Theoretical Properties of the MiCRO Negotiation Strategy

Dave de Jonge[1*]

[1*]IIIA, CSIC, Campus de la UAB, Bellaterra, 08193, Spain.

Corresponding author(s). E-mail(s): davedejonge@iiia.csic.es;

Abstract

Recently, we have introduced a new algorithm for automated negotiation, called MiCRO, which, despite its simplicity, outperforms many state-of-the-art negotiation strategies [1]. Furthermore, we claimed that under certain conditions which typically hold in the Automated Negotiating Agents Competition (ANAC), it is a game-theoretically optimal strategy. The goal of this paper is to formally prove those claims. Specifically, we define 'negotiation' as an extensive-form game and define the class of *consistent* strategies for this game, which consists of those strategies that satisfy a number of rationality criteria. We then prove that under the above mentioned conditions MiCRO is a best response against itself among all consistent negotiation strategies. Furthermore, we define the notion of a *balanced* negotiation domain, which is a domain in which two MiCRO agents would always come to an optimal agreement. Finally, we show that many of the domains used in ANAC indeed happen to be (approximately) balanced. The importance of this work is that if we know under which conditions MiCRO is theoretically optimal, then we can use this to test to what extent other negotiation algorithms are able to achieve similar results to MiCRO when applied under those same conditions. Furthermore, it would help researchers to design more challenging test cases for automated negotiation in which MiCRO is not optimal.

Keywords: Automated Negotiation, Multi-Agent Systems, The Bargaining Problem, Game Theory

1 Introduction

As the use of intelligent agents is becoming increasingly widespread, and those agents are becoming increasingly autonomous, it is to be expected that in the near future it will be common for multiple such agents to be deployed to work together, while representing different stakeholders with different objectives. This would require negotiation between such agents and we therefore expect that automated negotiation will become a more and more prevalent technology in our daily lives.

The field of automated negotiation deals with systems of autonomous agents that each have different objectives, but that need to cooperate to ensure beneficial outcomes. When presented with a problem, such agents propose potential solutions to one another, in the hope to find one that is acceptable to everyone. This means that, even though the agents are purely self-interested, they each still need to make sure that their proposals are also sufficiently beneficial to the other agents, otherwise none of those proposals would ever get accepted [2]. A simple example is the scenario of a buyer and a seller that are negotiating the price of a car. While the seller prefers to sell the car for the highest possible price, she has to keep in mind that the price must still be low enough for the buyer to accept the offer.

The main question in automated negotiation is how to make this trade-off, and is commonly known as *the bargaining problem*. In the case that the agents' utility functions are common knowledge, this problem has been solved theoretically [3–5]. However, the question how to solve it in the case that the agents do not know each others' utility functions is a long-standing open problem [6]. While we do not claim to have solved it in general, we do argue in this paper that our recently introduced algorithm called MiCRO [1] solves it under a number of specific conditions.

In the literature, many different negotiation strategies have been proposed that aimed to tackle the bargaining problem heuristically. Most of them start by making very selfish proposals and, as time passes, concede by making proposals that yield more and more utility to the opponent. While the agents are not given any knowledge about their opponents' utility functions, they typically try to infer this information at runtime, from the proposals they receive from their opponents. This is known as *opponent modeling*. To test such strategies, the annual Automated Negotiating Agent Competition (ANAC) has been organized since 2010. Most agents that were successful in this competition used various different machine learning techniques for opponent modeling. However, we have recently shown that many of the ANAC negotiation scenarios can in fact be tackled just as well without opponent modeling [1]. To demonstrate this, we presented a new negotiation strategy, called MiCRO, which does not employ any kind of opponent modeling or machine learning at all, and yet outperformed many of the strongest participants of ANAC. Furthermore, we claimed that, under certain conditions, it is game-theoretically optimal.

The main goal of this paper is to formally prove our claim that MiCRO is game-theoretically optimal on many of the ANAC domains. Specifically, we prove this in four steps:

1. We define the notion of a *consistent* negotiation strategy, which is a negotiation strategy that satisfies a number of rationality conditions, and prove that MiCRO is consistent (Section 5).
2. We define the notion of the *balance set*, which is the set of possible agreements that two MiCRO agents could make with each other (Section 6).

3. We then formally prove that (under a number of mild assumptions that typically hold in ANAC) MiCRO is a best response against itself among all possible consistent negotiation strategies (Section 7).
4. We define the notion of a *balanced* negotiation domain as a domain where the balance set is a subset of the set of optimal agreements (meaning that two MiCRO agents would make optimal agreements in such a domain) and we show that most of the domains used in ANAC are indeed (approximately) balanced (Section 8.2).

In addition, we also make the following contributions:

- We investigate why so many ANAC domains are balanced, and show that this is related to a certain kind of symmetry in those domains (Section 8.3).
- We propose a number of improvements to MiCRO that help defend it against inconsistent opponents (Section 9).
- We identify a number of research questions about opponent modeling that may be answered with the help of MiCRO (Section 10).

We feel, however, that we should stress the following.

Remark 1. *MiCRO is not intended to be used as a real-world negotiation algorithm. It is purely intended as a theoretical device that allows us to analyze the complexity of negotiation domains, and as a benchmark algorithm that allows us to assess the strength of other negotiation algorithms.*

We argue that our work is important for the following reasons:

1. The nice theoretical properties of MiCRO in combination with its simplicity make it an ideal benchmark strategy. After all, it does not make much sense to use a highly sophisticated machine learning algorithm if it does not even outperform a much simpler strategy like MiCRO.
2. The fact that we have identified a set of conditions under which MiCRO is theoretically optimal is useful because it allows us to test for any other negotiation strategy to what extent it is able to achieve similar results under those conditions. In other words, it provides us with an objective 'optimality test' for future negotiation algorithms. In Section 10 we will present a number of concrete research questions that one may ask for any new negotiation algorithm, and that may be answered thanks to our work.
3. Knowing under which conditions MiCRO is optimal will be useful for researchers to design more challenging negotiation test cases for which there is no known optimal algorithm.

Finally, we should remark that our original paper [1] contained a number of small errors, so some of the claims we made in that paper were not formulated correctly.[1] In this paper we fix those errors. Specifically, we have made a small change in the definition of an 'inconsistent proposal', and we have made a small change to the definition of MiCRO itself.

The source code of MiCRO, as well as our code to calculate the balance set and the balance values of a given domain, have been made publicly available at:
https://www.iiia.csic.es/~davedejonge/downloads

[1] We thank an anonymous reviewer for pointing this out.

2 Related Work

The first edition of ANAC was held in 2010, and has since then served as the reference for most research on automated negotiation. The negotiation domains that are used in most editions of this competition still form the most commonly used benchmark data set in the field of automated negotiation.

The first three editions of ANAC focused on basic bilateral negotiations with linear utility functions [7], and in 2013 the option was added for agents to learn from previous negotiation sessions [8]. In 2014 the focus shifted to very large domains with non-linear utility functions [9]. However, from 2015 onward the competition went back to smaller domains and linear utility, but focused on multilateral negotiations [10]. In 2019 and 2020 the focus shifted back to bilateral negotiations, but this time the agents only had *partial* knowledge about their own utility functions [11]. Since 2021 the agents' preferences have again been represented by ordinary linear utility functions, and, like in 2013, the option to learn from previous negotiation sessions was re-introduced. Furthermore, since 2017 ANAC has been extended with a number of separate leagues focused on more specific challenges, such as the game of Diplomacy [12], supply chain environments [11], the game of Werewolves [11], and negotiations between agents and humans [13].

Despite the increased attention to more complex scenarios, most work on automated negotiation still seems to be focused on small bilateral scenarios with linear utility functions. For example, the main leagues of ANAC 2019–2023 were all exclusively based on such scenarios. Furthermore, the domains of ANAC 2012–2013 are still widely considered to be the default benchmark for automated negotiation, and have been used in many recent high-level publications. For example, Sengupta et al. [14] used the ANAC 2013 domains to test their work, while Mirzayi et al. [15] used a small selection of domains from both ANAC 2012–2013, and Bakker et al. [16] used similar linear ANAC-style domains.

Even for the basic settings of ANAC 2010–2013, a plethora of different opponent modeling techniques have been applied by the participants. For example, to predict the opponent's concession strategy, Agent K [17], the winner of ANAC 2010, used an extrapolation algorithm based on the average and standard deviation of the utility the opponent has offered so far. Other participants applied more sophisticated machine learning techniques, such as non-linear regression (IAMhaggler [18]), Gaussian processes (IAMHaggler2011 [19]) or wavelet decomposition and cubic smoothing splines (OMAC [20]). To learn the opponent's utility function, a commonly used technique is Bayesian learning, which was first proposed for automated negotiation in [21] and later applied by many participants of ANAC, such as FSEGA [22], IAMhaggler [18], and Nice Tit-For-Tat [23]. Several other agents used reinforcement learning for this goal, such as HardHeaded [24] and ValueModelAgent [25]. A large survey of opponent modeling techniques for negotiation can be found in [26].

While the above mentioned agents all rely on heuristic methods, there is also a large body of work on theoretically optimal solutions for automated negotiation. However, these solutions typically cannot be applied to the ANAC competitions, because they are based on certain assumptions that do not hold in ANAC. Especially the assumption that both agents' know each others' utility functions has been studied extensively.

The most well-known example is the Nash bargaining solution [3]. Nash proved that if both negotiators have full knowledge of each others' utility functions, and the agreement space forms a convex set, then (under a number of additional assumptions) the outcome of the negotiation would be the agreement that maximizes the product of the agents' utilities. Rubinstein later showed that if the negotiation takes place over discrete rounds and the utility functions are time-discounted then the optimal *strategy* indeed leads to Nash's solution [4]. Much more recently, however, we showed that if the agreement space is discrete, rather than convex, and we assume that the negotiating agent will adopt each 'side' of the negotiation (e.g. 'buyer' and 'seller') equally often, then the optimal agreement is in fact the one that maximizes the *sum* of the utility functions, rather than the product [5].

Much less is known about optimal negotiation strategies for the case that the opponent's utility is unknown, although several optimal strategies for this scenario have been proposed as well [27, 28]. However, they do need to assume their agent has an accurate model of the probability that the opponent will accept any given proposal, and it remains unclear to what extent such models can realistically be obtained, even in a controlled environment such as ANAC. Furthermore, they only prove their algorithm is a best response against a specific type of opponent, but it remains unclear whether it is also a best response against itself, and therefore whether or not it can form a Nash equilibrium. Later, some improvements to these strategies were proposed in [29], but they do not resolve the above mentioned limitations. An optimal strategy for the acceptance of proposals was presented in [30], but their proof of optimality depends on the assumption that the agent cannot make any proposals itself.

To the best of our knowledge, MiCRO is the only known strategy that has provably optimal properties under the conditions of ANAC.

3 Definitions

In this section we introduce the formal definitions and notation that are required to understand the rest of this paper. Please note that we are not claiming that anything presented in this section is novel.

3.1 Negotiation Domains

In a classical scenario for automated negotiation, two agents α_1 and α_2 are bargaining to agree on the details of a contract. The agents have a fixed amount of time to make proposals to one another, according to some given negotiation protocol. That is, each agent may propose an offer ω, from some given set of possible offers Ω, to the other agent which may then either accept the proposal or reject it and make a counter proposal $\omega' \in \Omega$. The agents continue making proposals to each other until either the deadline has passed, or one of the agents accepts a proposal made by the other. Each agent α_i has a private utility function u_i that assigns to each offer $\omega \in \Omega$ a utility value $u_i(\omega) \in \mathbb{R}$, but which is not known to the other agent. When an offer ω gets accepted the agents receive their respective utility values $u_1(\omega)$ and $u_2(\omega)$ corresponding to this offer. On the other hand, if the negotiations fail because no proposal is accepted before the deadline, then each agent α_i receives a fixed utility value $rv_i \in \mathbb{R}$, which is known as its *reservation value.*

Definition 1. *A bilateral* ***negotiation domain*** *D is a tuple $\langle \Omega, u_1, u_2, rv_1, rv_2 \rangle$ where:*

- *Ω is a finite set known as the* ***offer space****, representing the possible offers the agents can propose to one another.*
- *u_1 and u_2 are two* ***utility functions*** *$u_i : \Omega \to \mathbb{R}$ (one for each agent) which assign to each offer $\omega \in \Omega$ a utility value $u_i(\omega) \in \mathbb{R}$.*
- *rv_1, $rv_2 \in \mathbb{R}$ are the* ***reservation values*** *of the respective agents.*

In the example of a buyer and a seller that bargain over the price of a car, the set of possible offers Ω would be the set of prices that one can reasonably expect them to offer or ask.

In the literature many different kinds of negotiation domains have been studied. For example, one can distinguish between so-called 'single-issue' domains, 'multi-issue' domains, or 'strategic' domains [31]. Furthermore, one can distinguish between linear utility functions and non-linear utility functions. However, in this work these distinctions are not relevant. Our work applies to any of these types of domains, as long as the offer space is not too large and for any given offer ω the agent can calculate its own utility value $u_i(\omega)$ efficiently.

Remark 2. *In this paper we do not make any assumptions about the type of utility functions (linear or non-linear), and we do not make any assumptions about the structure of the offer space (single-issue or multi-issue), other than that it is a finite set.*

Furthermore, it is sometimes assumed in the literature that the utility obtained by the agents also depends on time. That is, when the agents agree on some offer ω, the utility received by the agents is given by $\delta^t \cdot u_i(\omega)$ where δ is a real number between 0 and 1, called the *discount factor* and where t is the time at which they make the deal. This means that the later the deal is made, the less utility they receive. In this paper we do not take into account such discount factors. However, we will briefly argue in Section 8.4 that this does not matter much because the presence of discount factors only yields even more benefit to MiCRO.

3.2 Negotiation as an Extensive-Form Game

In this section we will formally define automated negotiation as an extensive-form game. To this end we will define the actions of this game (Def. 2), the states of this game (Def. 3), for each state, the set of actions that are legal in that state (Def. 4), and the state-transition function that determines how the actions of the players change the state of the game (Def. 6). One should keep in mind, however, that in this game the players do not know each other's utility functions.

We will always silently assume there is a fixed set of offers Ω.

Definition 2. *We define a* ***negotiation action*** *to be a tuple*

$$(i, \eta, \omega, t) \quad \in \quad \{1,2\} \times \{\mathfrak{p}, \mathfrak{a}\} \times \Omega \times \mathbb{R}$$

where i represents the index of the agent performing the action, and η represents the ***type*** *of the action, which can be either 'proposal', represented by the symbol $\mathfrak{p}$, or 'acceptance', represented by the symbol $\mathfrak{a}$. Furthermore, ω is the offer that is being proposed or accepted, and t is the time at which the agent proposes or accepts the offer.*

So, the notation $(1, \mathfrak{p}, \omega, t)$ means that agent α_1 proposes offer ω at time t, and the notation $(2, \mathfrak{a}, \omega, t)$ means that agent α_2 accepts offer ω at time t.

We use $\mathcal{A}$ to denote the set of all possible negotiation actions. Furthermore, we use $\mathcal{A}_1$ to denote the set of all negotiation actions with $i = 1$, and $\mathcal{A}_2$ for the set of all negotiation actions with $i = 2$. That is:

$$\begin{aligned} \mathcal{A}_1 &:= \{1\} \times \{\mathfrak{p}, \mathfrak{a}\} \times \Omega \times \mathbb{R} \\ \mathcal{A}_2 &:= \{2\} \times \{\mathfrak{p}, \mathfrak{a}\} \times \Omega \times \mathbb{R} \\ \mathcal{A} &:= \mathcal{A}_1 \cup \mathcal{A}_2 \end{aligned}$$

If $a = (i, \eta, \omega, t)$ and $a' = (i', \eta', \omega', t')$ are two negotiation actions with $t < t'$, then we say that a **comes before** a', or that a' **comes after** a. We may denote this as $a \lhd a'$.

$$(i, \eta, \omega, t) \lhd (i', \eta', \omega', t') \quad \overset{\text{def}}{\Longleftrightarrow} \quad t < t'$$

Definition 3. *We define a* ***negotiation state*** *s to be a sequence of negotiation actions:*

$$s = (a_1, a_2, \ldots, a_k)$$

sorted in chronological order (i.e. $a_j \lhd a_{j+1}$ for all $j \in \{1, 2, \ldots, k-1\}$). The integer k can be any arbitrary non-negative integer, including zero. The set of all possible negotiation states is denoted as $\mathcal{S}$.

