Weight-based negotiation mechanisms: Balancing personal utilities

Diemo Urbig

Department of Computer Science
Humboldt-Universität zu Berlin, Germany

Abstract. Conflict resolution, e.g. negotiation, is frequently about an interactive process that forces agents to make concessions in order to resolve the conflict. In multilateral negotiations, concessions might be directed to one or another partner. In isolated negotiations such directed concessions might be less useful, but may become important for interdependent negotiations. We present weight-based negotiation mechanisms that easily implement the concept of directed concessions. As an example we implement and simulate the weighted sum approach. We show that the presented class of negotiation mechanisms results in Pareto-optimal agreements. Not all weight-based mechanisms, especially the weighted sum approach, can generate all Pareto-optimal solutions, but for every discrete negotiation space there is a weight-based mechanism based on a continuous balancing function that can generate all Pareto-optimal solutions.

Keywords: negotiating agents, balanced personal utilities

1. Introduction

In multi-agent systems conflicts may occur between autonomous agents. Even if there is no direct conflict there may be situations where agents are better off by having some social coordination mechanism. Negotiation is a decentralized method for coordination and conflict resolution between autonomous agents [4]. In a narrow sense it is an interactive process that forces participants to make concessions, i.e. to step-wise withdraw from former conflict-raising claims [7]. Following this idea, concessions are a vital part of negotiation mechanisms. A negotiation mechanism includes a negotiation protocol and a negotiation strategy [9]. For example, the round-based monotonic concession protocol forces agents to make concessions [9, 13]. In fact, negotiations following this protocol end if either an agreement is reached or no participant has made a concession. For bilateral negotiations Rosenschein and Zlotkin implement concessions as the reduction of the agent’s own required utility [9]. Because they assume negotiation spaces
that only contain Pareto-optimal solutions, a concession is also related to the increase of the other agent’s utility. Wooldridge, Bussmann, and Klosterberg extend the bilateral monotonic concession protocol to enable multilateral negotiations [13]. They model real concessions as proposals, which compared to the previous proposal increase the utility of at least one partner while keeping all other agents excluding the agent itself at their current utility level or above. In multilateral negotiations this may prevent a complete search through the negotiation space. This means that despite there was a possible agreement that does not represent the conflict deal, the negotiation could stop without reaching it. Therefore, Wooldridge, Bussmann, and Klosterberg additionally introduce backtracking, which allows offering something that has not been offered before because it decreased at least one other agent’s utility while increasing another agent’s utility [13]. For the monotonic concession protocol it is shown that from a game-theoretical point of view the Zeuthen strategy is optimal.

The game-theoretic analysis that is applied in computer science (e.g. [9, 13]) usually assumes isolated negotiations, i.e. during a negotiation only the current negotiation space and the decisions done and planned in this negotiation matter. Contrary, in social systems and also in systems where agents are able to learn we find that negotiations are usually not isolated. Agents utilize their experiences from earlier negotiations, e.g. being able to favor or to discriminate other agents in following negotiations. Discriminating agents in a negotiation can imply that concessions are done in favor of a particular agent; in other words, agents should be able to make directed concessions. Negotiating agents that favor particular partners are also developed in [10], where agents support other agents that have helped to reach a beneficial agreement in a previous negotiation.

Let us take a closer look at the way concessions can be defined. In the Distributed Artificial Intelligence literature regarding game theory and negotiations, concessions are related to actions, more precisely to observable behavioral adaptations. Agents that make concessions are required to change their behavior by proposing something different [6, 9, 13]. We argue that concessions can also have a mental dimension. A negotiation partner might be willing to propose an outcome that is better for the other partner but decreases own utility. Despite the agent concedes mentally, it might stick to its old proposal because there are only possible outcomes that concede way too much. This delayed behavioral change may increase the partner’s incentive to find an alternative in between. After several rounds the agent might have mentally conceded to a point where it actually changes the proposal and does the big step. Wooldridge, Bussmann, and Klosterberg point out that such processes unnecessarily lengthen the negotiation [13]. We argue that such delayed concessions can be usefully applied by and to human negotiators, which are partners in hybrid systems as developed in the INKA project [5] and, perhaps more important, it can be applied for negotiations that also aim at a decentralized exploration of the negotiation space.

This article analyzes weight-based multilateral negotiation mechanisms, which balance negotiating agents’ utilities based on an adaptable private weight vector. The weight vector, which assigns a weight to every participant, allows for directed concessions. It is not required but for simplicity we will follow the idea of mental concessions; thus, we may loose several nice game-theoretic properties, e.g. incentive compatibility or the optimality of the Zeuthen strategy. We start with specifying the negotiation scenario as well as the agents’ internal reasoning processes. The most important parts of the internal reasoning processes are estimations of the partners’ weight vectors and utilities as well as an interactive utility

\[Z.S.:\] The agent with the smallest risk concedes; risk for an agent is defined as the utility it loses by accepting another agent’s offer divided by the utility it loses by not conceding and causing conflict, see also [9].
function that balances these partners’ utilities. In case of having a weighted sum as interactive utility function we show how agents can estimate the partner’s weight vector from the observed proposal. By running several simulations we explore some properties of the weighted-sum approach. Finally, we discuss Pareto-optimality of agreements reached by weight-based negotiation mechanisms. We present a class of weight-based negotiation mechanism, i.e. a parameterized modified $\rho$-norm, that allows determining a continuous interactive utility function that is able to generate all Pareto-optimal solutions as the outcome of a weight-based negotiation mechanism. Since these interactive utility functions differ only in one parameter, this parameter can also be used to characterize negotiation settings.

