercury	0,5448666667

Smart Grid Domain (DF = 1, preference profiles: 3)

Agent	Score
ParsAgent	0,784050261
RandomDance	0,7489140578
Atlas3	0,6801377496
myAgent	0,6689785474

Agent	Score
RandomDance	0,6615860619
PokerFace	0,6321992126
ParsAgent	0,6299589816
Atlas3	0,6257472526
Sengoku	0,61715051
myAgent	0,5679811582
AgentH	0,56224009
Mercury	0,5161723289
AresParty	0,496031746

University Domain (DF = 1, preference profiles: 4)

Agent	Score
Atlas3	0,909759828
RandomDance	0,9053405951
myAgent	0,8536265343
ParsAgent	0,8335918756

Agent	Score
RandomDance	0,8799054427
PokerFace	0,8677778474
myAgent	0,8567841876
Atlas3	0,8548707693
Sengoku	0,8411318619
AresParty	0,8213041314
ParsAgent	0,8211993277
AgentH	0,8147333474
Mercury	0,7883677522

Japan trip Domain (DF = 0.9, preference profiles: 3)

Agent	Score
Atlas3	0,9
RandomDance	0,9
myAgent	0,9
ParsAgent	0,9

Agent	Score
AresParty	0,9095238095
myAgent	0,908452381
Sengoku	0,9018353175
Atlas3	0,9008829365
RandomDance	0,8983035714
ParsAgent	0,8915376984
PokerFace	0,884077381
AgentH	0,8373710317
Mercury	0,824047619

Domain Ace (DF = 0.2, preference profiles: 3)

Agent	Score
ParsAgent	0,7310375781
Atlas3	0,5245164589
RandomDance	0,5195623532
myAgent	0,5179711486

Agent	Score
ParsAgent	0,5962544248
myAgent	0,546308379
PokerFace	0,5428359921
RandomDance	0,5359447686
Sengoku	0,5003368117
AresParty	0,463276188
AgentH	0,4527217978
Atlas3	0,4230460134
Mercury	0,3996599944

KDomain (DF = 0.7, preference profiles: 3)

Agent	Score
ParsAgent	0,7730555556
RandomDance	0,7148148148
myAgent	0,6522222222
Atlas3	0,5718518519

Agent	Score
ParsAgent	0,700406746
AresParty	0,6672718254
PokerFace	0,6616468254
RandomDance	0,6385416667
myAgent	0,6203472222
Sengoku	0,6093353175
AgentH	0,5674305556
Atlas3	0,5373809524
Mercury	0,4676190476

Symposium Domain (DF = 1, preference profiles: 6)

Agent	Score
ParsAgent	0,8970217084
RandomDance	0,8825971262
myAgent	0,8765666498
Atlas3	0,8488733692

Agent	Score
RandomDance	0,7934984955
ParsAgent	0,788925511
PokerFace	0,7812250129
Atlas3	0,777200518
Sengoku	0,7700236033
AgentH	0,7570881186
myAgent	0,7543523578
Mercury	0,7115705164
AresParty	0,6806330527

Electric Vehicle Domain (DF = 0.3, preference profiles: 3)

Agent	Score	Agent	Score
ParsAgent	0,925308642	ParsAgent	0,748015873
RandomDance	0,8012345679	PokerFace	0,7156415344
myAgent	0,7802469136	myAgent	0,6806547619
Atlas3	0,6790123457	RandomDance	0,666468254
		Sengoku	0,6489252646
		Atlas3	0,6246031746
		AgentH	0,6119212963
		Mercury	0,5237103175
		AresParty	0,5166005291

Bank Robbery Domain (DF = 1, preference profiles: 5)

Agent	Score	Agent	Score
ParsAgent	0,8518381595	PokerFace	0,7714391103
RandomDance	0,8426589062	ParsAgent	0,7679016759
Atlas3	0,8174504119	Atlas3	0,7528420614
myAgent	0,7930075628	myAgent	0,744775878
		RandomDance	0,7434195982
		Sengoku	0,7338512153
		AgentH	0,7177705255
		AresParty	0,7069225428
		Mercury	0,703716277

9.12 Conan Strategy incorporated in Hybrid MGT – Meta Strategy with Gahboninho opponent model

Party Domain (DF = 1, preference profiles: 4)

Agent	Score	Agent	Score
myAgent	0,6490837124	RandomDance	0,6197527138
RandomDance	0,6477513857	Atlas3	0,5666520728
ParsAgent	0,5774708003	Sengoku	0,5663837504
Atlas3	0,5663029553	PokerFace	0,5646358232
		ParsAgent	0,512597649
		AgentH	0,5089491187
		myAgent	0,4675076579
		AresParty	0,4174254755

Triangular Fight Domain (DF = 0.1, preference profiles: 3)