We may write (with slight abuse of notation) $a \in s$ to indicate that a is an element of the sequence s, and $a \notin s$ to indicate it is not in the sequence. Furthermore, whenever the letter i is used as the index of an agent α_i, we may use the notation $3 - i$ to refer to the index of the opponent. After all, if $i = 1$ then $3 - i = 2$, and if $i = 2$ then $3 - i = 1$. So, α_{3-i} is indeed the opponent of α_i.

A **negotiation protocol** Π specifies, for any state s whether the negotiations have finished or not and which agreements have been made, and for any state s and agent α_i which offers that agent may propose or accept. That is, it is a pair of maps $\Pi = (\Pi_{stat}, \Pi_{ac})$ where Π_{stat} represents the status of the negotiations and Π_{ac} represents the agents' legal actions.

Specifically, $\Pi_{stat} : \mathcal{S} \to \Omega \cup \{\mathfrak{o}, \mathfrak{f}\}$, where $\mathfrak{o}$ and $\mathfrak{f}$ are two abstract symbols, with the following meaning:

- $\Pi_{stat}(s) = \mathfrak{o}$ means that in state s the negotiations are ongoing.
- $\Pi_{stat}(s) = \mathfrak{f}$ means that in state s the negotiations have failed, i.e. finished without agreement.
- $\Pi_{stat}(s) = \omega \in \Omega$, means that in state s the negotiations have finished with agreement ω.

Furthermore, $\Pi_{ac} : \mathcal{S} \times \{1, 2\} \to 2^{\mathcal{A}}$ determines the legal actions for the agents. That is for any state s and any agent α_i, the legal actions for that agent in that state are given by $\Pi_{ac}(s, i)$.

Arguably the most commonly used negotiation protocol in the literature is the alternating offers protocol (AOP) [32]. In the AOP, the agents take turns alternately. In each turn, the agent whose turn it is may propose any offer ω from the offer space Ω, or may accept the previous offer proposed by the opponent. Negotiations finish either when the deadline passes, or as soon as one of the agents accepts a proposal. This protocol has three parameters: the offer space Ω, the deadline $T \in \mathbb{R}$, and the index $i \in \{1, 2\}$ of the agent that is the first to make a proposal.

Definition 4. *Let Ω be some set of offers, let $s = (a_1, a_2, \ldots, a_k)$ be any negotiation state of length k, and, in case $k \neq 0$, let $a_k = (i_k, \eta_k, \omega_k, t_k)$. Then, the* ***alternating offers protocol*** *with deadline $T \in \mathbb{R}$ and initial agent $i \in \{1, 2\}$, is defined as follows:*

$$\begin{array}{ll}
\textit{If } k = 0, \textit{ then:} & \Pi_{stat} = \mathfrak{o} \\
& \Pi_{ac}(s, i) = \{i\} \times \{\mathfrak{p}\} \times \Omega \times (0, \infty) \\
& \Pi_{ac}(s, 3 - i) = \emptyset \\
\\
\textit{else, if } t_k > T, \textit{ then:} & \Pi_{stat}(s) = \mathfrak{f} \\
& \Pi_{ac}(s, 1) = \Pi_{ac}(s, 2) = \emptyset \\
\\
\textit{else, if } \eta_k = \mathfrak{a}, \textit{ then:} & \Pi_{stat}(s) = \omega_k \\
& \Pi_{ac}(s, 1) = \Pi_{ac}(s, 2) = \emptyset \\
\\
\textit{else:} & \Pi_{stat} = \mathfrak{o} \\
& \Pi_{ac}(s, \hat{k}) = \big(\ \{\hat{k}\} \times \{\mathfrak{p}\} \times \Omega \times (t_k, \infty)\ \big) \quad \cup \\
& \qquad\qquad\quad \big(\ \{\hat{k}\} \times \{\mathfrak{a}\} \times \{\omega_k\} \times (t_k, \infty)\ \big) \\
& \Pi_{ac}(s, 3 - \hat{k}) = \emptyset
\end{array}$$

where:

$$\hat{k} := \begin{cases} 1 & \textit{if } k + i \textit{ is odd} \\ 2 & \textit{if } k + i \textit{ is even} \end{cases}$$

The first line says that in the initial state, agent α_i can propose any offer from Ω. The second line says that once any action has been made after the deadline, the negotiations finish with failure. The third line says that if an agent accepts a proposal before the deadline, then the negotiations finish, with the accepted proposal as the outcome. The fourth line says that, in all other cases, the next agent $\alpha_{\hat{k}}$ may propose any offer from the set Ω, or may accept the previous proposal.

Strictly speaking, this definition says that, unless a proposal is accepted before the deadline, the negotiations will only have finished once either of the agents has taken an action *after* the deadline. However, since any action taken after the deadline does not have any influence on the outcome anyway (since we will always have $\Pi_{stat}(s) = \mathfrak{f}$), we can in practice consider the negotiations to be finished immediately after the deadline has passed. The only reason we formally require some agent to take an action after the deadline, is to simplify the formalization.

In the literature, it is typically also assumed that agents are allowed to end the negotiations without agreement before the deadline. In the above formalization this is also possible, by choosing an action $(\hat{k}, \eta, \omega, t)$ with $t > T$, which means the agent chooses to wait until the deadline has passed before making its next action.

Definition 5. *Let Π be the alternating offers protocol. Then, a* ***negotiation strategy*** *σ for agent α_i is a partial function $\sigma : \mathcal{S} \rightarrow \mathcal{A}_i$, such that:*

- *if $\Pi_{ac}(s, i) \neq \emptyset$ then $\sigma(s) \in \Pi_{ac}(s, i)$*
- *if $\Pi_{ac}(s, i) = \emptyset$ then $\sigma(s)$ is undefined.*

In other words, the map σ is only defined for those states s in which it is α_i's 'turn', according to the AOP.

If $\sigma(s) = (i, \mathfrak{p}, \omega, t)$ it means that if the negotiation is in state s then the agent applying strategy σ waits until time t and then proposes offer ω. Similarly, if $\sigma(s) = (i, \mathfrak{a}, \omega, t)$ it means the agent waits until time t and then accepts offer ω. The condition $\sigma(s) \in \Pi_{ac}(s, i)$ simply means that the strategy only selects actions that the protocol allows it to select.

For any extensive-form game, its transition function τ defines, for a given state s and a given action a the next state $\tau(s, a)$. We will now define the transition function for our model of negotiation.

Definition 6. *The* ***transition function*** *τ is a map $\mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$, defined as follows. Let $s = (a_1, a_2, \ldots, a_k)$ be a negotiation state, and let $a = (i, \eta, \omega, t)$ be a negotiation action, then the next state $s' = \tau(s, a) \in \mathcal{S}$ is defined as:*

$$\tau(s, a) := (a_1, a_2, \ldots, a_k, a_{k+1})$$

where $a_{k+1} = (i, \eta, \omega, t + \epsilon_{k+1})$ and where ϵ_{k+1} is a strictly positive real number, drawn randomly from some probability distribution.

In other words, whenever an agent performs an action a, it will cause the original state s to be concatenated with a new action a_{k+1}. However, it is important to note that this action is *not* exactly the action a performed by the agent. Instead, if the agent performs action (i, η, ω, t), then the state is updated with action $(i, \eta, \omega, t + \epsilon_{k+1})$. This represents the fact that when the agent sends a propose- or accept- message at time t, it will take a small amount of time ϵ_{k+1} for that message to arrive, due to network latency. Since in general we do not know exactly how much time this takes, we model ϵ_{k+1} as a random variable. The probability distribution function of this random variable is not relevant for this paper. Note that this makes negotiation an *non-deterministic* extensive-form game. To be precise, we may assume that both agents are connected to a 'negotiation manager' over a network. Each time an agent sends a proposal or acceptance, this message has to go to the negotiation manager, and the state of the negotiation is determined by the time at which the negotiation manager receives the message.

Definition 7. *Let Π be the alternating offers protocol, and D be a negotiation domain, then the extensive-form game Γ_D is defined as follows:*

- *The players are the two negotiating agents α_1 and α_2.*
- *The players' sets of possible actions are given by $\mathcal{A}_1$ and $\mathcal{A}_2$ as in Def. 2.*
- *The set of states of the game are given by $\mathcal{S}$ as in Def. 3.*
- *The initial state of the game is the empty sequence $() \in \mathcal{S}$.*

- *For any given state s and agent α_i, the set of legal actions are given by $\Pi_{ac}(s, i)$ as in Def. 4.*
- *The state transition function is defined as τ in Def. 6.*
- *The set of terminal states is defined as the set of all states s for which $\Pi_{stat}(s) \neq \mathfrak{o}$*
- *The players' utility functions U_i over the terminal states are given by:*

$$U_i(s) = \begin{cases} u_i(\omega) & \textit{if } \Pi_{stat}(s) = \omega \in \Omega \\ rv_i & \textit{if } \Pi_{stat}(s) = \mathfrak{f} \end{cases}$$

where u_i and rv_i are the utility function and reservation value of the negotiation domain for each agent α_i.

Problem 1. ***(The Bargaining Problem)*** *Let D be any negotiation domain and Γ_D the corresponding extensive-form game as defined in Def. 7. What is the optimal strategy for the game Γ_D?*

3.3 More Notation

We will now introduce some more notation that we will use to state and prove our lemmas and theorems.

Recall that, if $a = (i, \eta, \omega, t)$ and $a' = (i', \eta', \omega', t')$, then $a \lhd a'$ means that $t < t'$. Similarly, we may use the notation $a \lhd t'$ to denote that the time t at which action a takes place is before t', or $t \lhd a'$ to denote that action a' takes place after time t.

For any state $s = (a_1, a_2, \ldots, a_k)$ we use $s_{<t}$ to refer to the subsequence of s consisting of all actions $a \in s$ that take place before t and we use $s_{>t}$ to denote the subsequence consisting of all actions $a \in s$ that take place after t. For example, if we have:

$$s = ((1, \mathfrak{p}, \omega, 0.1), (2, \mathfrak{p}, \omega', 0.3), (1, \mathfrak{p}, \omega'', 0.5), (2, \mathfrak{p}, \omega''', 0.7), (1, \mathfrak{a}, \omega''', 0.9))$$

then:

$$\begin{aligned} s_{<0.6} &= ((1, \mathfrak{p}, \omega, 0.1), (2, \mathfrak{p}, \omega', 0.3), (1, \mathfrak{p}, \omega'', 0.5)) \\ s_{>0.6} &= ((2, \mathfrak{p}, \omega''', 0.7), (1, \mathfrak{a}, \omega''', 0.9)). \end{aligned}$$

Formally, if $s = (a_1, a_2, \ldots, a_k)$, then:

$$s_{<t} := \begin{cases} () & \text{if } \ t \lhd a_1 \\ (a_1, a_2, \ldots, a_j) & \text{if } \ a_j \lhd t \quad \wedge \quad t \lhd a_{j+1} \\ s & \text{if } \ a_k \lhd t \end{cases}$$

$$s_{>t} := \begin{cases} () & \text{if } \ a_k \lhd t \\ (a_{j+1}, a_{j+2}, \ldots, a_k) & \text{if } \ a_j \lhd t \quad \wedge \quad t \lhd a_{j+1} \\ s & \text{if } \ t \lhd a_1 \end{cases}$$

Definition 8. *Let $s = (a_1, a_2, \dots, a_k)$ be any state. Then we say that an action a is a **reply** to another action a', denoted $a = reply_s(a')$, if a follows directly after a'. That is, for any $j \in \{1, 2, \dots, k-1\}$:*

$$reply_s(a_j) := a_{j+1}$$

Definition 9. *Let s be any given state. Assuming negotiations follow the AOP, we say that α_i **rejects** an offer ω at time t, denoted $rej_s(i, \omega, t)$, if its opponent α_{3-i} has proposed ω and α_i replies to it with a new proposal, rather than accepting it.*

That is, the predicate $rej_s(i, \omega, t)$ holds iff there exist $\omega' \in \Omega$ and $t' \in \mathbb{R}$ such that the following two conditions are both satisfied:

- $(3 - i, \mathfrak{p}, \omega, t') \in s_{<t}$
- $reply_s((3 - i, \mathfrak{p}, \omega, t')) = (i, \mathfrak{p}, \omega', t)$

3.4 The MiCRO Strategy

We now recall the MiCRO negotiation strategy, which we introduced in [1]. See Algorithm 1 for its implementation in pseudo-code.

Simply stated, MiCRO works as follows: whenever the opponent proposes a *new* offer, MiCRO also replies with a new offer. This offer is always the agent's offer with highest utility for itself that the agent has not yet proposed before. On the other hand, when the opponent repeats an offer it has already proposed before, then MiCRO also replies with an offer it has already proposed before.

More formally, let α_1 denote an agent that applies the MiCRO strategy, and α_2 its opponent (which may be applying any arbitrary strategy), and let $K := |\Omega|$ denote the size of the domain. Before the negotiations begin, our agent α_1 creates a list $(\omega_1, \omega_2, \dots, \omega_K)$ containing all offers in the domain, sorted in order of decreasing utility for itself. That is, $u_1(\omega_1) \geq u_1(\omega_2) \geq \dots \geq u_1(\omega_K)$. Then, whenever it is α_1's turn to make a proposal, it counts how many *different* offers it has so far received from the opponent (we denote this number by n), and how many *different* offers it has so far proposed to the opponent (we denote this number by m)[2]. If $m \leq n$ then MiCRO will propose ω_{m+1}. On the other hand, if $m > n$ then it picks a random integer r such that $1 \leq r \leq m$ and proposes ω_r. Of course, it should never propose any offer that is below its reservation value, so in case $u_1(\omega_{m+1}) < rv_1$, it also just repeats a random previous proposal, even if $m \leq n$ (see Algorithm 1, lines 9–12 and 23).

The intuition behind MiCRO, is that it is a kind of Tit-for-Tat strategy [23] that does not use any knowledge about the opponent's utility function. That is, MiCRO tries to make the same number of concessions as the opponent, but it does not care *how large* the opponent's concessions are. After all, since the opponent's utility is unknown, the size of the opponent's concession as perceived by MiCRO says nothing about the size of the concession the opponent *intended* to make. The opponent might make a large concession in terms of its own utility, but this may result in a very small concession measured in our agent's utility. For the same reason MiCRO never makes large concessions to its opponent. In fact, it always makes exactly the smallest possible

[2] Note that if the agents follow the AOP and one of them applies the MiCRO strategy, then we always have $|m - n| \leq 1$.

concession: it just proposes the next offer from its list. Another difference between MiCRO and classic TFT is that MiCRO uses a different definition of ‘concession’. That is, even if the opponent’s new proposal offers less utility to MiCRO than the opponent’s previous proposal, MiCRO still considers this a concession, as long as it is different from any of the opponent’s previous offers. After all, if the opponent makes offers in order of decreasing utility for itself, then every proposal is indeed a concession from his point of view.

For the rest of this paper we will assume a small adaptation to the original definition of MiCRO. That is, suppose, as above that $m \leq n$. Then, instead of directly proposing ω_{m+1}, it will first do the following:

1. If ω_{m+1} was already proposed earlier by the opponent, then propose (or accept) ω_{m+1}.
2. Otherwise, check if there exists any other offer ω such that $u_1(\omega) = u_1(\omega_{m+1})$ and such that ω has already been proposed by the opponent, but not yet by our agent. If that is the case, then swap ω and ω_{m+1} on the list. Then, propose (or accept) ω.

This is also displayed in Algorithm 1, lines 13–22. Note that, since the two offers that are swapped have the same utility, the list still remains sorted in order of decreasing utility.

While MiCRO can be combined with various acceptance conditions, in this paper we will always assume it accepts a received offer if and only if it is better than or equal to the lowest offer it is, at that time, willing to propose. More precisely, if agent α_1 applies MiCRO and we define:

$$\omega_{low} := \begin{cases} \omega_{m+1} & \text{if } m \leq n \\ \omega_m & \text{if } m > n \end{cases} \tag{1}$$

(with m and n defined as before) then a received offer ω is accepted by α_1 iff $u_1(\omega) \geq \max\{u_1(\omega_{low}), rv_1\}$. See also Algorithm 1, lines 1–8.

One might think that MiCRO is very slow, because it makes concessions of minimal size. However, in practice it turns out that the opposite is true: it is very fast because it does not have to update any opponent models. Furthermore, the time it takes to sort all the offers in the offer space turned out to be negligible in our experiments in [1].

4 Experimental Results

The focus of this paper is purely on the theoretical properties of MiCRO because the experimental evidence of the strength of MiCRO has already been presented in other work. Nevertheless, we think it is useful to here briefly summarize the experimental results of MiCRO.