2. Weight-based negotiation mechanisms

Consider a group of agents who face a joint decision. There is a set of possible decision outcomes. Since agents assign different utilities to these outcomes there can be a conflict, i.e. different agents prefer different outcomes. Agents negotiate to solve these conflicts. Round by round all agents propose outcomes. An agreement is reached when they all propose the same outcome; then the negotiation ends.

We assume that all participants stay from the beginning until the very end of the negotiation. We also assume that there is one possible outcome that represents the conflict deal, i.e. an outcome that is chosen if all agents agree not to agree. We also do only allow for single-outcome proposals, i.e. an agent has to propose one and only one outcome.

Definition 2.1. (Negotiation group)
Let set $G = \{1, \ldots, m\}$ be a group of agents that are labeled with 1, 2, and so forth. The number of group members is given by $m$.

Definition 2.2. (Negotiation outcomes and negotiation opportunities)
Let $o \in \mathbb{R}^m$ be a possible outcome for a decision of a group $G$ of $m$ agents, where $o_i$ is the utility that group member $i$ assigns to this outcome. A negotiation opportunity $O$ is a set of possible outcomes: $O \subseteq \mathbb{R}^m$.

We assume our agents to be utility maximizing. Many articles on agent negotiations assume that agents correctly estimate the utilities of other agents. In fact, this knowledge about partners is applied to find out which outcome may be preferred by another agent. It is also applied to find out what another agent can lose when a different agreement is reached. That is the case for the well-known monotonic concession protocol with Zeuthen strategy [9, 13, 8]. In this article we adhere to this assumption and assume that agents can perfectly estimate other agents’ utilities.

Definition 2.3. (Proposals and Agreements)
Let $p : \mathbb{N} \rightarrow O^m$ be a function that assigns each negotiation round $t \in \mathbb{N}$ a vector of proposed outcomes. $p(t) = (p_1(t), \ldots, p_m(t))$ is the proposal vector of group $G = \{1, 2, \ldots, m\}$ that is negotiating opportunity $O$, where $p_i(t) \in O$ is the outcome proposed by agent $i \in G$ in round $t$. In every negotiation round $t$ all agents make a proposal depending on their time-dependent internal states (given as a vector $w_i(t) \in \mathbb{R}^m$, see definition 2.4) and the proposal vector $p(t)$, i.e. $p_i(t + 1) = \text{proposal}(p(t), w_i(t))$. A negotiation group agrees on outcome $o \in O$ in round $t$ if $\forall i \in G : p_i(t) = o$.

As mentioned above, concessions can either refer to behavioral changes or to mental adaptations. Here we follow the idea that concessions refer to mental adaptations. Therefore we introduce a mental
state of agents. Every agent assigns a weight itself and to all other partners. These weights represent how much this agent cares about itself and about the partners. This concept is already applied in the general context of cooperation and coordination mechanisms [1, 3, 11], but also in the specific domain of agent negotiations [2]. These weights are usually related to altruistic behavior, where altruism is defined as pursuing other agents’ goals – at least partially – instead of only pursuing own goals. We apply the same concept to model concessions as a stronger anticipation of partners’ goals, i.e. to become temporarily more altruistic. It might happen that an agent gets irrelevant, i.e. nobody including itself cares about the irrelevant agent.

Definition 2.4. (Weight vector)
Let $w_i : N \rightarrow R^m$ be a function for agent $i \in G$ that assigns each negotiation round $t \in N$ a weight vector. $w_i(t) = (w_{i,1}(t), w_{i,2}(t), ..., w_{i,m}(t))$ is the weight vector of agent $i$ at time $t$ with $w_{i,j}(t) \geq 0$ and $w_i(t) \cdot \vec{1} = 1$. Group member $i \in G$ is irrelevant at time $t$ if $\forall j \in G : w_{j,i}(t) = 0$.

The weight vector, which for convenience is normalized, is an agent’s mental attitude to consider particular agents’ individual utilities. If an agent assigns to one partner a bigger weight than to another partner then this means that the utility of the first partner has more impact on this agent’s decision than the second has. A proposal is a possible outcome that is best according to a balance between the own and all partners’ utilities. This balancing is implemented by an interactive utility. For the interactive utility function we assume that with an increase in the utility of a negotiation partner the interactive utility does not decrease. An agent should not benefit only from a decrease in somebody’s utility without an increase elsewhere.

