Eliminating issue dependencies in complex negotiation domains

Article *in* Multiagent and Grid Systems · December 2010

DOI: 10.3233/MGS-2010-0162 · Source: DBLP

3 authors:

Koen V. Hindriks
Vrije Universiteit Amsterdam

Catholijn M. Jonker
Delft University of Technology

Dmytro Tykhonov

Multiagent and Grid Systems – An International Journal 6 (2010) 477–501
DOI 10.3233/MGS-2010-0162
IOS Press

Eliminating issue dependencies in complex negotiation domains

Koen Hindriks, Catholijn Jonker and Dmytro Tykhonov*
Man-Machine Interaction, Faculty of EEMCS, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands

Abstract. In multi-issue negotiations, issues may be negotiated independently or not. In the latter case, the utility associated with one issue depends on the value of another. Searching for good bids in a utility space based on multiple, dependent issues is in general intractable. Furthermore, existing negotiation strategies have proven to be efficient for negotiation in domains with independent issues cannot be used in case of dependencies between issues. To deal with this increased complexity, one can introduce a mediator in the negotiation setting, increase the power of computers exponentially, or approximate the utility space. Given the number of high quality and tractable algorithms that exist for independent issue sets, in this paper an approach that approximates the complex utility space by eliminating the dependencies is proposed. The approximated spaces can be used by existing negotiation algorithms that proved their worth in domains without issues dependencies. The approach exploits knowledge of experienced negotiators that determines for a domain and a set of negotiating parties the expected outcome range. The more specialised the knowledge of the negotiator, the more narrow the expected outcome range, and the more precise our approximation. Using the approximated space instead of the original without any further safeguards would be risky; what seems a good deal in the approximated space might be a bad deal in the original space. To mitigate this risk we introduce a safety procedures that interfere both in the approximation phase and in the bidding phase. The first safety procedure tunes the parameters of the approximation procedure to control the outcome deviation. We show how these parameters can be used to balance computational cost and accuracy of negotiation outcome. Based on experimental results specific values for the parameters are determined that, in general, provide a good balance between computational costs and accuracy. The second safeguard consists of a bid search wrapper algorithm that ensures that during the bidding no mistakes are made that are related to the use of the approximated space. Our approach is based on the assumption that the typical structure of general negotiation spaces is "near linear", making it worth to do the approximation and checking its accuracy.

Keywords: Multi-issue negotiation, issues dependencies, non-linear utility spaces, utility spaces approximation

1. Introduction

Negotiation is a process by which a joint decision is made by two or more parties [17]. The parties first express contradictory demands and then move towards agreement by a process of concession making. Negotiation is an important method for agents to achieve their own goals and to form cooperation agreements, see e.g. [1,2,21,22]. Raiffa [19] explains how to set up a preference profile for each negotiator that can be used during negotiation to determine the utility of exchanged bids. For more information on utility and other game theoretic notions the reader is referred to e.g. [16]. Representing agent's preferences in terms of mathematical formulae expressing relationships between values of issues and the utility of bids allows the development of software support for negotiations. The complexity of these relationships determines the computational costs of the negotiation process. One way to avoid such

*Corresponding author. Tel.: +31 15 278 3737; Fax: +31 15 2787141; E-mail: dmytro@tykhonov.com.

computational costs is, as proposed in e.g. [10,11], to build up profiles as combinations of independent and simple evaluation functions per issue. This approach corresponds to the way the average human tackles negotiation. Humans tend to simplify the structure of their preferences [24] and prefer to negotiate one issue at a time, which means that issues influence the utility of a bid independently from each other. Absence of issue dependencies allows for the use of efficient negotiation strategies.

A number of efficient negotiation strategies exist for negotiation domains with independent issues, e.g. see [4,5,8,10]. The strategies try to find offers close to Pareto efficient frontier. The Pareto efficient frontier consists of those offers b, such that there is no offer b' that is better than b for one of the negotiating parties without being worse for the other parties. These strategies all rely on the assumption that an efficient search algorithm is available to them. Such search algorithms exist for the domains where issues are independent of each other.

In some domains, however, issue dependencies influence the overall utility of a bid. For example, the structures of preference profiles of the AMPO vs City negotiation case study presented in [18] are mostly linear additive, but there are some weak dependencies between two issue. For instance, the City would gain some additional "bonus" utility if both the officers with less than 5 years of service as well those with more than 5 years of service would get an increase in vacation. The utility of the "bonus" cannot be modelled using a linear additive function because utility of the two options together is more than sum of the uitlities of the individual options.

Non-linear utility functions and preference elicitation methods are studied in the multiple criteria decision making and multiattribute utility theory [3,15,25]. Bar-Yam [1] shows that in multi-issue negotiations with issue dependencies utility can only be described by non-linear functions of multiple issue variables. In such cases it is no longer possible to negotiate one issue at a time. Furthermore, Klein and co-authors argue in [12] that there is no efficient method an agent can use to negotiate multiple issues, even if the agent tries to guess the opponent's profile. The authors propose to use a mediator who uses a computationally expensive evolutionary algorithm that can solve non-linear optimization tasks of high dimensionality.