In our paper that introduced MiCRO [1], we presented several experiments in which MiCRO competed against top agents from the ANAC competitions of 2012, 2013, 2018, and 2019. It was shown that MiCRO consistently outperformed each of those agents, both in terms of a tournament evaluation and in terms of an empirical game-theoretical evaluation.

Algorithm 1 Describes how MiCRO decides which offer to propose or accept (note that here *offers*$[m]$ corresponds to ω_{m+1} in the text).

```
Input:
$offers$               ▷ A sorted list containing all possible offers.
$receivedOffers$       ▷ The set of all offers so far proposed by the opponent.
$m$                    ▷ The number of unique offers so far proposed by us.
$n$                    ▷ The number of unique offers so far proposed by the opponent.
$rv$                   ▷ Our reservation value.
$u$                    ▷ Our utility function.
$\omega$               ▷ The last offer received from the opponent. Is null if we have not
                         received any proposals yet.
if $m \leq n$ then
    $\omega_{low} \leftarrow offers[m]$
else
    $\omega_{low} \leftarrow offers[m-1]$
end if
if $\omega \neq null$ and $u(\omega) \geq u(\omega_{low})$ and $u(\omega) > rv$ then
    return ACCEPT($\omega$)
end if

if $m > n$ or $u(offers[m]) \leq rv$ then
    $r \leftarrow getRandomInteger(0, m)$
    return PROPOSE($offers[r]$)
end if

▷ We are planning to propose $offers[m]$. However, first see if we can find any other offer
with the same utility, but that has already been proposed to us by the opponent. If yes,
then swap the two offers on the list.
$i \leftarrow m + 1$
while $u(offers[i]) = u(offers[m])$ do
    if $offers[i] \in receivedOffers$ then
        $temp \leftarrow offers[i]$
        $offers[i] \leftarrow offers[m]$
        $offers[m] \leftarrow temp$
        break
    end if
    $i \leftarrow i + 1$
end while

▷ Propose the offer that is (now) in position $m$ on the list.
return PROPOSE($offers[m]$)
```

Furthermore, MiCRO was submitted to the ANAC 2022 competition. While it only ended in 9th place out of 19 participants, it was later shown [33] that it was in fact the best participant from a game-theoretical point of view. That is, it was shown to form the best empirical Nash equilibrium among all strategies submitted. The reason that MiCRO still ended in a relatively low position, was that MiCRO did not perform well against lower classified agents. However, MiCRO clearly outperformed the top agents when looking only at direct confrontations with those top agents. This is consistent with the notion of game-theoretical optimality, which assumes that opponents are rational and therefore do not choose a weaker strategy.

Finally, MiCRO was submitted again to ANAC in 2023 and ended in second place, out of 15 participants.

5 Consistent Negotiation Strategies

In this section we present the formal definition of a consistent negotiation strategy.

5.1 Motivation

Ideally, we would like to show that MiCRO is a best response against itself, among *all* possible negotiation strategies, but unfortunately this is not true. As we will see in Section 9, there are strategies that do form a better response against MiCRO, but, as far as we can tell, such strategies always seem to require detailed knowledge of the opponent's utility function. Since the field of automated negotiation typically assumes that such knowledge is not available we could dismiss such strategies, and instead only try to show that MiCRO is a best response against itself among all strategies that do not require such knowledge. However, it turns out that this is very difficult to formalize. Instead, we will do something else. We will prove that MiCRO is a best response against itself, among all possible *consistent* strategies, which we define below.

Our justification for focusing only on consistent strategies, is that we argue that any *rational* agent would normally be consistent. The only reason to follow an inconsistent strategy would be if you want to behave irrationally on purpose to mislead and exploit an opponent that assumes that you are rational. We will show an example of such an inconsistent strategy in Section 9. Furthermore, we argue that such inconsistent strategies only work if they have precise knowledge of the opponent's utility function and strategy. Moreover, we will show at the end of Section 7 that the assumption that the opponents are consistent can be weakened, and in Section 9 we will show that, with some small adaptations to MiCRO, it can be weakened ever further.

Remark 3. *We argue informally that, without detailed knowledge of the opponent's utility function and strategy, a rational agent would not have any reason to follow an inconsistent strategy (except perhaps, when negotiations are very close to the deadline).*

We will not attempt to prove or even formalize this claim. We simply leave it up to the readers themselves to judge whether this is a reasonable belief or not. Of course, the formal claims we make in this paper do not depend on this belief.

The main advantage of focusing only on consistent strategies, is that we do not have to make any assumptions about the knowledge that agents have about each other's utility functions.

Remark 4. *All theorems and lemmas in this paper hold regardless of whether or not agents have knowledge about each other's utility functions.*

5.2 Definition of Consistency

	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$
Alice	(10 , 2)		(9 , 3)		$\underline{(6\ ,\ 4)}$
Bob		(8 , 10)		(4 , 9)	

Table 1: Example of an inconsistent proposal (underlined). At $t = 1$ Alice makes a proposal that would yield a utility of 10 to herself and 2 to Bob. At $t = 2$ Bob makes a counter proposal that yields a utility of 8 to Alice, and 10 to Bob himself. At $t = 5$ Alice makes a proposal that is inconsistent, because it only gives her a utility of 6, while earlier, at $t = 2$, Bob already showed willingness to make a deal that gives Alice a utility of 8. It would make more sense for Alice to first re-propose that offer from $t = 2$ before proposing any offer that only gives her 6. In other words, at $t = 5$, the offer (8 , 10) has strict priority for Alice over the offer (6 , 4).

In the following, we will first define the notions of an *inconsistent proposal* (Def. 11), an *inconsistent acceptance* (Def. 12), and an *inconsistent rejection* (Def. 13), and then we define a *consistent strategy* (Def. 14) as one that never takes any such inconsistent actions. Since these definitions are somewhat involved, we strongly recommend the reader to carefully look at the examples we give before each of them, so as to convince themselves that it would indeed be irrational to follow an inconsistent strategy.

Example 1. *Suppose a buyer and a seller are negotiating the price of some item. Initially, the seller asks a price of \$100, while the buyer offers only \$75. Then, at some later time t, the seller decides to drop his price and ask \$50. Clearly, this would be silly, since the buyer has already indicated she is willing to pay \$75. Obviously, any rational seller would accept the offer of \$75, rather than conceding to \$50. Of course, it may happen that at that point the buyer's offer of \$75 is no longer valid (e.g. because the protocol does not allow to accept offers from earlier rounds), but even then the seller should at least try to re-propose the offer of \$75 first, before dropping to \$50.*

We say that at time t, the proposal of \$75 has 'strict priority' for the seller over the proposal of \$50, and since the seller proposes the offer of \$50 anyway, even though there exists another offer that has strict priority over it, we say he is making an 'inconsistent proposal'.

For a more detailed example, see Table 1.

These concepts are formalized in the following two definitions.

Definition 10. *Let s be a negotiation state, let $t \in \mathbb{R}^+$ be any time, and let ω and ω' be any two different offers. Then we say that ω' has* ***strict priority*** *over ω, for agent α_i at time t, if the following three conditions all hold:*

1. *For agent α_i, the offer ω' is better than ω:* $u_i(\omega') > u_i(\omega)$
2. *Its opponent already proposed ω' at some earlier time t':*
   $\exists t' : (3 - i, \mathfrak{p}, \omega', t') \in s_{<t}$
3. *α_i itself has not yet re-proposed ω' between t' and t:*
   $\nexists t'' \in [t', t] : (i, \mathfrak{p}, \omega', t'') \in s_{<t}$

Or, if the following four conditions all hold:

4. *Agent α_i is indifferent between ω and ω':* $u_i(\omega') = u_i(\omega)$
5. *Its opponent already proposed ω' at some earlier time t':*
   $\exists t' : (3 - i, \mathfrak{p}, \omega', t') \in s_{<t}$

	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$
Alice	(10 , 2)		(9 , 3)		accept: (4 , 9)
Bob		(8 , 10)		(4 , 9)	

Table 2: Example of an inconsistent acceptance (underlined). Alice's acceptance at $t = 5$ is inconsistent, because it only gives her a utility of 4 while at $t = 2$ Bob already made a proposal that would have given her a utility of 8. So, it would be smarter for Alice to first try to re-propose that earlier offer instead of accepting the offer from $t = 4$.

6. *α_i itself has not yet re-proposed ω' between t' and t:*
   $\nexists t'' \in [t', t] : (i, \mathfrak{p}, \omega', t'') \in s_{<t}$
7. *Its opponent has not yet proposed ω before time t:*
   $\nexists t' : (3 - i, \mathfrak{p}, \omega, t') \in s_{<t}$

(note that conditions 2 and 3 are identical to conditions 5 and 6).

In other words, for any reasonable agent α_i, if the offer ω' is better than ω, and ω' has already been proposed by the opponent, then α_i would first try to re-propose ω', before proposing ω. Furthermore, if α_i is indifferent between the two offers, and ω' has already been proposed by the opponent, while ω was not, then α_i would also first try to re-propose ω'.

We say that an agent α_i makes an inconsistent proposal, if it proposes some offer ω, while at that moment some other offer ω' has strict priority over ω for α_i.

Definition 11. *Let s be a negotiation state. Then, we say that α_i makes an **inconsistent proposal** iff there exist ω, $\omega' \in \Omega$ and $t \in \mathbb{R}$ such that the following two conditions both hold:*

- $(i, \mathfrak{p}, \omega, t) \in s$
- *At time t offer ω' has strict priority over ω for agent α_i (see Def. 10).*

Note that in [1] we used a slightly different definition of 'inconsistent proposal'. Specifically, we did not include Condition 7 of Def. 10. This was an error, since in that case it could happen that ω has strict priority over ω', while at the same time ω' also has strict priority over ω, which would make it impossible to be consistent. Therefore, the definition given here should be considered the correct one.

Example 2. *Suppose again that the seller initially rejects an offer of \$75 from the buyer, but this time it is the buyer that later drops her price and offers \$50. If the seller accepts this offer it would be inconsistent, because if he is willing to accept \$50, then he should have certainly accepted the offer of \$75. Therefore, we say this acceptance was an 'inconsistent acceptance', unless the original offer of \$75 is no longer available and the seller has already tried to re-propose that offer himself, without success.*

For a more detailed example, see Table 2.

In general, an agent α_i makes an inconsistent acceptance if it accepts an offer ω after rejecting a strictly better offer ω', unless α_i itself has already re-proposed ω'.

Definition 12. *For some given state s, an agent α_i makes an **inconsistent acceptance** iff there exist ω, $\omega' \in \Omega$ and $t, t' \in \mathbb{R}$ such that all of the following conditions hold:*

- *Agent α_i accepts offer ω, at time t:* $\quad (i, \mathfrak{a}, \omega, t) \in s$
- *Agent α_i prefers ω' over ω:* $\quad u_i(\omega') > u_i(\omega)$

	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$
Alice	(10 , 2)		(8 , 3)		$\underline{(10\ ,\ 5)}$
Bob		(7 , 10)		(9 , 9)	

Table 3: Example of an inconsistent rejection. At $t = 5$ Alice does not accept Bob's last offer which would have given her a utility of 9, but instead she proposes a different offer. This means Alice made an inconsistent rejection, because she has earlier already proposed an offer with a value of 8, so she rejected an offer that was better than some offer she already proposed herself.

- *The opponent has already proposed ω', at time t' before time t:*

$$(3 - i, \mathfrak{p}, \omega', t') \in s_{<t}$$

- *Between the time t' that the opponent proposed ω' and the time t that α_i accepted ω, agent α_i did not try to re-propose ω':*

$$\nexists t'' : \quad t' < t'' < t \quad \wedge \quad (i, \mathfrak{p}, \omega', t'') \in s$$

Example 3. *Suppose that at some point, the seller proposes a price of \$80, but then later he rejects a better offer of \$120 from the buyer. We call this is an 'inconsistent rejection'.*

For a more detailed example, see Table 3.

In general, we say an agent α_i makes an *inconsistent rejection* whenever it rejects an offer ω that is better than or equal to some offer ω' that α_i itself has already proposed earlier.

Definition 13. *We say that, for some given state s, α_i makes an* ***inconsistent rejection*** *iff there exist $\omega, \omega' \in \Omega$ (possibly $\omega = \omega'$), and $t \in \mathbb{R}$ such that all of the following conditions hold:*

- *Agent α_i rejects ω at time t (see Def. 9):* $\quad rej_s(i, \omega, t)$
- *Agent α_i proposed offer ω' at some earlier time:* $\quad \exists t' : (i, \mathfrak{p}, \omega', t') \in s_{<t}$
- *For α_i, offer ω is better than or equal to ω':* $\quad u_i(\omega) \geq u_i(\omega')$

Definition 14. *We say an agent or a negotiation strategy is* ***consistent*** *if it never makes any inconsistent proposals, inconsistent acceptances, or inconsistent rejections.*

While we think it should be obvious that inconsistent proposals and acceptances do not make sense for a rational agent, it may be less obvious for inconsistent rejections. After all, it is possible that an agent α_1 proposes some offer ω, but later, thanks to some opponent modeling algorithm, learns that its opponent α_2 may be willing to accept certain offers that α_1 prefers over ω. Therefore, α_1 could change its mind and refuse to accept ω. However, this would mean that α_1 plays a rather weak strategy, because it means that when α_1 proposed ω, it made a concession that was too large, too early.

The following theorem is proved in Appendix A.

Theorem 1. *MiCRO is consistent.*

5.3 Useful Lemmas

We will now prove two lemmas that will be important in the rest of the paper.

The following lemma plays a key role in the proofs of Theorem 2 and Theorem 3.

Lemma 1. *Suppose that α_1 applies MiCRO and that α_2 is consistent. Furthermore, let s denote the final state of a negotiation (under the AOP) between the two agents, and suppose there are three offers ω_i, ω_j, and ω_k (with $\omega_j \neq \omega_k$) such that the following conditions all hold:*

1. $u_1(\omega_i) > u_1(\omega_k)$

2. $u_2(\omega_i) \geq u_2(\omega_j)$

3. At some point during the negotiations α_2 has proposed ω_j, i.e.: $\exists t : (2, \mathfrak{p}, \omega_j, t) \in s$.

Then the negotiations did not end with ω_k as the accepted offer (i.e.: $\Pi_{stat}(s) \neq \omega_k$).

Proof. Suppose the contrary, i.e. that ω_k is the accepted offer. Let us define t_{acc} to be the time at which ω_k was accepted. We know that before that, α_1 must have already proposed ω_i (by Condition 1 and the definition of MiCRO), and that α_2 must have already proposed ω_j (by Condition 3). In other words, we know that there are two numbers, t and t', such that:

- $(1, \mathfrak{p}, \omega_i, t') \in s_{<t_{acc}}$
- $(2, \mathfrak{p}, \omega_j, t) \in s_{<t_{acc}}$

This means we can consider two separate cases, namely the case that $t < t'$ and the case that $t' < t$ (note that $t = t'$ is impossible, because the AOP does not allow two agents to make a proposal at the same time). We can alternatively denote these two cases as $(1, \mathfrak{p}, \omega_i, t') \in s_{>t}$ and $(1, \mathfrak{p}, \omega_i, t') \in s_{<t}$. Now, the second case can be further split up into two subcases 2a and 2b depending on whether or not α_2 has proposed ω_i before before t. Then, case 2b can be split up again into two subcases 2b1 and 2b2 , depending on whether $u_2(\omega_i)$ is greater than $u_2(\omega_j)$, or equal. And finally, case 2b2 can again split up into two smaller cases, depending on whether or not α_1 has proposed ω_j before t. So, we now have five separate cases, and we will show for each of them that there is a contradiction.

Case 1: $(1, \mathfrak{p}, \omega_i, t') \in s_{>t}$

In this case, if α_2 replies to $(1, \mathfrak{p}, \omega_i, t')$ by accepting ω_i we would have a contradiction because the assumption was that the negotiations end with ω_k as the accepted offer. On the other hand, if α_2 does not accept ω_i, then this would be an inconsistent rejection (since $u_2(\omega_i) \geq u_2(\omega_j)$ and α_2 has already proposed ω_j at time t), so again we have a contradiction because α_2 was supposed to be consistent. A third possibility would be that α_2 does not reply to $(1, \mathfrak{p}, \omega_i, t')$ at all, because the deadline passes, but that again would be in contradiction with the assumption that the agents agree on ω_k.