Definition 2.5. (Interactive utility)
Based on the weight vector $w(t)$, the interactive utility $I$ balances group members’ utilities:
$I : O \times R^m \rightarrow R$ with $\forall i : \frac{\partial (o \cdot w_i)}{\partial o_j} \geq 0$.

Proposing a possible outcome To choose a proposal $p_i$, the agent $i$ selects a possible outcome that maximizes the interactive utility. Agents do not have to know all members of $O$, as long as they can determine the element with maximal interactive utility given a specific weight vector. This may get important if the space of possible outcomes is structured and an agent has only explored a specific part of it. If there are more maximal outcomes, then the agent designer can choose an appropriate selection method. For our simulations we will specify this below.

Definition 2.6. (Proposals maximizing interactive utility)
Let $M : \wp(O \times R^m) \rightarrow \wp(O)$, with $M(O, w_i(t)) \subseteq O$ be the set of possible outcomes that maximize the interactive utility of agent $i$ for a given weight vector $w_i(t)$ in a negotiation group of $m$ agents. For every proposed outcome $p_i(t) \in O$ we require that it has a maximal interactive utility, i.e. $p_i(t) \in M(O, w_i(t))$ with $M(O, w_i(t)) = \arg\max_{o \in O} I(o, w_i(t))$.

The idea to put altruism in an interactive utility is taken from Chapelle et al., where it is called interactive satisfaction [3]. Brainov calls it social preference [2].

If it is obvious or irrelevant which agent a weight vector $w_{i,j}(t)$ belongs to and it is clear whether we refer to a vector or an element of it, then we omit the first index and write $w_j(t)$. If it is also clear what time we talk about, we further simplify and write $w_j$. 
In negotiations, every participating agent that is designed to follow a weight-based negotiation strategy has its own weight vector. To reach an agreement all participating agents have to propose the same outcome. For simple interactive utility functions, e.g. weighted sum or products, an agreement requires that the weight vectors of negotiation partners are sufficiently close. Thereby "sufficiently" depends on the structure of negotiation opportunity $O$ and it depends on the interactive utility function. A concession should make an agreement more possible. Favoring particular agents in a negotiation may imply that concessions make agreements with these agents more likely. Hence, in such cases we can implement a directed concession as a "move" in the space of feasible weight vectors. The move should decrease the distance between the own weight vector and the weight vector of the partner the concession is directed to. Basically, we translate the negotiation from a search in a (possibly discrete) negotiation space into a search in the (continuous) weight vector space. Prerequisite for such moves is a sufficiently good estimation of the other agent’s weight vector, i.e. its mental state. In the next section we show for the weighted sum approach how agents can estimate other agents’ weight vectors.

3. An example: The weighted sum approach

In the last section we skipped the precise specification of the interactive utility function and the way to estimate another agent’s weight vector, which both are required to move toward other agents’ weight vectors. In this section we present and simulate the weighted sum approach as an example and show how directed concessions can be implemented.

Definition 3.1. (Weighted sum approach)
The weighted sum approach assumes the interactive utility function to be the scalar product of an agent’s weight vector $w(t)$ and a possible outcome $o \in O$. It implements Definition 2.5:

$$I_{w\text{Sum}}(o, w(t)) = w(t) \cdot o$$

We assume that every agent correctly believes that the other agents also follow a weighted sum approach. This implies that we can map the question of getting closer to somebody’s proposal to getting closer to the other one’s weight vector. In future work we will explore whether these assumptions about other agents can be relaxed.

3.1. Estimating other agents’ weight vectors

An agent making a directed concession towards a negotiation partner has to identify the partner’s position. In our approach it has to estimate the partner’s weight vector based on the other agent’s proposal. For a precise calculation of the space where the other agent’s weight vector can be, we might consider all possible outcomes and calculate which weight vectors give the proposal a maximal interactive utility. The weight vector is in the subspace where the proposal has a maximal interactive utility. In case that many possible outcomes are considered this can require a high computational effort. Therefore, we apply a simpler but less precise algorithm (see Figure 1 for an illustration): The agent only considers the own and the selected partner’s proposals. For both proposals $p_i$ and $p_j$ of agents $i$ and $j$ it identifies the area in the space of weights that represents all weight vectors $w$ such that both proposals are equally valued according to the interactive utility. To move closer to the other agent’s weight vector an agent has to cross this area, which we refer to as the border. More precisely, if we move orthogonal towards this border,
we get closer to all weight vectors that the partner can have. For the weighted sum approach, the border is given by (1).

\[ w \cdot p_i = w \cdot p_j \quad (1) \]

To deal with the condition that the weight vector is normalized, i.e. \( \vec{1} \cdot w = 1 \), we transform (1) such that we can arbitrarily choose one element of \( w \), e.g. the \( k \)-th one. Later we can choose this element such that condition \( \vec{1} \cdot w = 1 \) holds. We expand (1) to \((p_i - p_j) \cdot w - (p_{i,k} - p_{j,k}) = -(p_{i,k} - p_{j,k})\). Since \( \vec{1} \cdot w = 1 \) we can write this as given in (2).