Agent	Score	Agent	Score
ParsAgent	0,6646258503	PokerFace	0,6783042394
myAgent	0,6095238095	ParsAgent	0,6752920036
RandomDance	0,5737179487	myAgent	0,6495238095
Atlas3	0,568627451	RandomDance	0,6240788346
		Sengoku	0,5893782383
		AresParty	0,5802913453
		AgentH	0,5617414248
		Atlas3	0,5473090278
		Mercury	0,4311111111

Smart Grid Domain (DF = 1, preference profiles: 3)

Agent	Score
myAgent	0,6254867919
ParsAgent	0,5812119563
RandomDance	0,5633678478
Atlas3	0,5406390598

Agent	Score
RandomDance	0,5642763459
Sengoku	0,5358044347
Atlas3	0,5331870822
PokerFace	0,531187315
ParsAgent	0,4976805771
myAgent	0,4842000065
AgentH	0,4807635605
AresParty	0,3642533937
Mercury	0,3545870653

University Domain (DF = 1, preference profiles: 4)

Agent	Score
myAgent	0,9008396302
RandomDance	0,8663838422
Atlas3	0,8592080839
ParsAgent	0,8390012519

Agent	Score
RandomDance	0,8390778079
Atlas3	0,8106353443
PokerFace	0,8044902162
Sengoku	0,8000524829
AgentH	0,7778688723
ParsAgent	0,7693998058
myAgent	0,7481532327
Mercury	0,7089959241
AresParty	0,6623299823

Japan trip Domain (DF = 0.9, preference profiles: 3)

Agent	Score
myAgent	0,9
RandomDance	0,9
Atlas3	0,9
ParsAgent	0,9

Agent	Score
AresParty	0,9053648069
myAgent	0,904628821
Mercury	0,9017094017
Atlas3	0,9
Sengoku	0,9
RandomDance	0,8932954545
ParsAgent	0,8928366446
PokerFace	0,8802359551
AgentH	0,8618518519

Domain Ace Domain (DF = 0.2, preference profiles: 3)

Agent	Score
ParsAgent	0,5943744212
myAgent	0,5385625964
RandomDance	0,4654647477
Atlas3	0,4509158565

Agent	Score
ParsAgent	0,5214537861
myAgent	0,5060915312
PokerFace	0,4753994747
RandomDance	0,4699027287
Sengoku	0,4431684066
AresParty	0,4247502724
AgentH	0,4183884198
Atlas3	0,3931832019
Mercury	0,3512410965

KDomain (DF = 0.7, preference profiles: 3)

Agent	Score
myAgent	0,6201851852
ParsAgent	0,6180555556
RandomDance	0,5598148148
Atlas3	0,4781481481

Agent	Score
ParsAgent	0,5981719128
PokerFace	0,5719856459
myAgent	0,5651452785
RandomDance	0,5591460396
AresParty	0,5536579572
Sengoku	0,5270460358
AgentH	0,5152784504
Mercury	0,4982
Atlas3	0,477755611

Symposium Domain (DF = 1, preference profiles: 6)

Agent	Score
ParsAgent	0,7922735415
myAgent	0,7901858804
Atlas3	0,7702694761
RandomDance	0,7638694862

Agent	Score
RandomDance	0,7568954609
Atlas3	0,743849961
ParsAgent	0,7386777897
Sengoku	0,7377057116
PokerFace	0,7293802367
Mercury	0,7241275043
AgentH	0,7050392496
myAgent	0,704535411
AresParty	0,6229227761

Electric Vehicle Domain (DF = 0.3, preference profiles: 3)

Agent	Score
myAgent	0,8101851852
ParsAgent	0,7907407407
Atlas3	0,6638888889
RandomDance	0,6294753086

Agent	Score
ParsAgent	0,6988267771
myAgent	0,6898611111
PokerFace	0,6594744122
Sengoku	0,6229031092
RandomDance	0,6084201389
Atlas3	0,6075167038
AgentH	0,5996810207
Mercury	0,5077715356
AresParty	0,5051069703

Bank Robbery Domain (DF = 1, preference profiles: 5)

Agent	Score
myAgent	7,27037171
ParsAgent	7,15133288
RandomDance	6,73812085
Atlas3	6,53517918

Agent	Score
RandomDance	6,63503127
ParsAgent	6,62027907
PokerFace	6,60882742
myAgent	6,41085631
Sengoku	6,41077908
Atlas3	6,28526127
AgentH	6,10898103
AresParty	5,67620057
Mercury	5,45733305

9.13 Conan Strategy incorporated in Hybrid MGT – Meta Strategy with Bayesian opponent model

Party Domain (DF = 1, preference profiles: 4)

Agent	Score
myAgent	0,5757174224
RandomDance	0,5260612101
Atlas3	0,5077176775
ParsAgent	0,4643585155