Case 2a: $(1, \mathfrak{p}, \omega_i, t') \in s_{<t} \quad \wedge \quad \exists t'' : (2, \mathfrak{p}, \omega_i, t'') \in s_{<t}$

In this case, note that at time t both agents have previously proposed ω_i. Clearly, if either of them accepted the offer, then we would have a contradiction with the assumption that ω_k was going to be the accepted offer. However, if neither of them accepted it, then it means that one of them is rejecting an offer he already proposed himself before, which is an inconsistent rejection (this is a special case of Def. 13 with $\omega_i = \omega = \omega'$).

Case 2b1: $(1, \mathfrak{p}, \omega_i, t') \in s_{<t} \quad \wedge \quad \nexists t'' : (2, \mathfrak{p}, \omega_i, t'') \in s_{<t} \quad \wedge \quad u_2(\omega_i) > u_2(\omega_j)$

In this case, we have that ω_i has strict priority over ω_j for α_2 at time t, so $(2, \mathfrak{p}, \omega_j, t)$ was an inconsistent proposal.

Case 2b2a: $(1, \mathfrak{p}, \omega_i, t') \in s_{<t} \quad \wedge \quad \nexists t'' : (2, \mathfrak{p}, \omega_i, t'') \in s_{<t} \quad \wedge \quad u_2(\omega_i) = u_2(\omega_j) \quad \wedge \quad \nexists t'' : (1, \mathfrak{p}, \omega_j, t'') \in s_{<t}$

In this case, again, we have that ω_i has strict priority over ω_j for α_2 at time t, so $(2, \mathfrak{p}, \omega_j, t)$ was an inconsistent proposal.

Case 2b2b: $(1, \mathfrak{p}, \omega_i, t') \in s_{<t} \quad \wedge \quad \nexists t'' : (2, \mathfrak{p}, \omega_i, t'') \in s_{<t} \quad \wedge \quad u_2(\omega_i) = u_2(\omega_j) \quad \wedge \quad \exists t'' : (1, \mathfrak{p}, \omega_j, t'') \in s_{<t}$

In this case, α_1 itself has already proposed ω_j before α_2 proposed it at time t. This means that α_1 would then reply to α_2's proposal by accepting ω_j (by definition of MiCRO). However, since we have assumed that the negotiations end with agreement ω_k, and that $\omega_j \neq \omega_k$, we again have a contradiction.

□

The next lemma is a small variation of Lemma 1, and will be useful in our proof of Proposition 1. Compared to Lemma 1 we drop the assumption that $\omega_j \neq \omega_k$, but at the expense of the stronger condition $u_2(\omega_i) > u_2(\omega_j)$.

Lemma 2. *Suppose that α_1 applies MiCRO and that α_2 is consistent. Furthermore, let s denote the final state of a negotiation (under the AOP) between the two agents, and suppose there are three offers ω_i, ω_j, and ω_k (possibly with $\omega_j = \omega_k$) such that the following conditions all hold:*

1. $u_1(\omega_i) > u_1(\omega_k)$
2. $u_2(\omega_i) > u_2(\omega_j)$
3. *At some point during the negotiations α_2 has proposed ω_j, i.e.:* $\exists t : (2, \mathfrak{p}, \omega_j, t) \in s$.

then negotiations did not end with ω_k as the accepted offer.

Proof. To prove this we simply refer to the proof of Lemma 1. The main difference, is that we now have to consider that ω_j and ω_k could be equal. However, in the proof of Lemma 1 the fact that they were different only played a role in Case 2b2b, and since we are now assuming that $u_2(\omega_i) > u_2(\omega_j)$, this case no longer applies. □

6 The Balance Set

In this section we present the notion of the *balance set*. For any negotiation domain, the balance set is a subset of its offer space Ω. Its importance, is that it is the set of possible outcomes that two agents may agree upon if they both apply the MiCRO strategy (which we prove at the end of this section).

We will later show (in Section 8) that in many of the ANAC domains the balance set happens to coincide with the set of optimal agreements, which means that in such domains two MiCRO agents would always negotiate an optimal deal.

6.1 Definition of the Balance Set

In the following, for any real number x we define Ω_i^x to be the set of all offers for which $u_i(\omega) \geq x$.

$$\Omega_i^x := \{\omega \in \Omega \mid u_i(\omega) \geq x\}$$

Note that the set of offers that MiCRO is willing to propose or accept is always of the form Ω_i^x, with x decreasing every time the opponent makes a new new proposal.

Also, in the rest of this paper we will always assume that $(\omega_1, \omega_2, \ldots \omega_K)$ denotes a list containing all offers in ω, sorted in order of decreasing utility for agent α_1 (where $K = |\Omega|$ is the size of the domain), and that π is the permutation of the integers 1 to K, such that $(\omega_{\pi(1)}, \omega_{\pi(2)}, \ldots \omega_{\pi(K)})$ is a list of all offers sorted in order of decreasing utility for agent α_2. Furthermore, for any integer i between 1 and K we define:

$$x_i := u_1(\omega_i) \quad \text{and} \quad y_i := u_2(\omega_{\pi(i)}).$$

Definition 15. *We define the* ***balance index*** *b to be the smallest integer for which $\Omega_1^{x_b} \cap \Omega_2^{y_b} \neq \emptyset$, and we define the* ***balance set*** *to be the set $\Omega_1^{x_b} \cap \Omega_2^{y_b}$. Furthermore, we define*

$$x^\beta := min\{u_1(\omega) \mid \omega \in \Omega_1^{x_b} \cap \Omega_2^{y_b}\}$$
$$y^\beta := min\{u_2(\omega) \mid \omega \in \Omega_1^{x_b} \cap \Omega_2^{y_b}\}$$

The values x^β and y^β are called the ***balance values*** *of α_1 and α_2 respectively.*

These concepts are also illustrated in Figures 1 and 2. The intuitive idea is that if α_1 and α_2 both apply the MiCRO strategy, then x_i and y_i are the minimum utility values they are respectively willing to accept after they have both made $i-1$ proposals, and $\Omega_1^{x_i}$ and $\Omega_2^{y_i}$ represent the sets of offers they are then respectively willing to accept. Initially, the two agents will only be willing to accept the offers in $\Omega_1^{x_1} = \{\omega_1\}$ and $\Omega_2^{y_1} = \{\omega_{\pi(1)}\}$ respectively, and their intersection will, in general, be disjoint. But, as the negotiation progresses, the number of unique proposals they have made increases, which means that i increases, which means that x_i and y_i decrease (as a function of i), and so $\Omega_1^{x_i}$ and $\Omega_2^{y_i}$ become larger. Then, after they have both made $b-1$ unique offers, their intersection becomes nonempty, so at that point there are some offers that both agents are willing to accept. The balance set is by definition the set of these offers, and so the utility values the agents receive from the accepted offer must be greater than or equal to their balance values.

The balance index defines how many negotiation rounds are necessary in order for two MiCRO agents to come to an agreement. It turns out that for many domains that were used in ANAC, the balance index is much smaller than the total number of offers, meaning that MiCRO can still negotiate successfully even though the domain might seem too large. For example, the *Smartphone* domain from ANAC 2013 has a size of $|\Omega| = 12\ 000$, but its balance index is only 139, so two MiCRO agents only need to exchange 139 proposals each, to come to an agreement.

From the definition it is immediately clear that $x^\beta \geq x_b$ and $y^\beta \geq y_b$ must hold. However, it may not be clear that these inequalities can sometimes be strict. Therefore, we show in Figure 3 an example where indeed $x^\beta > x_b$.

The following lemma is necessary for our proof of Lemma 4. Geometrically, it says that if we draw a diagram of the domain, such as in Figures 1 and 2, we can always draw a horizontal line and a vertical line such that each element of the balance set is on one of those two lines.

Lemma 3. *Let x^β and y^β denote the balance values. Then, for any offer ω in the balance set we either have $u_1(\omega) = x^\beta$ or $u_2(\omega) = y^\beta$.*

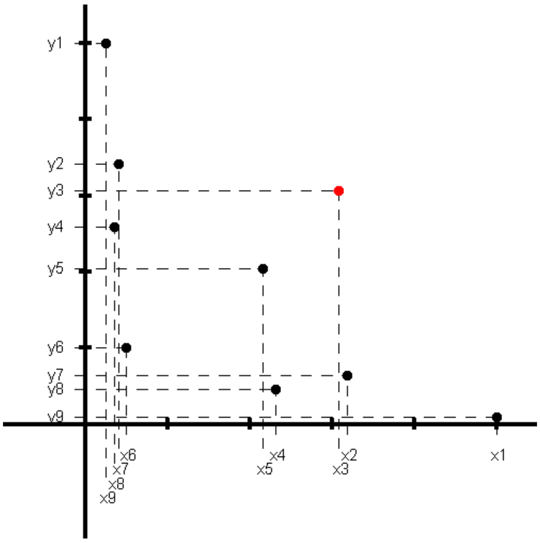

Fig. 1: Diagram of the *Ultimatum* domain, which was used in ANAC 2013, containing nine offers. The horizontal axis represents the utility u_1 of agent α_1, while the vertical axis represents the utility u_2 of agent α_2. Each dot represents one offer. The red dot represents an offer that maximizes both the sum and the product of the utility of the two agents.

Proof. Let $\omega_i \in \Omega_1^{x_b} \cap \Omega_2^{y_b}$ be some element from the balance set. We need to prove that either $u_1(\omega_i) = x^\beta$ or $u_2(\omega_i) = y^\beta$. In fact, we will prove the slightly stronger conclusion that either $u_1(\omega_i) = x_b$ or $u_2(\omega_i) = y_b$.

Recall that, by definition of the balance set, $\Omega_1^{x_{b-1}} \cap \Omega_2^{y_{b-1}}$ must be empty, so we have $\omega_i \notin \Omega_1^{x_{b-1}} \cap \Omega_2^{y_{b-1}}$, which means that either $x_{b-1} > u_1(\omega_i)$ or $y_{b-1} > u_2(\omega_i)$. Furthermore, since $\omega_i \in \Omega_1^{x_b} \cap \Omega_2^{y_b}$ we have $u_1(\omega_i) \geq x_b$ and $u_2(\omega_i) \geq y_b$. Combining these facts we conclude that we must have:

$$x_{b-1} > u_1(\omega_i) \geq x_b \quad \text{or} \quad y_{b-1} > u_2(\omega_i) \geq y_b. \tag{2}$$

Now, recall that x_{b-1} and x_b are the utility values for α_1 of ω_{b-1} and ω_b, and these two offers are consecutive in α_1's sorted list, so there is no offer ω for which $x_{b-1} > u_1(\omega) > x_b$. And for the same reason there is no offer for which $y_{b-1} > u_2(\omega) > y_b$. Combining this with (2), we conclude that we must have either $u_1(\omega_i) = x_b$ or $u_2(\omega_i) = y_b$, but x_b and y_b are the lowest utility values that any offer in $\Omega_1^{x_b} \cap \Omega_2^{y_b}$ could possibly have, so if $u_1(\omega_i) = x_b$ then $u_1(\omega_i) = x^\beta$ and similarly, if $u_2(\omega_i) = y_b$ then $u_2(\omega_i) = y^\beta$. □

6.2 Pareto Optimality

The concepts defined in the following definition are common in the literature, but we repeat them here for the sake of self-containment.

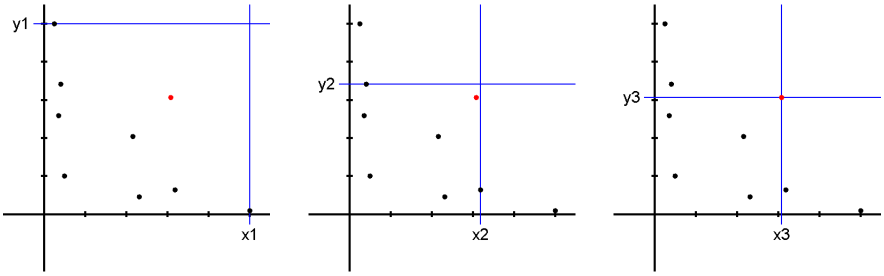

Fig. 2: The Ultimatum domain. **Left:** the blue lines are drawn through the values x_1 and y_1. Therefore, the area to the right of the vertical blue line represents $\Omega_1^{x_1}$ and the area above the horizontal blue line represents $\Omega_2^{y_1}$. We see that $\Omega_1^{x_1} \cap \Omega_2^{y_1} = \emptyset$. **Center:** the blue lines are drawn through the values x_2 and y_2. Therefore, the area to the right of the vertical blue line represents $\Omega_1^{x_2}$ and the area above the horizontal blue line represents $\Omega_2^{y_2}$. We see that $\Omega_1^{x_2} \cap \Omega_2^{y_2} = \emptyset$. **Right:** the blue lines are drawn through the values x_3 and y_3. We now see that $\Omega_1^{x_3} \cap \Omega_2^{y_3}$ contains exactly one offer. This means that the balance index equals 3, and that $\Omega_1^{x_3} \cap \Omega_2^{y_3}$ is the balance set. Furthermore, we notice that the only element of the balance set happens to be exactly the offer that maximizes both the product and the sum of the agents' utilities, so we say this is a *balanced* domain.

Definition 16. *We say an offer ω_i **weakly dominates** another offer ω_j if $u_1(\omega_i) \geq u_1(\omega_j)$ and $u_2(\omega_i) \geq u_2(\omega_j)$, and at least one of these inequalities is strict. We say that ω_i **strongly dominates** ω_j if both inequalities are strict. An offer ω_i is **weakly Pareto-optimal** if it is not strongly dominated by any other offer, and we say ω_i is **strongly Pareto-optimal** if it is not weakly dominated by any other offer.*

Lemma 4. *All elements of a balance set are weakly Pareto optimal.*

Proof. Suppose that ω_i is an element of the balance set $\Omega_1^{x_b} \cap \Omega_2^{y_b}$ which is not weakly Pareto optimal. So, there exists another offer ω_j with $u_1(\omega_j) > u_1(\omega_i)$ and $u_2(\omega_j) > u_2(\omega_i)$. Clearly, this means that $\omega_j \in \Omega_1^{x_b} \cap \Omega_2^{y_b}$, so ω_j is also in the balance set. Now, note that by definition of the balance values we have $u_1(\omega_i) \geq x^\beta$ and $u_2(\omega_i) \geq y^\beta$, and by Lemma 3, we must have either $u_1(\omega_j) = x^\beta$ or $u_2(\omega_j) = y^\beta$. Combined this means that either $u_1(\omega_i) \geq u_1(\omega_j)$ or $u_2(\omega_i) \geq u_2(\omega_j)$, which contradicts our assumption about ω_j. □

The following proposition will be useful later on to prove Theorem 2, but we think it is also interesting by itself.

Proposition 1. *If MiCRO makes an agreement with a consistent opponent (under the AOP), then this agreement will be weakly Pareto-optimal.*

Proof. We prove this by contradiction. Assume that α_1 applies MiCRO, and that the two agents make an agreement ω_k that is not weakly Pareto-optimal, i.e. there exists some other offer ω_i with $u_1(\omega_i) > u_1(\omega_k)$ and $u_2(\omega_i) > u_2(\omega_k)$.

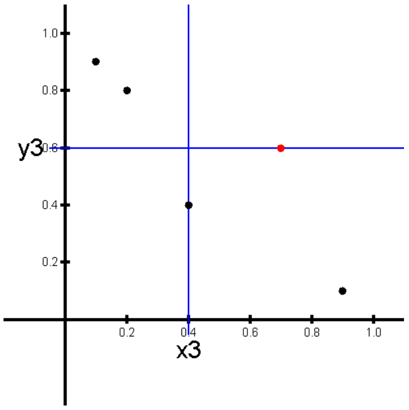

Fig. 3: An example of a domain where $x^\beta > x_b$. This domain has balance index $b = 3$. Note that the offer indicated in red is the only element of the balance set, and has utility values $u_1(\omega) = 0.7$ and $u_2(\omega) = 0.6$. So, these are also the balance values: $x^\beta = 0.7$, $y^\beta = 0.6$. However, we see that $x_b = x_3 = 0.4$.

Let s denote the final state of the negotiations, and t the time at which ω_k was accepted. We consider two cases: 1) the case that α_1 accepted ω_k and 2) the case that α_2 accepted ω_k.

Case 1: $(\mathbf{1}, \mathfrak{a}, \omega_\mathbf{k}, \mathbf{t}) \in \mathbf{s}$. If α_1 accepted ω_k then it must have been proposed by α_2. This means all conditions of Lemma 2 are satisfied, with $\omega_j = \omega_k$, so indeed we have a contradiction.