\[
(p_i - p_j) - \vec{1} \cdot (p_{i,k} - p_{j,k}) \cdot w = -(p_{i,k} - p_{j,k})
\]

With normal vector \( n = p_i - p_j - \vec{1} \cdot (p_{i,k} - p_{j,k}) \) and \( d = -(p_{i,k} - p_{j,k}) \) equation (2) describes a hyper plane in the weight vector space in Hesse form: \( n \cdot w = d \). Dimension \( k \) can be chosen arbitrarily because for every \( k \) the normal vector has the element \( n_k = 0 \). We can easily calculate the foot \( f \) from the current weight vector \( w(t) \) to the hyper plane as given in (3). Due to our simplification we can chose \( f_k \) such that condition \( \vec{1} \cdot f = 1 \) holds, i.e. \( f \) is in the border.

\[
f = w(t) - \frac{w(t) \cdot n - d}{n \cdot n} \cdot n \quad \text{with} \quad f_i = 1 - \sum_{i \in G/n \neq l} f_i
\]

The foot \( f \) is the point that gives the direction for adapting the own weight vector to get closer to the partner’s proposal. But, currently we have only ensured that the sum of all weights is 1, but we do not enforce that every weight is non-negative. If for the foot this condition does not hold, then we move the foot such that it is in the feasible area. Of course, the new point, which we refer to as the matching point \( m \), is not necessarily a foot of a perpendicular from \( w(t) \) anymore, but we aim at minimizing the distance between the matching point \( m \) and the foot \( f \). Our approach is to set the dimensions that are negative to zero. To remain within the border we have to adjust the positive elements appropriately.

Let \( f^- \) contain all elements of the foot \( f \) that are negative and let \( f^c \) be the corrections of the foot’s non-negative elements. The matching point \( m \) is then given by (4). Now we have to specify \( f^c \) such that the length of vector \( f^c \) is as short as possible.

\[
m = f - f^- + f^c \quad \text{with} \quad \vec{1} \cdot f^c = \vec{1} \cdot f^- \quad \text{and} \quad f_i^c = \begin{cases} f_i, & \text{if} \quad f_i \leq 0 \\ 0, & \text{if} \quad f_i > 0 \end{cases}
\]

The matching point \( m \) has to be in the border; hence, \( n \cdot m = d \) holds. Together with (4) we get \( n \cdot (f - f^- + f^c) = d \). Because also the foot is in the border, i.e. \( n \cdot f = d \), we get \( n \cdot f^- = n \cdot f^c \). Let \( n^+ \) be the normal vector with zero for all elements that are negative within the foot \( f \). Because these dimensions are also zero in \( f^c \), we have \( n \cdot f^c = n^+ \cdot f^c \). This gives us all possible corrections \( f^c \) as given by (5).

\[
n^+ \cdot f^c = n \cdot f^- \quad \text{with} \quad \vec{1} \cdot f^c = \vec{1} \cdot f^-
\]

We want to choose the matching point that is as close as possible to the original foot. Therefore we have to minimize the distance between the foot \( f \) and the matching point \( m \), which implies that we want to minimize \( | f^c | \). We can do this by drawing a perpendicular from the origin \( \vec{0} \) to the space of possible corrections \( f^c \). Doing this we have to make sure that the sum of vector \( f^c \) equals the sum of vector \( f^- \).
Therefore, we multiply both sides of (5) with $\frac{1}{f^-}$. Now we can apply the same way of calculation as described above to $n^+ \cdot f^c = \frac{n \cdot f^-}{\|f^-\|} \cdot f^c - n_l$ with $n_l > 0$ can be chosen arbitrarily.

$$(n^+ - \bar{1} \cdot n_l) \cdot f^c = \frac{n \cdot f^-}{\|f^-\|} - n_l \text{ with } \bar{1} \cdot f^c = 1 \quad (6)$$

After calculating the foot from $\tilde{0}$ to the hyper plane given by (6) we get $f^c$ as given in (7). Replacing $f^c$ in (4) we can now easily derive the matching point $m$.

$$f^c = \frac{n_f \cdot f^- \cdot f_c}{(n^+ - \bar{1} \cdot n_l)(n^+ - \bar{1} \cdot n_l)} (n^+ - \bar{1} \cdot n_l) \text{ with } f_c^l = \bar{1} \cdot f^- - \sum_{i \in G \cap i \neq l} f^c_i \quad (7)$$

**Iterated weight vector estimation** For estimating another agent’s weight vector it can happen that for the identified matching point neither the own nor the other agent’s proposal has a maximal interactive utility, but instead the outcome $o$ has it. A first idea to utilize this for making the weight estimation better is to take $o$, to identify the border between $o$ and the partner’s proposal $p_j$, and finally to identify a new refined matching point $m'$. By simulations we will check whether this approach can shorten negotiations by iteratively estimating the partner’s weight vector.