Agent	Score
RandomDance	0,6144229742
Sengoku	0,5745474363
Atlas3	0,5723032349
PokerFace	0,5663815069
myAgent	0,5489612132
AgentH	0,5290950042
ParsAgent	0,5193479514
Mercury	0,4633157315
AresParty	0,4510926994

Triangular Fight Domain (DF = 0.1, preference profiles: 3)

Agent	Score
myAgent	0,5571428571
RandomDance	0,5091503268
ParsAgent	0,5027777778
Atlas3	0,5

Agent	Score
myAgent	0,6882591093
PokerFace	0,6821024617
ParsAgent	0,6736082474
RandomDance	0,628
Sengoku	0,6189134809
AresParty	0,5831341301
AgentH	0,5622754491
Atlas3	0,5407258065
Mercury	0,5309476474

Smart Grid Domain (DF = 1, preference profiles: 3)

Agent	Score
myAgent	0,5920176764
ParsAgent	0,4851116846
RandomDance	0,4771636522
Atlas3	0,4655135382

Agent	Score
RandomDance	0,5698643601
Atlas3	0,5593248591
Sengoku	0,5446017624
PokerFace	0,5387993297
myAgent	0,536899645
ParsAgent	0,5040428033
AgentH	0,4977350287
Mercury	0,4374762004
AresParty	0,3849206349

University Domain (DF = 1, preference profiles: 4)

Agent	Score
myAgent	0,8462391845
Atlas3	0,8068397287
RandomDance	0,7998742142
ParsAgent	0,7608607958

Agent	Score
RandomDance	0,8242089638
PokerFace	0,8026436621
Atlas3	0,7957060733
Sengoku	0,7864613561
AgentH	0,7837969404
ParsAgent	0,7632626938
Mercury	0,7176115291
myAgent	0,7142058392
AresParty	0,6439920556

Japan trip Domain (DF = 0.9, preference profiles: 3)

Agent	Score	Agent	Score
myAgent	0,9	myAgent	0,9122103175
Atlas3	0,9	AresParty	0,9103174603
RandomDance	0,9	Sengoku	0,9016071429
ParsAgent	0,9	Atlas3	0,9011706349
		RandomDance	0,8975099206
		ParsAgent	0,8909920635
		PokerFace	0,8826587302
		AgentH	0,8318353175
		Mercury	0,8230059524

Domain Ace Domain (DF = 0.2, preference profiles: 3)

Agent	Score	Agent	Score
myAgent	0,5326138022	myAgent	0,5576020684
ParsAgent	0,4698972941	ParsAgent	0,5347807653
RandomDance	0,4116440069	RandomDance	0,4943403864
Atlas3	0,3841982096	PokerFace	0,4797959043
		Sengoku	0,4710643664
		AresParty	0,4459878848
		AgentH	0,4143191618
		Atlas3	0,3975950008
		Mercury	0,3846937156

KDomain (DF = 0.7, preference profiles: 3)

Agent	Score	Agent	Score
myAgent	0,5703703704	ParsAgent	0,609672619
ParsAgent	0,5088888889	PokerFace	0,5899900794
Atlas3	0,4655555556	RandomDance	0,5779960317
RandomDance	0,4587962963	AresParty	0,5728968254
		Sengoku	0,5674007937
		myAgent	0,5605456349
		AgentH	0,5216468254
		Atlas3	0,5200496032
		Mercury	0,425327381

Symposium Domain (DF = 1, preference profiles: 6)

Agent	Score	Agent	Score
myAgent	0,7504890075	RandomDance	0,7372783338
Atlas3	0,7219829166	Atlas3	0,7364021018
ParsAgent	0,7003345626	Sengoku	0,7314700651
RandomDance	0,6980644858	ParsAgent	0,7312491403
		PokerFace	0,7280454419
		AgentH	0,7109031365
		myAgent	0,7105894766
		Mercury	0,6730916647
		AresParty	0,6253169693

Electric Vehicle Domain (DF = 0.3, preference profiles: 3)

Agent	Score	Agent	Score
myAgent	0,7034391534	myAgent	0,8759259259
ParsAgent	0,7009259259	ParsAgent	0,8
PokerFace	0,6611937831	RandomDance	0,650154321
Sengoku	0,6404265873	Atlas3	0,6311728395
Atlas3	0,6175429894		
RandomDance	0,6150462963		
AgentH	0,5881448413		
Mercury	0,5321263228		
AresParty	0,5223544974		

Bank Robbery Domain (DF = 1, preference profiles: 5)

Agent	Score	Agent	Score
ParsAgent	0,7073028478	myAgent	0,7543151422
PokerFace	0,7053346828	ParsAgent	0,6870544916
myAgent	0,702565645	Atlas3	0,6698908637
Atlas3	0,7021114371	RandomDance	0,6612771817
Sengoku	0,6897965652		
RandomDance	0,6869049939		
AgentH	0,6732122685		
Mercury	0,6539559181		
AresParty	0,6374428885		