Case 2: $(\mathbf{2}, \mathfrak{a}, \omega_\mathbf{k}, \mathbf{t}) \in \mathbf{s}$. If α_2 accepted ω_k, then α_1 must have proposed it. Since MiCRO proposes all possible offers one by one in order of decreasing utility, this means that, at some earlier time t', it must also have proposed ω_i, i.e. $(1, \mathfrak{p}, \omega_i, t') \in s_{<t}$. Note that the first three conditions of Def. 12 are now satisfied. So, if the fourth one is true as well then α_2 accepting ω_k would be an inconsistent acceptance, which is a contradiction. On the other hand, if the fourth condition does not hold, we have $\exists t'' : t' < t'' < t \wedge (2, \mathfrak{p}, \omega_i, t'') \in s$ (i.e. at t'' agent α_2 has re-proposed ω_i after α_1 proposed it earlier at time t'). But in that case, by definition of MiCRO, α_1 would have accepted ω_i directly after t'', so the final agreement would have been ω_i instead of ω_k. This is again a contradiction. □

6.3 Importance of the Balance Values

In this subsection we will show that, as long as its opponents are consistent, MiCRO never makes agreements below its balance values.[3]

Theorem 2. *If an agent α_i applies MiCRO and its opponent is consistent, then (under the AOP) they will never make any agreement ω for which $u_i(\omega)$ is below α_i's balance value.*

Proof. We will assume (w.l.o.g.) that it is α_1 that applies MiCRO.

[3] It may occasionally happen that MiCRO makes *proposals* below its balance value, but a consistent opponent would never accept those.

Let ω_i be an offer in $\Omega_1^{x_b} \cap \Omega_2^{y_b}$ such that $x_i = x^\beta$ (i.e. ω_i is an element in the balance set, for which α_1's utility is minimal) and let ω_k be any offer for which $x_i > x_k$. We need to prove that ω_k will never be accepted. We distinguish two separate cases, namely 1) the case that $x_b > x_k$ and 2) the case that $x_k \geq x_b$.

Case 1: Suppose $x_b > x_k$. We will show that there is an offer ω_j such that the conditions of Lemma 1 are satisfied. First, note that the first condition of Lemma 1 is indeed satisfied because, as we already noted, $x_i > x_k$. Furthermore, by definition of MiCRO, α_1 would only propose or accept ω_k if α_2 has made at least $k-1$ different proposals, and since $x_b > x_k$, which implies $b < k$, this means that α_2 must have made at least b different proposals. Note that by definition of the number y_b there are at most $b-1$ proposals ω for which $u_2(\omega) > u_2(\omega_{\pi(b)}) = y_b$, so α_2 must have proposed at least one proposal ω_j for which $y_b \geq u_2(\omega_j)$. Finally, note that since ω_i is in the balance set we have $u_2(\omega_i) \geq y_b$, and thus $u_2(\omega_i) \geq u_2(\omega_j)$. This means the second and third condition of Lemma 1 are also satisfied, which proves Case 1.

Case 2: Suppose $x_k \geq x_b$. We first claim that in this case we have $u_2(\omega_i) > u_2(\omega_k)$. Suppose the opposite, i.e. that $u_2(\omega_k) \geq u_2(\omega_i)$. Note that from the assumption that ω_i is in the balance set we have $u_2(\omega_i) \geq y_b$, so we then must have $u_2(\omega_k) \geq y_b$. But note that if both $x_k \geq x_b$ and $u_2(\omega_k) \geq y_b$ hold, then $\omega_k \in \Omega_1^{x_b} \cap \Omega_2^{y_b}$, and therefore, by the definition of x^β, we have $x_k \geq x^\beta = x_i$, but that is in contradiction to our assumption that $x_i > x_k$. So we conclude that $u_2(\omega_i) > u_2(\omega_k)$ indeed holds. If we now combine this with the fact that $x_i > x_k$ we conclude that ω_i strictly dominates ω_k. Therefore, by Proposition 1, ω_k could never be the accepted offer, which proves Case 2. □

Corollary 1. *If two agents that both apply the MiCRO strategy make an agreement (under the AOP), then that agreement must be an element of the balance set.*

Proof. Suppose they make an agreement ω. Then, by combining Theorem 1 and Theorem 2, it follows that we must have $u_1(\omega) \geq x^\beta$ and $u_2(\omega) \geq y^\beta$. This means that, by definition of the balance values, we must also have $u_1(\omega) \geq x_b$ and $u_2(\omega) \geq y_b$, which in turn means that ω is in the balance set. □

When the balance set contains more than one offer, the question which of those offers two MiCRO agents would agree upon depends on who starts the negotiations. For more details about this we refer to Appendix B.

7 Equilibrium Properties of MiCRO

In this section we will prove that MiCRO is a best response against itself. While our goal is to prove this under the assumption that the agents do not have knowledge about each other's utility functions, we actually prove something much stronger, because we never actually use this assumption anywhere. That is, all our lemmas and theorems hold regardless of whether the agents have such knowledge or not.

Specifically, if agent α_1 applies MiCRO, then it does not matter how much information it has about the utility function of α_2, because the MiCRO strategy simply does not use such information. Furthermore, when we say that the best response for α_2 is to also use MiCRO, we mean that there is no better strategy for α_2 *even if* α_2

has full knowledge of α_1's utility function, and therefore it *certainly* does not have any better strategy if it does not have such knowledge.[4]

7.1 MiCRO is an Equilibrium Strategy

Before we can prove our main theorem we need the following lemma.

Lemma 5. *Suppose we have a balance set with two different offers ω_i and ω_k, such that $u_1(\omega_i) > u_1(\omega_k)$ and $u_2(\omega_k) > u_2(\omega_i)$ (i.e. neither of the two weakly dominates the other). Then, there are exactly $b-1$ offers ω with $u_1(\omega) > u_1(\omega_k)$ and exactly $b-1$ offers ω with $u_2(\omega) > u_2(\omega_i)$, where b is the balance index.*

Proof. By definition of the balance set we have $u_1(\omega_k) \geq x_b$ and $u_2(\omega_i) \geq y_b$, so we must have:

$$u_1(\omega_i) > x_b \quad \text{and} \quad u_2(\omega_k) > y_b$$

We will only give the proof that there are exactly $b-1$ offers ω with $u_1(\omega) > u_1(\omega_k)$. The proof that there are also exactly $b-1$ offers ω with $u_2(\omega) > u_2(\omega_i)$ goes analogously.

First, suppose the number of offers ω with $u_1(\omega) > u_1(\omega_k)$ is less than $b-1$. Then, for the offer ω_{b-1} that appears on position $b-1$ in α_1's list, we would have $u_1(\omega_k) \geq u_1(\omega_{b-1}) = x_{b-1}$. Furthermore, since $u_2(\omega_k) > y_b$, we have that $u_2(\omega_k) \geq y_{b-1}$, and thus that $\omega_k \in \Omega_1^{x_{b-1}} \cap \Omega_2^{y_{b-1}}$. But that is not possible, because, by definition of the balance index, that set should be empty. On the other hand, if there were more than $b-1$ offers ω for which $u_1(\omega) > u_1(\omega_k)$, then we would have $u_1(\omega_b) > u_1(\omega_k)$. But then $\omega_k \notin \Omega_1^{x_b}$ and thus $\omega_k \notin \Omega_1^{x_b} \cap \Omega_2^{y_b}$, which is in contradiction to the assumption that ω_k is in the balance set. □

We will next state our main theorem. Although it requires a fair amount of conditions, it happens that for all ANAC domains that we analyzed (those of ANAC 2012, 2013, and 2021), the first two conditions are indeed satisfied. Moreover, note that the second condition is only a very minor one, since we recall from Lemma 4 that the elements of a balance set are always at least *weakly* Pareto-optimal. Regarding the third condition, experiments in [1] have shown that the typical deadline of 180 seconds used in the ANAC competitions is indeed sufficiently long for two MiCRO agents to come to an agreement, so this last condition typically also holds.[5]

Remark 5. *All three conditions of Theorem 3 typically hold in (the main leagues of) the ANAC competitions.*

Theorem 3. *Consider the extensive-form game defined by a negotiation domain D and the alternating offers protocol with deadline T (as defined in Section 3.2), and assume that the following conditions hold:*

- *The balance values of D lie above the agents' respective reservation values.*
- *All elements of the balance set of D are strongly Pareto-optimal.*
- *The deadline T is sufficiently long for two MiCRO agents to come to an agreement.*

[4] For readers familiar with Bayesian games, we can say that MiCRO vs. MiCRO is an *ex-post* Nash Equilibrium.

[5] In some recent editions of ANAC a deadline of 60 seconds was used, but even in that case MiCRO only fails on the very largest domains.

Then, among all consistent negotiation strategies, MiCRO is a best response against itself.

Proof. Suppose α_1 plays MiCRO, and suppose that α_2 has two options to select as its strategy: MiCRO and some other consistent strategy σ. We will show that if α_2 selects σ, it will not be able to score higher than when it selects MiCRO.

First note that if both agents apply MiCRO, and the deadline is long enough, then the agents will certainly come to an individually rational agreement. This follows directly from the definition of MiCRO and the assumption that the balance values lie above the reservation values. Let us call this agreement ω_i.

Now, let us first suppose that if α_2 selects σ, then they do *not* come to an agreement. In that case, the agents receive their respective reservation values, so α_2 would have been better off selecting MiCRO. So, for this case the theorem indeed holds.

Next, let us suppose that if α_2 selects σ, then they *do* come to an agreement, which we will call ω_k. To prove the theorem we need to show that α_2 does not prefer ω_k over ω_i. That is, we will show that the following assumption leads to contradiction:

$$u_2(\omega_k) > u_2(\omega_i) \tag{3}$$

The rest of this proof consists of showing that the conditions of Lemma 1 are satisfied, so indeed we have a contradiction. We will do this in three steps: Firstly, we will show that $u_1(\omega_i) > u_1(\omega_k)$ holds (which is the first condition of Lemma 1). Secondly, we will show that ω_k must be in the balance set. Thirdly, we will use this fact to show that α_2 must have proposed at least one offer ω_j for which $u_2(\omega_i) \geq u_2(\omega_j)$ before ω_k was accepted (the other two conditions of Lemma 1).

Step 1: We know from Corollary 1 that ω_i must be in the balance set. Therefore, by Eq. (3) and the assumption that all elements of the balance set are strongly Pareto-optimal, we have: $u_1(\omega_i) > u_1(\omega_k)$,

Step 2: Since ω_i is in the balance set, we have $u_2(\omega_i) \geq y_b$. Combined with Eq. (3), this leads to:

$$u_2(\omega_k) > y_b \tag{4}$$

Furthermore, by Theorem 2 we must have $u_1(\omega_k) \geq x^\beta$. And since by definition of the balance values we have $x^\beta \geq x_b$, we have:

$$u_1(\omega_k) \geq x_b \tag{5}$$

Combining Equations (4) and (5) we conclude that ω_k is also in the balance set.

Step 3: We will now show that α_2 must have proposed at least one offer ω_j with $u_2(\omega_i) \geq u_2(\omega_j)$ before ω_k was accepted. Since we know that ω_i and ω_k are both in the balance set, and since we have assumed that the elements of the balance set are strongly Pareto-optimal, we can apply Lemma 5. Thus, we know there are exactly $b-1$ offers ω with $u_2(\omega) > u_2(\omega_i)$. This means that, at any time during the negotiations, we can distinguish between the following four situations:

- *Situation 1:* α_2 has made less than $b-1$ different proposals.
- *Situation 2:* α_2 has made more than $b-1$ different proposals.

- *Situation 3:* α_2 has made exactly $b-1$ different proposals, and for at least one of them it holds that $u_2(\omega_i) \geq u_2(\omega)$.
- *Situation 4:* α_2 has made exactly $b-1$ different proposals, and they are exactly those offers for which $u_2(\omega) > u_2(\omega_i)$.

In the first situation, α_1 would not yet be willing to propose or accept ω_k. This is because, by Lemma 5, we know there are exactly $b-1$ offers that are strictly better for α_1 than ω_k, which means that ω_k must be at position b or later on MiCRO's list, and thus α_1 would only be willing to propose or accept it when α_2 has made at least $b-1$ different proposals. So, α_1 will propose some other offer, and α_2 will not accept that offer, because that would contradict the assumption that negotiations end with ω_k being accepted. Therefore, negotiations will continue until at least one of the three other situations has occurred. If Situation 2 or 3 occurs, then indeed we have shown what we wanted to show. Therefore, we are only left to consider Situation 4. In this situation α_2 must have already proposed ω_k (by Eq. 3). The question is now how α_1 responds to this situation. We can consider the following options:

- *Response 1:* α_1 accepts ω_k.
- *Response 2:* α_1 proposes ω_k.
- *Response 3:* α_1 accepts some other offer.
- *Response 4:* α_1 proposes some other offer.

Now, it is important to understand that if α_1 and α_2 *both* play MiCRO, then at some point Situation 4 would also occur, and that α_1 should then reply to Situation 4 in exactly the same way, as when α_2 plays σ. This is because, by definition of MiCRO, α_1's response only depends on which unique offers have been proposed by the opponent, and not on the order in which they were proposed.[6]

We therefore conclude that Response 1 cannot occur, because if α_1 accepts ω_k, then it would do the same if α_2 plays MiCRO, but that would be in contradiction with the assumption that MiCRO vs. MiCRO results in ω_i being accepted.

For the same reason, Response 2 also cannot occur, because if α_1 proposes ω_k when α_2 plays MiCRO, then α_2 would accept it (since α_2 itself had already proposed ω_k earlier), but that would again be in contradiction to the assumption that MiCRO vs. MiCRO results in ω_i being accepted.

Furthermore, the fact that Response 3 cannot occur is trivial, because we have assumed that the agents agree on ω_k.

So, we conclude that Response 4 must occur, which means that α_1 will propose some offer that is not ω_k, and therefore α_2 will not accept that offer (because we assumed only ω_k would be accepted). This means that negotiations must continue until either Situation 2 or Siutation 3 has occurred, which means that at some point α_2 must propose some offer ω_j with $u_2(\omega_i) \geq u_2(\omega_j)$, which is what we wanted to show.

Combining this result with the result of Step 1, we see that all conditions of Lemma 1 are satisfied, so we have a contradiction. □

[6] Actually, α_1's response also depends on how exactly MiCRO sorts its list when there are multiple offers between which it is indifferent. However, we are looking at what happens when α_2 makes a *unilateral* deviation from MiCRO to σ, so we have to assume that α_1 sorts his list in exactly the same way in either case.

7.2 Semi-consistent Strategies

We will now present a stronger version of Theorem 3, which allows the opponents to make inconsistent acceptances.

Definition 17. *We say a strategy is* ***semi-consistent*** *if it never makes any inconsistent proposals or inconsistent rejections (but it may make inconsistent acceptances).*

Theorem 4. *Suppose the same conditions as Theorem 3. Then, among all semi-consistent negotiation strategies, MiCRO is a best response against itself.*

Proof. Suppose again that α_1 plays MiCRO, and that α_2 plays some alternative strategy σ. If α_2 does not make any inconsistent acceptance, then it behaves like a consistent strategy, so by Theorem 3 σ cannot be a better response against MiCRO. On the other hand, suppose that α_2 does accept some offer ω and that this is an inconsistent acceptance. This means that at some earlier time α_1 proposed a better offer ω'. Then, α_2 would have been better off accepting that offer, and thus σ was not a best response. □

7.3 Is MiCRO vs. MiCRO a Subgame Perfect Equilibrium?

We have shown that MiCRO vs. MiCRO forms a Nash equilibrium. However, since we defined negotiations as an extensive-form game, an important question is whether it is also a subgame perfect equilibrium (SPE). We will here quickly discuss this question, informally. Recall that a pair of strategies is an SPE iff, for every game state s, it is a Nash equilibrium on the subgame starting at s.

First, note that the common way to prove subgame perfection is to use backward induction. However, we cannot use this technique for negotiations, because, as we recall from Section 3.2, the state transition is determined by a random variable which can have an infinite number of possible values. Therefore, each state has an infinite number of possible successor states, which makes backward induction difficult to apply. Instead, we use a different approach.

Suppose we have a state $s = (a_1, a_2, \dots, a_k)$ and let t_k be the time of a_k. Then, as long as t_k is small enough compared to T, the proof of Theorem 3 continues to hold when the negotiations reach state s. In other words, for any state s that is far enough from the deadline, MiCRO vs. MiCRO is indeed a Nash equilibrium for the subgame starting at s.