### 3.2. Making concessions

The matching point gives an agent the direction to go for to get closer to the partner’s weight vector. An agent making concessions adapts its weight vector toward this matching point $m$. The adaptation can be big or only small; we refer to this as the *step length*. If we assume a step length that depends proportionally on the distance between $m$ and the agent’s current weight vector $w(t)$, e.g. 30% of this distance, then we might end up with an infinite process never reaching an agreement. To avoid this we implement a fixed step length $s$ with one exception: If the distance between the old weight vector and the matching point $m$ is smaller than the fixed step length, then the agent adopts $m$.

The proposal selection that we will introduce in the next subsection applies a majority rule for solving situations where more than one possible outcome has a maximal interactive utility. This selection method can cause cyclic proposals. That happens if agents move their weight vectors from one location to another and back. To avoid this we keep track of the movement of the weight vector. If such a cycle is detected then the step length is temporarily divided by 2 which lets the movement stop in between.

$$w(t + 1) = \begin{cases} m & \text{if } w(t - 1) \neq m \land |w(t) - m| \leq s \\ s \cdot m + (1 - s) \cdot w(t) & \text{if } w(t - 1) \neq m \land |w(t) - m| > s \\ 0.5 \cdot m + 0.5 \cdot w(t) & \text{if } w(t - 1) = m \land |w(t) - m| \leq s \\ 0.5 \cdot s \cdot m + (1 - 0.5 \cdot s) \cdot w(t) & \text{if } w(t - 1) = m \land |w(t) - m| > s \end{cases} \quad (8)$$
Figure 1. Estimating other agent’s weight vector in a negotiation with three agents: (a) the space of possible weight vectors, (b) agent 3 disagrees with 1 (reduced to 2 dimensions), and (c) calculating the matching point with one iteration.

3.3. Making proposals

For weight-based negotiation mechanisms we defined that an agent’s proposal has to have a maximal interactive utility, i.e. \( p_i \in M_i(O, w_i(t)) \). We did not specify how we choose the proposal if set \( M_i(O, w_i(t)) \) contains more than one element. If we simply choose randomly from this set, then the negotiations become very inefficient. The more agents participate and the more outcomes are best, the longer they need to accidentally agree. But if we follow another strategy, we have to take care that we do not get into deadlocks, e.g. situations where agents alternate between two or more different proposals but do not reach an agreement.

If the set \( M_i(O, w_i(t)) \) contains only one element then this is proposed. We now only deal with the case that more than one outcome maximizes the interactive utility. If the set contains outcomes that were proposed in the last negotiation round by any agent then only these outcomes are considered. The agent chooses randomly from those outcomes that were proposed by most agents. This proposal selection strategy ensures that agents being indifferent between several outcomes join the majority. This optimization has a drawback; in fact it can lead to a situation where agents end up in cyclic behavior, never reaching an agreement. The previously mentioned specific rule for weight vector adaptation solves most of such situations (see subsection 3.2).

3.4. Simulating the weighted sum approach

To explore some fundamental properties of the weighted sum approach we now simulate negotiating agents using the weighted sum approach. Four interesting parameters can be identified: number of possible outcomes, iterations of weight vector estimation, step length, and number of agents. Regarding the impact of these parameters on the length of negotiations we expect the following: First, we claim that the number of possible outcomes of a negotiation opportunity should not significantly affect the length of negotiations. The fact that we have mapped the search in a discrete negotiation space onto a search in continuous weight vector space implies that there should no impact of the number of possible outcomes on the length of negotiation. At most a small effect might be expected since the quality of estimations of a negotiation partner’s weight vector depends on the number and structure of possible outcomes. Second,
Figure 2. Simulations for different parameter dependencies: (a) outcomes per negotiation opportunity, (b) agents per negotiation, (c) iterations for estimating partner’s weight vector, and (d) step length versus the average number of steps per negotiation. To determine how the weight vector estimation influences the other dependencies, we have plotted – where useful – three lines: solid for 4, dotted for 2, and dashed for 1 iteration(s).

we expect that an iterated estimation of the partner’s weight vector improves the estimation, thus shortens the negotiation process. Third, we think that bigger steps decrease the length of negotiations. Finally, we expect that with an increase of the number of agents also the average length of a negotiation gets bigger.

For the simulations we assume – if nothing else is given – that the number of possible outcomes is 20, the number of agents $m$ is 4, three iterations are applied to weight vector estimation, and the possible outcomes are randomly drawn but are restricted by: $\forall o \in O : \forall i \in [1, m] : o_i \in \mathbb{R}$ and $o_i \in [0; 5]$. To explore the impact of the four parameters we set up different simulations; for each we did 1,500 simulation runs. We simulated with 2, 3, 4, 5, 6, 8, 10, 12, 14, and 16 agents. The number of outcomes varied among 5, 15, 20, and so forth until 50. The iterations were taken from 1, 2, ... , 10. Finally, we simulated for step lengths of 0.02, 0.03, 0.05, 0.08, 0.12, 0.17, 0.23, 0.3, 0.38, and 0.47. Figure 2 shows our simulation results.\(^4\)

In Figure 2 we can see that the impact of the number of possible outcomes is small. The fact that we have mapped the search in a discrete negotiation space onto a search in continuous weight vector space implies that there is no impact of the number of possible outcomes on the length of negotiation. But,\(^4\)

\(^4\)Since the calculations are based on tests of equality between real numbers, we had to do much work to make the algorithms robust against the restricted precision of the simulation platform. The simulation is only an illustration; therefore we do not give all the details regarding the robustness here. Still, these restrictions of precision cause some biases, but they are not significant.
because the quality of the weight vector estimation depends on the structure and number of outcomes there is a small influence. The number of agents seems to have a significant effect on the negotiation length. But, the more precise the partner’s weight vector is estimated the less is the impact of the number of agents on the negotiation length. If calculation time is very important, then two iterations should be a good choice. The simulations varying only the number of iterations among a bigger range also confirm this. The number of iterations for the weight vector estimation has a bigger effect the more agents negotiate and the bigger the step length. In the fourth diagram we can observe that the step length has a big influence when it is small. But the bigger the step length is the less is its impact.

4. Properties of the negotiation mechanisms

The previous sections have introduced negotiation mechanisms utilizing the concept of maximizing a weight-based interactive utility. For the weighted sum approach, which is a specific weight-based mechanism, we also addressed the problem of estimating a partner’s weight vector. We now want to characterize the possible agreements that can result from these negotiation mechanisms. An important concept for characterizing negotiation results is Pareto-optimality. We first show that the weight-based mechanisms with interactive utility functions that are not decreasing in the agents’ individual utilities result only in Pareto-optimal agreements. We additionally show that the weighted sum approach cannot reach all Pareto-optimal agreements, but there is a value for the parameter of a $\rho$-norm-like interactive utility function such that all Pareto-optimal possible outcomes can be reached.

4.1. Pareto-optimality of the weight-based negotiation mechanisms

Pareto-optimality is an often-used concept to characterize results of negotiation processes. An agreement is Pareto-optimal if there is no other possible outcome such that at least one participant gets more utility while keeping all other’s utilities equal.

**Definition 4.1. (Pareto-optimal outcomes)**

We define $PO \subseteq O$ as the subset of an opportunity $O$ that contains all outcomes that are Pareto-optimal. An outcome $o \in O$ is Pareto-optimal in $O$ if $\forall o' \in O : \forall i : o_i \geq o'_i$.

**Theorem 4.1.** Assume a weight-based negotiation mechanism as defined above. If no agent is irrelevant and they only propose outcomes such that there is a weight vector that makes the interactive utility of the proposed outcome bigger than or equal to all other outcomes, then the agreement is Pareto-optimal.

**Proof:**

Let there be an agreement with all agents proposing $o \in O$. This implies that $o \in M_i(O, w)$ for all agents $i \in G$. Now assume that there is another outcome $o' \in O$ such that all agents receive the same utility but one agent, e.g. agent $i$, gets more. Because the interactive utility function is not decreasing in the individual utilities and no agent is irrelevant (which implies that for every agent $i$ there is at least one agent that has a positive weight $w_i > 0$), the interactive utility of at least one agent is not maximal for $o$. But this contradicts the assumption of $o$ being proposed by all agents, which requires that the interactive utility is maximized by $o$ for all agents. \(\square\)
4.2. A modified $\rho$-norm as interactive utility

In previous sections we have implemented the interactive utility as a weighted sum. We now assume the interactive utility to be a modified $\rho$-norm with parameter $\rho$ and two additional less important parameters $\alpha$ and $\beta$. We show that for the set of parameters we need, this function is increasing in the individual utilities; hence, if it is applied to the mechanism introduced above, then only Pareto-optimal solutions are reached. We will see that the weighted sum approach cannot reach all Pareto-optimal solutions. But we can show that there is a $\rho$ for this interactive utility function such that all Pareto-optimal agreements are reachable. The $\rho$-norm as given in (9) does not allow for weighting the different dimensions of $p$. Therefore we consider (10) that is a modified $\rho$-norm such that weighted sum approach is a specific instance of it. Due to parameters $\alpha$ and $\beta$ also several other ways to implement weights are covered.

$$I_\rho(p) = \left(\sum_{i \in G} p_i^\rho\right)^{\frac{1}{\rho}}$$  \hspace{1cm} (9)

$$I_{m\rho}(\rho, w) = \left(\sum_{i \in G} \left(w_i^\alpha \cdot (w_i^\beta p_i)^\rho\right)\right)^{\frac{1}{\rho}}$$  \hspace{1cm} (10)

By setting parameters $\rho$, $\alpha$, and $\beta$ we can define different interesting interactive utility functions. For instance, if we choose $\rho = 1$, and $\alpha = 2$ and $\beta = -1$ the function corresponds to the weighted sum approach. To apply (10) as interactive utility function we have to take care that it is increasing the individual outcomes. It is easy to show that this holds for $\alpha + \beta \rho > 0$. Many interesting cases are captured by parameters as follows: $\rho \leq 1$, $\alpha \geq 2$, and $\beta \leq -1$. These restrictions, which guarantee the applicability of (10) as interactive utility function, we will consider in the rest of this article.