However, if the the negotiations reach a state that is too close to the deadline, it may happen that there is not enough time for two MiCRO agents to make an agreement, and so the agents would be better off if they deviated to a strategy that concedes faster, in order to secure a deal. Therefore, MiCRO vs. MiCRO is technically not an SPE.

This means that an agent could try to stall the negotiations on purpose, so as to reach a state where the agents are forced to deviate to a different strategy. However, under the conditions of Theorem 3, for at least one of the two agents this would result in a worse outcome (since any deal between MiCRO and MiCRO would be Pareto optimal). So, an agent that tries to apply this approach would have to make sure it concedes slower than its opponent. But that would mean that if two agents *both* tried to apply this approach, then they would fail to make an agreement. Therefore, trying to stall the negotiations in order to reach a subgame where MiCRO vs. MiCRO is not a Nash equilibrium will not yield any benefit.

We therefore argue, informally, that, even though MiCRO vs. MiCRO is technically not an SPE, the players cannot benefit from deviating to any other consistent strategy. That is, it forms a Nash equilibrium on all *relevant* subgames, and therefore still captures the essence of an SPE.

8 Optimality of MiCRO in the ANAC Domains

Theorem 3 shows that, under the given conditions, MiCRO vs. MiCRO forms a Nash equilibrium. By itself this does not necessarily mean that MiCRO is an optimal strategy, because there could be other strategies that also form Nash equilibria but with better outcomes. However, in this section we will show that on many of the ANAC domains the agreement made between two MiCRO agents is, in fact, an optimal one.

8.1 Optimality

We first need to formally define the notion of an optimal agreement. This is a complex topic and many different definitions have been proposed in the literature. Here, we will focus on two such definitions: the Nash Bargaining Solution, and the Maximum Social Welfare Solution.

Nash proved that, under certain assumptions, an optimal agreement between two negotiators, is one that maximizes the product of the agents' utilities [3]. This is known as the *Nash bargaining solution* (NBS).

Definition 18. *The Nash bargaining solution is a set of offers $\mathcal{N} \subseteq \Omega$ defined as:*

$$\mathcal{N} := \arg\max_{\omega \in \Omega}\{(u_1(\omega) - rv_1) \cdot (u_2(\omega) - rv_2)\} \tag{6}$$

However, one of Nash's assumptions is that the offer space is a convex set, which is not the case for the ANAC domains, which have discrete offer spaces, so the NBS may not be the right solution concept for the ANAC domains. Another consequence of the non-convexity of the ANAC domains, is that the NBS is not always a single offer, but rather a *set* of offers.

It was therefore shown in [5] that in a tournament setting with finite domains, two optimal negotiation strategies would in fact come to the agreement that maximizes the *sum* of the agents' utilities, rather than the product. We will here refer to this as the maximum social welfare solution (MSWS). Note that, just like the NBS, the MSWS may return a *set* of offers, rather than a single offer.

Definition 19. *The maximum social welfare solution is a set of offers $\mathcal{SW} \subseteq \Omega$ defined as:*

$$\mathcal{SW} := \arg\max_{\omega \in \Omega}\{u_1(\omega) + u_2(\omega)\} \tag{7}$$

Since we do not want to get into the discussion here which of these solution concepts is better, we will consider both of them. Furthermore, we will see that in most of the ANAC domains the two concepts coincide anyway.

8.2 Balanced Negotiation Domains

We now define the notion of a *balanced* negotiation domain. On such domains, any agreement made by two MiCRO agents is always an optimal one. So, two MiCRO agents are, *by definition*, optimal on such domains. Surprisingly, it turns out that many of the ANAC domains indeed happen to be balanced, or close to balanced, even though we do not see any obvious reason why this would have to be true.

Definition 20. *Given some negotiation domain D, let $\mathcal{SC} \subseteq \Omega$ be some set of Pareto-optimal offers that are considered 'optimal' (according to some solution concept) and let $\mathcal{B}$ denote the balance set of D. Then we say the domain is* ***balanced*** *with respect to $\mathcal{SC}$, if and only if $\mathcal{B} \subseteq \mathcal{SC}$.*

In other words, a domain is balanced if any agreement made between two MiCRO agents is in the set of optimal agreements $\mathcal{SC}$.

The Ultimatum domain, displayed in Figures 1 and 2 is a nice example of a domain that is balanced w.r.t. the NBS, as well as w.r.t. the MSWS (in fact, the NBS and the MSWS coincide in this domain).

For those domains that are not balanced, we still want to know how close they are to being balanced, so we want to define a quantity $bs_{\mathcal{SC}}$, which we call the **balance score**, that measures this. To explain how we calculate it, first suppose there is exactly one optimal solution ω^* and the balance set contains exactly one element ω. Then, note that for any agent α_i, the quantity $u_i(\omega^*) - u_i(\omega)$ represents how much that agent loses from making agreement ω instead of the optimal agreement ω^*. Furthermore, since ω^* must be Pareto optimal, and ω is weakly Pareto optimal (by Lemma 4), this value can only be positive for one of the two agents. So, we calculate $bs_{\mathcal{SC}}$ as the loss for that agent. That is, $bs_{\mathcal{SC}} = \max_{i\in\{1,2\}}\{u_i(\omega^*) - u_i(\omega)\}$. If $bs_{\mathcal{SC}} = 0$ it means ω is itself optimal, so the domain is balanced, while a higher value of $bs_{\mathcal{SC}}$ means that ω is farther away from the optimal solution, so the domain is less balanced.

In case there are multiple optimal offers $\omega^* \in \mathcal{SC}$ or multiple offers $\omega \in \mathcal{B}$ in the balance set, then we can generalize this as follows:

$$bs_{\mathcal{SC}} := \max_{\omega\in\mathcal{B}} \min_{\omega^*\in\mathcal{SC}} \max_{i\in\{1,2\}} \{u_i(\omega^*) - u_i(\omega)\} \tag{8}$$

Here, the max and min operators are chosen such that we have $bs_{\mathcal{SC}} = 0$ if and only if the domain is balanced.

We have calculated the balance scores of the 35 domains[7] that were used in ANAC 2012–2013, with respect to the NBS ($bs_{\mathcal{N}}$) and to the MSWS ($bs_{\mathcal{SW}}$). They are displayed them in Table 4. These domains are normalized so that for each agent the worst possible offer always has utility value 0.0 and the best possible offer always has utility value 1.0. We see that for 29 of these domains we have $bs_{\mathcal{N}} \leq 0.1$, while for 20 of them we have $bs_{\mathcal{N}} \leq 0.05$, and for 14 of them we even have $bs_{\mathcal{N}} = 0.0$, so indeed in these domains the elements of the balance set often lie very close to the NBS. Similarly, if we use the MSWS as the reference optimal solution, then we find that for 25 of these domains we have $bs_{\mathcal{SW}} \leq 0.1$, while for 18 of them we have $bs_{\mathcal{SW}} \leq 0.05$, and for 12

[7] With their reservation values set to 0.

Domain	$\lvert\Omega\rvert$	b	Balance Set	NBS	MSWS	$bs_{\mathcal{N}}$	$bs_{\mathcal{SW}}$
Acquisition	384	50	(0.91 , 0.91)	(0.94 , 0.90)	(0.94 , 0.90)†	0.03	0.03
AirportSiteSelection	420	36	(0.77 , 0.81)	(1.00 , 0.67)	(1.00 , 0.67)	0.23	0.23
Amsterdam	3024	128	(0.83 , 0.86)	(0.83 , 0.86)	(0.83 , 0.86)	0.00	0.00
Barbecue	1440	112	(0.82 , 0.83)	(0.80 , 0.87)	(0.80 , 0.87)†	0.04	0.04
Barter	80	34	(0.65 , 0.65)	(0.65 , 0.65)	(0.47 , 0.85)	0.00	0.20
Camera	3600	82	(0.82 , 0.89)	(0.82 , 0.89)	(0.82 , 0.89)	0.00	0.00
Car	15625	2443	(0.93 , 0.94)	(0.94 , 0.93)	(0.94 , 0.93)	0.01	0.01
Energy	390625	66489	(0.63 , 0.62)	(0.65 , 0.61)	(0.78 , 0.48)	0.02	0.15
EnergySmall	15625	2692	(0.71 , 0.64)	(0.68 , 0.71)	(0.68 , 0.71)	0.07	0.07
EnglandvsZimbabwe	576	52	(0.82 , 0.79)	(0.91 , 0.73)	(0.91 , 0.73)	0.09	0.09
Fitness	3520	293	(0.82 , 0.79)	(0.77 , 0.84)	(0.77 , 0.84)	0.05	0.05
FiftyFifty	11	6	(0.50 , 0.50)	(0.50 , 0.50)	(0.50 , 0.50)†	0.00	0.00
FlightBooking	48	14	(0.74 , 0.90)	(0.74 , 0.90)	(0.74 , 0.90)	0.00	0.00
Grocery	1600	21	(0.84 , 0.90)	(0.84 , 0.90)	(0.84 , 0.90)	0.00	0.00
HouseKeeping	384	64	(0.78 , 0.83)	(0.84 , 0.78)	(0.94 , 0.69)	0.06	0.16
ItexvsCypress	180	35	(0.63 , 0.71)	(0.72 , 0.67)	(0.72 , 0.67)	0.10	0.10
Laptop	27	4	(0.87 , 0.87)	(1.00 , 0.82)	(1.00 , 0.82)	0.13	0.13
MusicCollection	4320	65	(0.89 , 0.89)	(0.94 , 0.86)	(0.94 , 0.86)	0.05	0.05
NiceOrDie	3	2	(0.30 , 0.30)	(1.00 , 0.16)†	(1.00 , 0.16)†	0.70	0.70
Outfit	128	12	(0.86 , 0.86)	(0.80 , 0.94)	(0.74 , 1.00)	0.08	0.14
Phone	1600	21	(0.84 , 0.89)	(0.84 , 0.89)	(0.84 , 0.89)	0.00	0.00
RentalHouse	60	21	(0.80 , 0.74)	(0.80 , 0.74)	(0.80 , 0.74)†	0.00	0.00
Supermarket	112896	5174	(0.79 , 0.71)	(0.86 , 0.68)	(0.86 , 0.68)†	0.07	0.07
Travel	188160	5425	(0.88 , 0.80)	(0.88 , 0.80)	(0.92 , 0.76)	0.00	0.04
Animal	1152	28	(0.91 , 0.94)	(0.91 , 0.94)	(0.91 , 0.94)	0.00	0.00
Coffee	112	33	(0.64 , 0.67)	(0.92 , 0.56)	(0.92 , 0.56)	0.28	0.28
DefensiveCharms	36	8	(0.78 , 0.77)	(0.72 , 0.84)	(0.72 , 0.84)	0.07	0.07
DogChoosing	270	16	(0.99 , 0.95)	(0.99 , 0.95)	(0.99 , 0.95)	0.00	0.00
Icecream	720	42	(0.89 , 0.90)	(0.89 , 0.90)	(0.89 , 0.90)	0.00	0.00
Kitchen-domain	15625	1816	(0.95 , 0.96)	(0.95 , 0.98)	(0.95 , 0.98)	0.02	0.02
Lunch	3840	692	(0.72 , 0.69)	(0.83 , 0.63)	(0.86 , 0.60)	0.11	0.14
planes	27	3	(0.88 , 0.88)	(1.00 , 0.82)	(1.00 , 0.82)	0.12	0.12
SmartPhone	12000	139	(0.84 , 0.83)	(0.89 , 0.80)	(0.89 , 0.80)	0.05	0.05
Ultimatum	9	3	(0.61 , 0.62)	(0.61 , 0.62)	(0.61 , 0.62)	0.00	0.00
Wholesaler-domain	56700	739	(0.78 , 0.79)	(0.78 , 0.79)	(0.78 , 0.79)	0.00	0.00

Table 4: The balance scores of the ANAC domains. For each of these domains the balance set contains just one element. The cases where $\mathcal{NBS}$ or $\mathcal{MSWS}$ contain more than one element are are marked with a dagger. In those cases, the offer shown in the table is the one selected by the max or min operators in Eq. (8).

of them we even have $bs_{\mathcal{SW}} = 0.0$. So, we conclude that the elements of the balance set also often lie very close to the MSWS.

8.3 Why Are So Many ANAC Domains Balanced?

In this section we will try to answer the question why so many of the ANAC domains are (close to) balanced. The following proposition may give some intuition about this. Specifically, it tells us that it is related to a kind of symmetry.

Proposition 2. *Let ω^* be some offer, and let $x^* := u_1(\omega^*)$ and $y^* := u_2(\omega^*)$. Then, if ω^* is Pareto-optimal and $|\Omega_1^{x^*}| = |\Omega_2^{y^*}|$, we have that x^* and y^* are exactly the balance values.*

Domain Type	$bs_{\mathcal{N}} \leq 0.10$	$bs_{\mathcal{N}} \leq 0.05$	$bs_{\mathcal{N}} = 0.0$
ANAC 2012/2013 (handcrafted)	29/35 (83%)	20/35 (57%)	14/35 (40%)
Genius Web (randomly generated)	32/50 (64%)	24/50 (48%)	17/50 (34%)
Nestlé/Pladis (real-world)	6/10 (60%)	3/10 (30%)	1/10 (10%)

Table 5: This table displays, for each type of domain, in how many cases the balance score bs_N was smaller than or equal to 0.10, 0.05 or 0.0, respectively. We see that the handcrafted domains are more balanced than the randomly generated domains, while the randomly generated domains, in turn, are more balanced than the real-world domains.

Domain Type	$bs_{\mathcal{SW}} \leq 0.10$	$bs_{\mathcal{SW}} \leq 0.05$	$bs_{\mathcal{SW}} = 0.0$
ANAC 2012/2013 (handcrafted)	25/35 (71%)	18/35 (51%)	12/35 (34%)
Genius Web (randomly generated)	27/50 (54%)	20/50 (40%)	12/50 (24%)
Nestlé/Pladis (real-world)	2/10 (20%)	1/10 (10%)	0/10 (0%)

Table 6: This table displays, for each type of domain, in how many cases the balance score bs_{SW} was smaller than or equal to 0.10, 0.05 or 0.0, respectively. Again, we see that the handcrafted domains are more balanced than the randomly generated domains, while the randomly generated domains, in turn, are more balanced than the real-world domains.

Proof. Let $(\omega_1, \omega_2, \dots \omega_K)$ be a list containing all offers in the domain, such that $u_1(\omega_1) \geq u_1(\omega_2) \geq \dots \geq u_1(\omega_K)$, and such that among all offers ω for which $u_1(\omega) = u_1(\omega^*)$, ω^* appears last in the list. We let x_i be shorthand for $u_1(\omega_i)$, so we have:

$$x_1 \geq x_2 \geq \dots \geq x_k \tag{9}$$

Then, if j is the position of ω^* in this list, we have $x_j = x^*$ and $|\Omega_1^{x^*}| = |\Omega_1^{x_j}| = j$. Similarly, we can sort the orders according to α_2's utility. If j' is the position of ω^* in that list, then we get $y_{j'} = y^*$ and $|\Omega_2^{y^*}| = |\Omega_2^{y_{j'}}| = j'$, but we have by assumption that $|\Omega_1^{x^*}| = |\Omega_2^{y^*}|$, so $j = j'$. We therefore have that $\omega^* \in \Omega_1^{x_j} \cap \Omega_2^{y_j}$.

Now, let k be the smallest integer for which $\omega^* \in \Omega_1^{x_k} \cap \Omega_2^{y_k}$. Clearly, $k \leq j$, which, by (9), implies $x_k \geq x_j$. Furthermore, since $\omega^* \in \Omega_1^{x_k}$ we have $x^* \geq x_k$, and since $x^* = x_j$ we conclude that $x_k = x_j = x^*$. Similarly we can show that $y_k = y_j = y^*$.

We now want to show that $\Omega_1^{x_k} \cap \Omega_2^{y_k}$ is the balance set. To do this, we have to prove that $\Omega_1^{x_{k-1}} \cap \Omega_2^{y_{k-1}}$ is empty. Suppose the contrary. Then there is some offer $\omega \in \Omega_1^{x_{k-1}} \cap \Omega_2^{y_{k-1}}$, which means that $u_1(\omega) \geq x_{k-1} \geq x_k = x^* = u_1(\omega^*)$ and $u_2(\omega) \geq y_{k-1} \geq y_k = y^* = u_2(\omega^*)$. Now, if one of these inequalities is strict, then we have a contradiction, because it would mean that ω dominates ω^*, while ω^* was assumed to be Pareto-optimal. On the other hand, if all these inequalities are actually equalities, then we also have a contradiction, because then we would have $x_{k-1} = x^*$ and $y_{k-1} = y^*$ which would imply $\omega^* \in \Omega_1^{x_{k-1}} \cap \Omega_2^{y_{k-1}}$, which contradicts our definition of k as the smallest integer for which $\omega^* \in \Omega_1^{x_k} \cap \Omega_2^{y_k}$.