For the modified $\rho$-norm we claim that for every negotiation opportunity with a finite number of possible outcomes there is a $\rho$ bigger than negative infinity such that all Pareto-optimal outcomes can be reached, i.e. there is a weight vector for every Pareto-optimal outcome such that all agents propose this outcome.

**Theorem 4.2.** Let $I_{m\rho}(\rho, w) = \left(\sum_{i \in G} \left(w_i^\alpha \cdot (w_i^\beta p_i)^\rho\right)\right)^{\frac{1}{\rho}}$ be the parameterized interactive utility function for a weight-based negotiation mechanism with parameter $\rho$ and parameters $\alpha \geq 2$ and $\beta \leq -1$. If there is a finite set of outcomes, then for every Pareto-optimal outcome there is a value of $\rho$ and a weight vector $w$ such that this outcome is a proposal of all negotiation partners, i.e. an agreement.

**Proof:**

Fix an arbitrary Pareto-optimal outcome $q$. Theorem 4.2 holds, if for every outcome $s$ with $s \neq q$ the condition $\exists \rho \exists w : I_{m\rho}(q, w) \geq I_{m\rho}(s, w)$ holds. This immediately gives (11).

$$\forall s \neq q \exists w \exists \rho : \left(\sum_{i \in G} \left(w_i^\alpha \cdot (w_i^\beta q_i)^\rho\right)\right)^{\frac{1}{\rho}} \geq \left(\sum_{i \in G} \left(w_i^\alpha \cdot (w_i^\beta s_i)^\rho\right)\right)^{\frac{1}{\rho}}$$  \hspace{1cm} (11)

We now assume $\rho < 0$, which allows us to derive (12).

$$\forall s \neq q \exists w \exists \rho : \sum_{i \in G} w_i^\rho \cdot (w_i^\beta q_i)^\rho \cdot \left(1 - \left(\frac{s_i}{q_i}\right)^\rho\right) \leq 0$$  \hspace{1cm} (12)
To simplify, let us fix the weight vector as \( w_i = q_i^{-\frac{1}{\beta}} \cdot 1/ \sum_{j \in G} q_j^{-\frac{1}{\beta}} \). The weight is normalized such that the sum equals one. If condition (12) holds for this specific case, then it also holds for the case without these restrictions. Applying the fixed weights and the restriction on \( \rho \) we can transform (12) into (13).

\[
\forall s \neq q \exists \rho : \sum_{i \in G} q_i^{-\frac{1}{\beta}} \left( 1 - \left( \frac{s_i}{q_i} \right)^\rho \right) \leq 0 \tag{13}
\]

Because \( q \) is Pareto-optimal we know that \( \exists i : q_i > s_i \). Therefore, there is a term \( \left( \frac{s_i}{q_i} \right)^\rho \) that is bigger than 1. With \( \rho < 0 \) we know that there is a term \( 1 - \left( \frac{s_i}{q_i} \right)^\rho \) that is negative and that goes to negative infinity the more \( \rho \) approaches negative infinity. For all \( i \) with \( q_i < s_i \) the term \( 1 - \left( \frac{s_i}{q_i} \right)^\rho \) is at most 1. Since \( q_i^{-\frac{1}{\beta}} \) is a constant positive factor, there is a \( \rho \) such that the positive summands are counterbalanced by the negative one. This implies that (13) holds and finally that the theorem holds.

\[\text{Estimating } \rho\]

Depending on the negotiation opportunity, i.e. the set of outcomes, we can estimate the value of \( \rho \), where all Pareto-optimal agreements can be reached. For doing this we apply (13) to all outcomes of an opportunity \( o \). Thereby we get (14).

\[
\forall q \in PO \forall s \neq q : \sum_{i \in G} q_i^{-\frac{1}{\beta}} \leq \sum_{i \in G} \left( \frac{s_i}{q_i} \right)^\rho \tag{14}
\]

Because all outcomes \( q \in PO \) are Pareto-optimal, we know that there is one \( j \) such that \( q_j > s_j \); and therefore we know that \( \frac{s_j}{q_j} < 0 \). If it is not the case then because of \( q = s \) the equality of (14) holds.

For all \( i \neq j \) we assume the worst case, i.e. element \( q_i^{-\frac{1}{\beta}} \left( \frac{s_i}{q_i} \right)^\rho \) does not add anything to the sum; we set these elements to zero. We can now simplify (14) to \( \forall q \in PO \forall s \neq q : \sum_{i \in G} q_i^{-\frac{1}{\beta}} < q_j^{-\frac{1}{\beta}} \left( \frac{s_j}{q_j} \right)^\rho \), which then can be resolved to \( \rho \). Equation (15) gives an upper bound for the value of \( \rho \) such that all Pareto-optimal outcomes can be an agreement.

\[
\forall q \in PO \forall s \neq q : \rho \leq \frac{\ln \left( \sum_{i \in G} \left( \frac{s_i}{q_i} \right)^{-\frac{1}{\beta}} \right)}{\ln \left( \frac{s_j}{q_j} \right)} \tag{15}
\]

\[\text{The weighted minimum approach}\]

We know that for a sufficiently small \( \rho \) every Pareto-optimal outcome can be reached. If we do not need a continuous interactive utility function we can apply the weighted minimum approach that results from the modified \( \rho \)-norm for a parameter \( \rho \) that approaches negative infinity: \( I_{wMin}(w, p) = \lim_{\rho \to -\infty} I_{m\rho}(w, q, \rho) = \min_i \left\{ w_i^\beta p_i \right\} \).

\[\text{Classification of negotiation settings}\]

Based on the parameter \( \rho \) one can develop a single scale measure to evaluate negotiation settings or proposals in a negotiation setting. In future work we will address questions like: Are there negotiation strategies that perform better for one \( \rho \) while performing badly for others? Or do particular negotiation strategies and protocols favor different Pareto-optimal outcomes for different instances of \( \rho \)?
Figure 3. The upper three diagrams plot the interactive utility for all possible weights (only weight $w_1$ is shown because $w_2 = 1 - w_1$) for three different instances of $\rho$ for all possible outcomes of opportunity $\{(4,2), (3.3, 2.2), (3.1, 3.1), (3, 2.5), (2, 4)\}$. Outcome $(3, 2.5)$ is not Pareto-optimal; hence, it does not have a maximum interactive utility for any weight in any case of $\rho$. If $\rho = 1$ then the interactive utility of $(3.3, 2.2)$ is below maximum for all weights $w_1$ ($w_2 = 1 - w_1$); hence, it is not a possible agreement for this case as well as in case $\rho = -4$. But, if $\rho = -\infty$ it is a possible agreement. In the second row of diagrams each point represents a possible outcome. The line marks all outcomes that are reachable for the given parameter $\rho$.

4.3. Example

To illustrate the concepts and equations we now consider a simple example. We set $\alpha = 2$ and $\beta = -1$. Now consider two agents 1 and 2 that negotiate opportunity $\{(4,2), (3.3, 2.2), (3.1, 3.1), (3, 2.5), (2, 4)\}$. The first row of Figure 3 plots the interactive utilities of all possible proposals for three different instances of $\rho$. For $\rho = 1$ it can be seen that if we can arbitrarily choose the weights, then only three different proposals can be made. The second row of Figure 3 makes this more visible. The line represents all outcomes that could be proposed. We can see that despite proposal $(3.3, 2.2)$ is Pareto-optimal, for a weighted sum approach it cannot be proposed. But we can see that the smaller $\rho$ is chosen, the more likely it is that all Pareto-optimal proposals can actually be proposed. Our estimation gives a maximal $\rho$ of $-24.978$. 
5. Conclusions

Intending to develop negotiating agents that are able to make directed concessions in negotiations, we presented weight-based negotiation mechanisms that transform the search through a negotiation space to a search through a continuous weight vector space. These mechanisms are characterized by the fact that an agent maintains a weight vector that assigns a weight to each negotiation partner. The agent only proposes outcomes that maximize an interactive utility that balances the own and the partners' estimated utilities based on the weight vector. We require that this interactive utility is increasing in the individual utilities. Based on the weight vectors we were able to implement the concept of directed concessions: The weight vector is moved toward the weight vector of the agent that should benefit from a concession.

We continued with an implementation and simulation of the weighted sum approach, i.e. a special case of weight-based negotiation mechanisms. For the weighted sum approach we have presented an algorithm to estimate the negotiation partner's weight vector. We also presented a way to adapt weight vectors that avoids some difficulties when more than one outcome is best according to several agents interactive utility. By simulations we documented the impact of step length, number of iterations for weight vector estimation, number of agents, and number of possible outcomes on the length of negotiations.

In the last part of this article we have shown that the weighted sum approach cannot reach all Pareto-optimal solutions. But we showed that they could be reached by a continuous interactive utility function based on a modified $\rho$-norm. This is also guaranteed by the weighted minimum approach, which is not continuous. Since weight-based approaches are not only used in negotiation setting but also for other coordination mechanisms these results are not limited to negotiation but are relevant for the broader context of coordination mechanism.

Acknowledgment This article is a result of the effort done within the INKA project. The INKA project has been granted by the German Research Foundation (DFG) priority program ”Socionics” under 788/4-2 and 788/4-3. Many thanks to Hans-Dieter Burkhard, Kay Schröter and Dagmar Monett Díaz, who were partners in many discussions related to the issues of this article. I also thank Daniel Johannson for fruitful discussion of specific aspects of this article. A preliminary version of this article was discussed at and printed in the proceedings of the Workshop on Concurrency, Specification, and Programming (CS&P’2004) in Caputh, Germany [12].

References