Finally, since $x^* = x_k$ and $y^* = y_k$, and since x_k and y_k are by definition the lowest possible utility values any offer in $\Omega_1^{x_k} \cap \Omega_2^{y_k}$ can have, and since we have showed that $\Omega_1^{x_k} \cap \Omega_2^{y_k}$ is the balance set, we conclude that x^* and y^* are indeed the balance values. □

Note that if ω^* is in the NBS or MSWS (or any other solution that could be considered 'optimal') then it is indeed Pareto-optimal. So, if the domain happens to satisfy the symmetry condition $|\Omega_1^{x^*}| = |\Omega_2^{y^*}|$, then the utility values of this optimal solution coincide exactly with the balance values.

We can see in the right-hand image of Figure 2 that this indeed holds for the Ultimatum domain. If we draw a vertical line through the NBS, then there are exactly 3 offers on or to the right of this line, so we have $|\Omega_1^{x^*}| = 3$. Similarly, if we draw a horizontal line through the NBS, then there are exactly 3 offers on or above this line, so we have $|\Omega_2^{y^*}| = 3$. This means we have $|\Omega_1^{x^*}| = |\Omega_2^{y^*}|$, so, according to Proposition 2, the balance values should coincide with the NBS. Indeed, we see in the center image of Figure 2 that $\Omega_1^{x_2} \cap \Omega_2^{y_2} = \emptyset$, while we can see in the right-hand image that $\Omega_1^{x_3} \cap \Omega_2^{y_3} \neq \emptyset$. Therefore, $\Omega_1^{x_3} \cap \Omega_2^{y_3}$ is the balance set and it contains exactly one element, which happens to be the NBS. So, indeed, the balance values coincide with the NBS.

Of course, this still begs the question why exactly these domains are so symmetrical. One thing to note is that these domains were handcrafted by people [8, 34], so one hypothesis is that the symmetry is caused by the fact that the people who created them had a (possibly subconscious) preference for symmetry. Another hypothesis, is that this symmetry may emerge naturally when linear domains are created at random (similar to how a sum of multiple random variables naturally converges towards a Gaussian distribution).

To investigate this, we also analyzed two other types of domains: a set of randomly generated negotiation domains, and a set of negotiation domains obtained from a real-world industrial problem. For the randomly generated domains we use the 50 domains from the GeniusWeb framework [35] that were used in ANAC 2021, while for the real-world domains we use the Nestlé/Pladis domains from [36]. These represent the negotiation between two logistics companies about the exchange of truck loads, and were generated from real-world data.[8] The results of this analysis are displayed in Tables 5 and 6.

We see in Table 5 that for the handcrafted domains in 83% of the cases (i.e. in 29 out of the 35 domains) the balance score w.r.t. the NBS was smaller than or equal to 0.10, while for the randomly generated domains this was only 64%, and for the real-world problems this was only 60%. Similarly, if we use $s_{\mathcal{N}} \leq 0.05$ or $s_{\mathcal{N}} = 0.0$ as the criterion, then we also see that the handcrafted domains are more balanced than the random domains, while the random domains are in turn more balanced than the real-world domains. Furthermore, we can make the same observations from Table 6, which uses $s_{\mathcal{SW}}$ instead of $s_{\mathcal{N}}$.

These numbers suggest that both the hypothesized effects may actually be playing a role. Furthermore, it is striking to see that even the real-world domains still seem to display a relatively large degree of balancedness (at least when using $s_{\mathcal{N}}$ as its

[8] This data set can be downloaded from https://www.iiia.csic.es/~davedejonge/downloads. Each instance in this data set has two versions. We used the versions that only contained individually rational deals.

measure), which again suggests that a there may be some natural mathematical reason for this type of symmetry.

8.4 Negotiations with Discount Factors

As mentioned earlier, we did not take the presence of discount factors into account in our analysis. In theory, if there are discount factors, there could be strategies that are better than MiCRO. For example, two agents that immediately propose and accept an optimal solution would be better. However, we do not see how such a strategy could be implemented without knowledge of the opponent's utility function.

Furthermore, while we do not have any theoretical guarantees about the optimality of MiCRO in negotiations with discount factors, we know from experiments that two MiCRO agents typically come to a deal much faster than any other pair of agents [1]. This is because MiCRO does not have to do any calculations to update any opponent models. Therefore, in practice, the presence of discount factors only seems to increase the advantage of MiCRO over other strategies.

9 Inconsistent Strategies

In the previous sections we have always assumed that the opponents are (semi-)consistent. In general, this seems like a reasonable assumption. However, it is possible that an opponent could behave inconsistently on purpose because it would allow them to exploit MiCRO. We will here give an example of such a strategy, which we call *Anti-MiCRO*. However, it should be noted that in order to implement this strategy deliberately, the opponent would need to know MiCRO's utility function. Furthermore, we will show that we only need to make a minor adaptation to MiCRO to defend it against Anti-MiCRO.

9.1 The Anti-MiCRO Strategy

Suppose that α_1 applies MiCRO, and that the domain has K offers, which all have a different utility value for agent α_1. That is, we have:

$$u_1(\omega_1) > u_1(\omega_2) > \cdots > u_1(\omega_K)$$

Then, there exists an inconsistent strategy for α_2, which we call *Anti-MiCRO*, that, for any offer $\omega \in \Omega \setminus \{\omega_K\}$, can force MiCRO to propose ω. It works as follows. If α_2 has to make the first proposal of the negotiation, then it will propose ω_2. Otherwise, whenever α_1 proposes some offer ω_j, then α_2 will reply by proposing ω_{j+2}, unless α_1 proposes the desired offer ω, in which case α_2 will accept it.

An example of a negotiation between MiCRO and Anti-MiCRO is displayed in Table 7, in which Anti-MiCRO wants to force MiCRO to propose ω_5.

We can see that this strategy works, because when Anti-MiCRO plays against MiCRO, it will never repeat any proposals, and therefore MiCRO will also keep making new proposals. Furthermore, MiCRO will never accept any of α_2's proposals, because when α_2 proposes ω_j, agent α_1 will have made in total $j - 2$ different proposals, and

	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$	$t = 6$	$t = 7$	$t = 8$	$t = 9$	$t = 10$
MiCRO	ω_1		ω_2		ω_3		ω_4		ω_5	
Anti-MiCRO		ω_3		ω_4		ω_5		ω_6		accept: ω_5

Table 7: MiCRO vs. Anti-MiCRO. Note that at $t = 6$, Anti-MiCRO makes an inconsistent rejection, because it rejects ω_3 while Anti-MiCRO itself already proposed ω_3 earlier.

therefore MiCRO will only be willing to propose or accept ω_{j-1}, at best. This means that MiCRO keeps conceding, and therefore will eventually propose the offer desired by α_2.

It is easy to see that Anti-MiCRO is inconsistent, because for any offer ω_j, if α_2 proposes it, then MiCRO will re-propose it three turns later, but then α_2 will reject it (unless it is the desired offer), which is an inconsistent rejection. In fact, it is a *detectable* inconsistent rejection, which we will define below.

However, there are three problems with Anti-MiCRO that make it unsuitable in practice:

1. It requires α_2 to know the exact preference ordering of α_1.
2. It is an extremely risky strategy, because some the offers proposed by α_2 may actually be very bad for α_2. So, if α_1 is actually not applying MiCRO, but instead some other strategy, then it could happen that α_1 unexpectedly accepts one of those bad proposals.
3. With just a small adaptation, MiCRO can easily defend itself against Anti-MiCRO, as we will see next.

9.2 Defending against Anti-MiCRO

We now propose three possible ways to adapt MiCRO that allow it to defend itself against strategies like Anti-MiCRO. The idea behind each of them, is to take repercussions once it detects the opponent makes an inconsistent rejection.

Definition 21. *We say an agent makes a* ***detectable*** *inconsistent rejection, if it first proposes some offer ω, and then later rejects that same offer.*

Note that this definition is indeed a special case of an inconsistent rejection (See Def. 13) with $\omega = \omega'$. It is called *detectable* because in this case our agent can see that the opponent's rejection is inconsistent, even without knowing the opponent's utility function. On the other hand, it is impossible for an agent to know that its opponent is making any other kind of inconsistent action without knowing its utility function. For this reason, our adaptations to MiCRO only involve detectable inconsistent rejections. Luckily, this is enough to defend against Anti-MiCRO.

It should be noted that all the previously stated lemmas and theorems about MiCRO that assume the opponent is (semi-)consistent, continue to hold for the adaptations of MiCRO that we propose here. This is because all of these adaptations behave exactly the same as the original MiCRO when the opponent is (semi-)consistent.

9.2.1 MiCRO-Retire

The simplest adaptation is the following: whenever the opponent makes a detectable inconsistent rejection, immediately break off the negotiations. We call this MiCRO-Retire (RETreat after Inconsistent REjection). While this approach is rather harsh, it allows us to easily prove an even stronger version of Theorems 3 and 4.

Theorem 5. *Suppose the same conditions as Theorem 3. Then, among all negotiation strategies that never make inconsistent proposals and never make undectectable inconsistent rejections, MiCRO-Retire is a best response against itself.*

Proof. If α_1 and α_2 both apply MiCRO-Retire then the outcome would be the same as when they both applied MiCRO. Now, suppose α_2 deviates to some other strategy σ that never makes inconsistent proposals or undetectable inconsistent rejections. If σ also does not make any detectable inconsistent rejections, then σ is in fact semi-consistent and MiCRO-Retire will behave exactly like MiCRO. Therefore, we know by Theorem 4 that σ cannot be a better response. On the other hand, if, at some point, σ does make a detectable inconsistent rejection, then α_1 will withdraw and the negotiations will fail, so α_2 will be worse off than if it had chosen MiCRO-Retire. So, again, σ would not be a better response. □

9.2.2 MiCRO-Repair and MiCRO-Inspire

Withdrawing from the negotiations immediately after an inconsistent rejection may be an overly harsh reaction, since some opponents might make inconsistent rejections by accident (e.g. because of bugs in the code, or merely by suboptimal implementation), or on purpose, but for reasons that have nothing to do with exploiting MiCRO. We therefore also propose two more forgiving adaptations.

The first option is called *MiCRO-Repair* (REpeat old Proposals After Inconsistent Rejection). Whenever it detects an inconsistent rejection by the opponent, it does continue negotiating, but it will not propose any *new* offers anymore. It will just keep repeating (random) proposals it already made before, until the end of the negotiations.

The second option is even more forgiving. We call it *MiCRO-Inspire* (Ignore iNSincere Proposals that were Inconsistently REjected). Define the numbers m and n as in Section 3.4. Furthermore, define l as the number of unique offers that were proposed by the opponent, but then later rejected by that opponent. Then, we obtain MiCRO-Inspire by replacing every occurrence of n in Algorithm 1, by $n - l$. That is, in any turn, MiCRO-Inspire will propose a new offer only if $m \leq n - l$. Otherwise it will just repeat an earlier proposal. The idea behind this, is that if the opponent first proposes a given offer, but later rejects it, then it was not a genuine proposal, and therefore it should not be counted.

We suspect that Theorem 5 still holds if we use MiCRO-Repair or MiCRO-Inspire instead of MiCRO-Retire. We will not attempt to prove this, however.

10 Conclusions and Future Work

We have defined the notion of a consistent negotiation strategy, and have shown that the MiCRO negotiation strategy forms a best response against itself among consistent negotiation strategies. Furthermore, we have defined the notion of a balanced negotiation domain, and we have shown that the MiCRO strategy is a game-theoretically optimal strategy on such domains. By itself, this is not surprising, because the notion of a balanced domain was defined exactly with that purpose. However, what is surprising, is that in practice many of the ANAC domains turn out to be balanced.

Although it is not entirely clear why so many of these domains are balanced, we have provided evidence that suggests this may be caused by two different effects. Firstly, it may be partially due to the fact that they were designed by people, who apparently have a (possibly subconscious) preference for domains that exhibit some degree of symmetry. And secondly, it may also partially be the natural consequence of some yet unknown mathematical mechanism. Anyway, we think that understanding the concept of a balanced negotiation domain will help researchers to design more challenging negotiation test cases in the future.

In summary, we draw the following main conclusions:

1. Under the conditions of ANAC, MiCRO is a best response against itself among all consistent strategies.
2. Many domains used in ANAC are approximately balanced, meaning that, among all consistent strategies, MiCRO is game-theoretically optimal on such domains.
3. While there are inconsistent strategies that are able to exploit MiCRO, as far as we can tell such strategies always need very precise knowledge of their opponent's utility function, and be absolutely sure that their opponent is applying MiCRO.

One important open question is to what extent realistic negotiation strategies are indeed consistent. While we argued that without knowledge of opponent utility, they should indeed be consistent, we know that in reality agents may use opponent modeling algorithms to obtain such knowledge. If such algorithms are good enough they could be used to implement agents that are inconsistent on purpose, so as to exploit MiCRO. We therefore think that MiCRO can be very useful as a tool to assess the quality of such opponent modeling algorithms. Specifically, it can be used to answer the following questions:

- Are opponent modeling algorithms strong enough to successfully implement Anti-MiCRO (or some other inconsistent strategy that exploits MiCRO)?
- If yes, can MiCRO itself also make use of such opponent modeling algorithms to defend itself against such opponents (e.g. by detecting inconsistent proposals or rejections from the opponent and then taking repercussions)?
- If a learning algorithm negotiates against MiCRO, on a balanced domain, will it learn to behave like MiCRO?
- If two learning algorithms negotiate against each other on a balanced domain, will they learn to behave like MiCRO?

Furthermore, we are planning to develop generalizations of MiCRO for multilateral negotiations, for negotiations over larger domains in which there is not enough time to propose all offers one by one (e.g. domains with more than a million offers), and for negotiations over extremely large domains in which there is not even enough time to sort the offers in a list (e.g. domains with 10^{20} offers).

11 Declarations

Funding This work was supported by a Juan de la Cierva - Incorporación research grant from the Spanish Ministry of Science and Innovation (IJC2018-036443-I) and by Grant no. TED2021-131295B-C31 funded by MCIN/AEI /10.13039/501100011033 and by the European Union NextGenerationEU/PRTR.

Conflicts of interest/Competing interests The author has no conflicts of interest to declare that are relevant to the content of this article.

Data and Code Availability The datasets analyzed in this paper are publicly available with the Genius framework: https://ii.tudelft.nl/genius/

The source code of MiCRO and the code to determine the balance set of a negotiation domain are publicly available at: https://www.iiia.csic.es/~davedejonge/downloads

Appendix A Proof that MiCRO is consistent

We here provide the proof that MiCRO is consistent. That is, we will prove that that it never makes inconsistent rejections, acceptances, or proposals. We prove each of these separately. In all cases we will suppose that it is agent α_1 that applies the MiCRO strategy.

Lemma 6. *MiCRO never makes inconsistent rejections.*

Proof. Suppose there exist ω, $\omega' \in \Omega$ such that the second and third condition of Def. 13 hold. We will show that the first condition then cannot hold. Let ω_{low} denote the offer with lowest utility that α_1 is willing to propose at time t, as defined by Eq. (1). Now, since the third condition says that at some time t' before t agent α_1 has proposed ω', we know that, by definition of MiCRO, we must have $u_1(\omega') \geq u_1(\omega_{low})$. But, according to the second condition, we have $u_1(\omega) \geq u_1(\omega')$, and therefore we also have $u_1(\omega) \geq u_1(\omega_{low})$. As explained in Section 3.4, this means that α_1 will accept ω, instead of rejecting it. So indeed, the first condition fails to hold. □

Lemma 7. *MiCRO never makes inconsistent acceptances.*

Proof. We assume that all four conditions of Def. 12 hold, and show that this leads to a contradiction. Note that the third condition implies that ω' appears on MiCRO's sorted list before ω, and therefore MiCRO must have proposed ω' before accepting ω, at some time t''. So, we have $(i, \mathfrak{p}, \omega', t'') \in s$, with $t'' < t$. Furthermore, the first two conditions say we have $(i, \mathfrak{a}, \omega, t) \in s$ and $(3 - i, \mathfrak{p}, \omega', t') \in s$, with $t' < t$. There are now two possibilities: either $t' < t''$ or $t'' < t'$. In the first case, the fourth condition of Def. 12 is violated, so we must have $t'' < t'$. However, that would mean that MiCRO would accept ω' directly after t', so the negotiations would end with ω' as the accepted offer, violating the first condition. □

Lemma 8. *MiCRO never makes inconsistent proposals.*

Proof. Suppose the contrary. That is, suppose that at some time t agent α_1 proposes ω, while ω' has strict priority over ω for α_1.

This means that the opponent has already proposed ω' at some earlier time t' before t.

We will first show that MiCRO must have also proposed ω' at some time before t. To do this, we will consider two separate cases, namely $u_1(\omega') > u_i(\omega)$, and $u_1(\omega') = u_i(\omega)$ (the case that $u_1(\omega') < u_i(\omega)$ is not possible because we assumed that ω' had strict priority over ω).

In the case that $u_1(\omega') > u_i(\omega)$ the fact that α_1 would have proposed ω' before proposing ω follows immediately from the definition of MiCRO.

In the case that $u_1(\omega') = u_i(\omega)$, it follows from Conditions 5 and 7 of Def. 10 that the opponent must have already proposed ω', while it has not yet proposed ω. When that happens, MiCRO makes sure that ω' comes before ω on its list (see Algorithm 1, lines 13–22), so in that case α_1 would also propose ω' before ω.

Now that we have established that there is some time t'' at which α_1 proposed ω', we see from Conditions 3 and 6 of Def. 10 that this cannot have happened after time t'.

So, we know that α_1 first proposed ω' and then at a later time t' the opponent α_2 re-proposed it. But then, by definition of MiCRO, α_1 would have accepted that α_2's proposal at time t', so the negotiations would have finished before time t. This contradicts our assumption.

□

Appendix B More Properties of the Balance Set

In this appendix we discuss a number of properties of the balance set. These are not directly relevant to the rest of the paper, but they may help to get a better understanding of the concept.

We have already seen that when two MiCRO agents negotiate with each other, the outcome will be an element of the balance set. We here show, however, that we cannot say more than that, because the balance set may contain multiple offers and the question which of those the agents will agree upon depends on the details of the negotiation protocol and on the way the agents break ties whenever they are indifferent between offers.

We will use the notation $\omega \sim (40, 60)$ to indicate that for offer ω agent α_1 a utility value of 40, while agent α_2 receives a utility value of 60, i.e. $u_1(\omega) = 40$ and $u_2(\omega) = 60$.

Observation 1. *The balance set of a negotiation domain may contain more than one element.*

We can demonstrate this observation with the following example:

$$\omega_1 \sim (100, 25), \quad \omega_2 \sim (75, 50), \quad \omega_3 \sim (50, 75), \quad \omega_4 \sim (25, 100)$$

Then we have:

$$\begin{array}{llll} \Omega_1^{x_1} = \{\omega_1\} & \text{and} \quad \Omega_2^{y_1} = \{\omega_4\}, & \text{so:} & \Omega_1^{x_1} \cap \Omega_2^{y_1} = \emptyset \\ \Omega_1^{x_2} = \{\omega_1, \omega_2\} & \text{and} \quad \Omega_2^{y_2} = \{\omega_3, \omega_4\}, & \text{so:} & \Omega_1^{x_2} \cap \Omega_2^{y_2} = \emptyset \end{array}$$

$$\Omega_1^{x_3} = \{\omega_1, \omega_2, \omega_3\} \quad \text{and} \quad \Omega_2^{y_3} = \{\omega_2, \omega_3, \omega_4\}, \quad \text{so:} \quad \Omega_1^{x_3} \cap \Omega_2^{y_3} = \{\omega_2, \omega_3\}$$

So, indeed, we see that $\Omega_1^{x_3} \cap \Omega_2^{y_3}$ is the balance set and it has more than one element.

Observation 2. *If a balance set has more than one element, then the outcome of a negotiation between two agents applying MiCRO may depend on which agent makes the first proposal.*

Look again at the example of Observation 1. If agent α_1 starts, then they agree on offer ω_3, while if agent α_2 starts, then the agreement will be ω_2.

Observation 3. *If there is at least one agent that is indifferent between at least two offers, then the balance set may contain more than two elements.*

Consider a domain with the following offers (note that α_2 is now indifferent between ω_1 and ω_2):

$$\omega_1 \sim (100, 25), \quad \omega_2 \sim (75, 25), \quad \omega_3 \sim (50, 75), \quad \omega_4 \sim (25, 100)$$

Then we have:

$$\begin{array}{llll}
\Omega_1^{x_1} = \{\omega_1\} & \text{and} \quad \Omega_2^{y_1} = \{\omega_4\}, & \text{so:} & \Omega_1^{x_1} \cap \Omega_2^{y_1} = \emptyset \\
\Omega_1^{x_2} = \{\omega_1, \omega_2\} & \text{and} \quad \Omega_2^{y_2} = \{\omega_3, \omega_4\}, & \text{so:} & \Omega_1^{x_2} \cap \Omega_2^{y_2} = \emptyset \\
\Omega_1^{x_3} = \{\omega_1, \omega_2, \omega_3\} & \text{and} \quad \Omega_2^{y_3} = \{\omega_1, \omega_2, \omega_3, \omega_4\}, & \text{so:} & \Omega_1^{x_3} \cap \Omega_2^{y_3} = \{\omega_1, \omega_2, \omega_3\}
\end{array}$$

So, again, $\Omega_1^{x_3} \cap \Omega_2^{y_3}$ is the balance set, but this time it has three elements.

Under the AOP, the outcome of a negotiation between two MiCRO agents is determined by who makes the first proposal. This means there are only two elements of the balance set that can actually be the outcome. However, if we allow a small adaptation to the AOP in which agents can accept any previously proposed offer (instead of only the last one), then the agents may also agree on other elements of the balance set.

References

[1] Jonge, D.: An analysis of the linear bilateral ANAC domains using the MiCRO benchmark strategy. In: Raedt, L.D. (ed.) Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, 23-29 July 2022, pp. 223–229. ijcai.org, Vienna, Austria (2022). https://doi.org/10.24963/ijcai.2022/32

[2] Faratin, P., Sierra, C., Jennings, N.R.: Negotiation decision functions for autonomous agents. Robotics and Autonomous Systems **24**(3-4), 159–182 (1998) https://doi.org/10.1016/S0921-8890(98)00029-3 . Multi-Agent Rationality

[3] Nash, J.F.: The bargaining problem. Econometrica **18**, 155–162 (1950)

[4] Rubinstein, A.: Perfect equilibrium in a bargaining model. Econometrica: Journal of the Econometric Society, 97–109 (1982)

[5] Jonge, D.d.: A new bargaining solution for finite offer spaces. Applied Intelligence (2023) https://doi.org/10.1007/s10489-023-05009-1

[6] Gatti, N.: Game theoretic models for strategic bargaining. In: Lopes, F., Coelho, H. (eds.) Negotiation and Argumentation in Multi-Agent Systems: Fundamentals, Theories, Systems and Applications, pp. 30–47. Bentham Science Publishers, - (2014). Chap. 2

[7] Baarslag, T., Hindriks, K.V., Jonker, C.M., Kraus, S., Lin, R.: The first automated negotiating agents competition (ANAC 2010). In: New Trends in Agent-Based Complex Automated Negotiations. Studies in Computational Intelligence, vol. 383, pp. 113–135. Springer, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-24696-8_7

[8] Gal, Y.K., Ilany, L.: The fourth automated negotiation competition. In: Next Frontier in Agent-Based Complex Automated Negotiation. Studies in Computational Intelligence, vol. 596, pp. 129–136. Springer, Tokyo (2015). https://doi.org/10.1007/978-4-431-55525-4_8

[9] Fujita, K., Aydogan, R., Baarslag, T., Ito, T., Jonker, C.M.: The fifth automated negotiating agents competition (ANAC 2014). In: Recent Advances in Agent-based Complex Automated Negotiation [revised and Extended Papers from the 7th International Workshop on Agent-based Complex Automated Negotiation, ACAN 2014, Paris, France, May 2014]. Studies in Computational Intelligence, vol. 638, pp. 211–224. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-30307-9_13

[10] Fujita, K., Aydoğan, R., Baarslag, T., Hindriks, K., Ito, T., Jonker, C.: The sixth automated negotiating agents competition (anac 2015). In: Modern Approaches to Agent-based Complex Automated Negotiation, pp. 139–151. Springer, Cham (2017)

[11] Aydogan, R., Baarslag, T., Fujita, K., Mell, J., Gratch, J., Jonge, D., Mohammad, Y., Nakadai, S., Morinaga, S., Osawa, H., Aranha, C., Jonker, C.: Challenges and main results of the automated negotiating agents competition (anac) 2019. In: Multi-Agent Systems and Agreement Technologies. 17th International Conference EUMAS 2020 and 7th International Conference AT 2020. Thessaloniki, Greece September 14-15, 2020. Revised Selected Papers. Springer, Cham (2020)

[12] Jonge, D., Baarslag, T., Aydoğan, R., Jonker, C., Fujita, K., Ito, T.: The challenge of negotiation in the game of diplomacy. In: Agreement Technologies, 6th International Conference, AT 2018, Bergen, Norway, December 6-7, 2018, Revised Selected Papers. Lecture Notes in Computer Science, vol. 11327, pp. 100–114. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17294-7_8

[13] Mell, J., Gratch, J., Baarslag, T., Aydogan, R., Jonker, C.M.: Results of the first annual human-agent league of the automated negotiating agents competition. In:

Proceedings of the 18th International Conference on Intelligent Virtual Agents, IVA 2018, Sydney, NSW, Australia, November 05-08, 2018, pp. 23–28. Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3267851.3267907

[14] Sengupta, A., Mohammad, Y., Nakadai, S.: An autonomous negotiating agent framework with reinforcement learning based strategies and adaptive strategy switching mechanism. In: Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '21, Online, May 5-7, 2021, pp. 1163–1172. International Foundation for Autonomous Agents and Multiagent Systems, - (2021)

[15] Mirzayi, S., Taghiyareh, F., Nassiri-Mofakham, F.: An opponent-adaptive strategy to increase utility and fairness in agents' negotiation. Applied Intelligence (2021) https://doi.org/10.1007/s10489-021-02638-2

[16] Bakker, J., Hammond, A., Bloembergen, D., Baarslag, T.: RLBOA: A modular reinforcement learning framework for autonomous negotiating agents. In: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019, pp. 260–268 (2019). http://dl.acm.org/citation.cfm?id=3331701

[17] Kawaguchi, S., Fujita, K., Ito, T.: Compromising strategy based on estimated maximum utility for automated negotiation agents competition (ANAC-10). In: Modern Approaches in Applied Intelligence - 24th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2011, Syracuse, NY, USA, June 28 - July 1, 2011, Proceedings, Part II. Lecture Notes in Computer Science, vol. 6704, pp. 501–510. Springer, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21827-9_51

[18] Williams, C.R., Robu, V., Gerding, E.H., Jennings, N.R.: Iamhaggler: A negotiation agent for complex environments. In: New Trends in Agent-Based Complex Automated Negotiations. Studies in Computational Intelligence, vol. 383, pp. 151–158. Springer, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-24696-8_10

[19] Williams, C.R., Robu, V., Gerding, E.H., Jennings, N.R.: Using gaussian processes to optimise concession in complex negotiations against unknown opponents. In: IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, July 16-22, 2011, pp. 432–438. IJCAI, Barcelona, Catalonia, Spain (2011). https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-080

[20] Chen, S., Weiss, G.: An efficient and adaptive approach to negotiation in complex environments. In: ECAI 2012 - 20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012) System Demonstrations Track, Montpellier, France, August 27-31

, 2012. Frontiers in Artificial Intelligence and Applications, vol. 242, pp. 228–233. IOS Press, Amsterdam, The Netherlands (2012). https://doi.org/10.3233/978-1-61499-098-7-228

[21] Hindriks, K.V., Tykhonov, D.: Opponent modelling in automated multi-issue negotiation using bayesian learning. In: 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), May 12-16, 2008, Volume 1, pp. 331–338. IFAAMAS, Estoril, Portugal (2008). https://dl.acm.org/citation.cfm?id=1402433

[22] Serban, L.D., Silaghi, G.C., Litan, C.M.: AgentFSEGA: Time constrained reasoning model for bilateral multi-issue negotiations. In: New Trends in Agent-Based Complex Automated Negotiations. Studies in Computational Intelligence, vol. 383, pp. 159–165. Springer, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-24696-8_11

[23] Baarslag, T., Hindriks, K.V., Jonker, C.M.: A tit for tat negotiation strategy for real-time bilateral negotiations. In: Complex Automated Negotiations: Theories, Models, and Software Competitions. Studies in Computational Intelligence, vol. 435, pp. 229–233. Springer, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-30737-9_18

[24] Krimpen, T., Looije, D., Hajizadeh, S.: Hardheaded. In: Complex Automated Negotiations: Theories, Models, and Software Competitions. Studies in Computational Intelligence, vol. 435, pp. 223–227. Springer, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-30737-9_17

[25] Frieder, A., Miller, G.: Value model agent: A novel preference profiler for negotiation with agents. In: Complex Automated Negotiations: Theories, Models, and Software Competitions. Studies in Computational Intelligence, vol. 435, pp. 199–203. Springer, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-30737-9_12

[26] Baarslag, T., Hendrikx, M.J.C., Hindriks, K.V., Jonker, C.M.: Learning about the opponent in automated bilateral negotiation: a comprehensive survey of opponent modeling techniques. Autonomous Agents Multi Agent Systems **30**(5), 849–898 (2016) https://doi.org/10.1007/s10458-015-9309-1

[27] Baarslag, T., Hadfi, R., Hindriks, K.V., Ito, T., Jonker, C.M.: Optimal non-adaptive concession strategies with incomplete information. In: Fukuta, N., Ito, T., Zhang, M., Fujita, K., Robu, V. (eds.) Recent Advances in Agent-based Complex Automated Negotiation [revised and Extended Papers from the 7th International Workshop on Agent-based Complex Automated Negotiation, ACAN 2014, Paris, France, May 2014]. Studies in Computational Intelligence, vol. 638, pp. 39–54. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-30307-9_3

[28] Baarslag, T., Gerding, E.H., Aydogan, R., Schraefel, M.M.C.: Optimal negotiation decision functions in time-sensitive domains. In: IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, WI-IAT 2015, December 6-9, 2015 - Volume II, pp. 190–197. IEEE Computer Society, Singapore (2015). https://doi.org/10.1109/WI-IAT.2015.161

[29] Mohammad, Y.: Optimal time-based strategy for automated negotiation. Appl. Intell. **53**(6), 6710–6735 (2023) https://doi.org/10.1007/S10489-022-03662-6

[30] Baarslag, T., Hindriks, K.V.: Accepting optimally in automated negotiation with incomplete information. In: Gini, M.L., Shehory, O., Ito, T., Jonker, C.M. (eds.) International Conference on Autonomous Agents and Multi-Agent Systems, AAMAS '13, May 6-10, 2013, pp. 715–722. IFAAMAS, Saint Paul, MN, USA (2013). http://dl.acm.org/citation.cfm?id=2485033

[31] Jonge, D., Zhang, D.: GDL as a unifying domain description language for declarative automated negotiation. Auton. Agents Multi Agent Syst. **35**(1), 13 (2021) https://doi.org/10.1007/s10458-020-09491-6

[32] Rosenschein, J.S., Zlotkin, G.: Rules of Encounter. The MIT Press, Cambridge, USA (1994)

[33] Renting, B., Jonge, D., Hoos, H., Jonker, C.: Analysis of Learning Agents in Automated Negotiation. Under review

[34] Williams, C.R., Robu, V., Gerding, E.H., Jennings, N.R.: An overview of the results and insights from the third automated negotiating agents competition (ANAC2012). In: Novel Insights in Agent-based Complex Automated Negotiation. Studies in Computational Intelligence, vol. 535, pp. 151–162. Springer, Tokyo (2014). https://doi.org/10.1007/978-4-431-54758-7_9

[35] Pasman, W., Murukannaiah, P.K., , Jonker, C.M.: GeniusWeb. https://ii.tudelft.nl/GeniusWeb/ Accessed 2022-05-03

[36] Jonge, D., Bistaffa, F., Levy, J.: Multi-objective vehicle routing with automated negotiation. Applied Intelligence (2022) https://doi.org/10.1007/s10489-022-03329-2