This paper introduces the WAID (Weighted Approximation for Issue Dependencies) approach to tackle the complexity problem of a utility space with issue dependencies. The reasons to try an approximation approach for negotiation spaces even though it is well known that for general non-linear spaces any approximation that reduces the complexity of the space to linearity will have serious flaws are twofold. The original spaces can be simply said "too wild" for an approximation approach to work. The first reason is that not all bids are equally important for negotiation: there are some bids that are not acceptable for the agent or are too optimistic to be an outcome of the negotiation. We pose that in effect, it is possible to indicate an expected region of utility of the outcome. The second reason is that in real life cases the profiles to be modeled by utility functions are far from "wild"; their structure is "near linear". However, understanding that a thorough analysis of the structure of real life negotiation spaces has never been conducted, we insisted on determining the necessary safeguards under which an approximation will lead us to efficient negotiation outcomes, where the efficiency of the outcome is, of course, determined with respect to the original space.

The first safeguard put into place functions during the approximation phase. It consists of an comparative analysis of the original versus the approximated space. The analysis results in an assessment of the error with respect to the range of utility values that correspond to the expected outcome range. As that range in the negotiation space will be used most during the bidding, the approximation should be be good (enough) in that range, whereas for other ranges of utility values the approximation can be less precise.

The second of those safeguards introduced in this paper is a bid search wrapping algorithm. The wrapper ensures that the error that is made during the negotiation that can be attributed to the use of the approximated space is small enough (a factor δ). That is, no bid is offered to the other party of which the real utility is not tested in the original space. If the difference in utility is acceptable, then the bid may be used in the bidding, otherwise another bid must be found by the search algorithm. In that way, the bid search wrapper provides a way to check the adequacy of the approximation by a measure of the introduced error. The wrapper can be used on any search algorithm that searches for bids with a particular utility and can easily be adapted for other types of bid search algorithms.

The third and last safeguard put into place is an experimental analysis of our whole proposal from the start of the approximation to the end of the bidding. The experimental analysis is based on the bid search wrapper and takes into account that the negotiation outcome does not only depend on the negotiation space induced by the preference profile. It also depends on the process of negotiation itself. It is to be expected that the risk of obtaining a significantly less optimal outcome due to the use of an approximation cannot be avoided completely even if the approximation is quite good and even though there are two runtime safeguards. By varying the factor δ from perfect (0 deviation) to arbitrary large deviations (factor of 1), the experimental set up tests the impact of the bid search wrapper safeguard.

The results of this experimental analysis shows that by allowing arbitrary large deviations in the bid search wrapper in some domains the risk of obtaining significantly worse negotiation deals may be unacceptable. In general, during the bidding process bids can be proposed of which the utility deviate δ from their utility in the original space. The experimental set up is required to investigate the accumulated effect of such errors over a bidding sequence.

Of course, the checking procedure that is part of the bid search wrapper introduces some additional computational costs. One of the main contributions of this paper is that it shows that a trade-off can be made between computational efficiency and approximation accuracy, which is directly related to the negotiation outcome. The parameters of the checking procedure allow the tuning of the negotiation algorithm to increase either the computational efficiency or decrease the risk of erroneous bids. Derived from experimental results, we propose specific values for these parameters that ensure a reasonable balance between computational costs and outcome deviation (in terms of utility) in many domains. Finally, the experimental results show that adding a checking procedure in the bid search wrapper is scalable and allows an agent to negotiate about high-dimensional utility spaces.

As an illustrative example of dependent issues, in this paper, we consider the negotiation of an employment contract where two important issues are at stake: the number of days that have to be worked and the number of days that childcare will be provided by an employer. In the example, the candidate employee additionally has to take into account a dependency between these two issues: working time (issue variable x_1) needs to be balanced with the time s/he needs to spend with his/her child (issue variable x_2). Assuming that the partner of the candidate is working too and can take responsibility for only part of the childcare, the candidate has promised that s/he will take care of the child for at least 2 days, either by taking care in person, or by finding professional childcare. Thus the child care issue is really important and in case the employer proposes a contract for 5 days our candidate will try to negotiate a result which includes at least 2 days of childcare. In terms of utility, bids with 5 working days and less than 2 days of childcare have a low utility (e.g. $u(5, 0) \approx 0.1$, $u(5, 1) \approx 0.5$). In case the employer proposes a contract for only 4 days, the candidate will need to negotiate a result including only one day of childcare and a bid of 4 working days and one day of childcare has an acceptable utility value associated with it (e.g. $u(4, 0) \approx 0.25$, $u(4, 1) \approx 0.55$) though the candidate would prefer to work more. With respect to bids of the employer that require the candidate to work 3 days or less, there is no problem regarding the caretaking of the child. In that case, the childcare issue has much less influence on the value of the bid (e.g. $u(3, 0) \approx 0.35$, $u(3, 1) \approx 0.55$. Even in this relatively simple example, the values associated with each of the issues cannot be modelled independently and overall utility cannot be calculated using a linear additive function. The contribution of the childcare issue to overall utility depends on the number of working days associated with the other issue and vice versa in a way that introduces non-linear dependencies between the issues.

The paper is organized as follows. The next section reports on the related work in the field. Section 3 provides a formal definition of utility spaces with dependencies between issues and gives a leading case study that is used throughout the paper to illustrate the method. Section 4 describes the approximation method for eliminating issue dependencies. The theme of Section 4.5 is the bid search wrapper algorithm based on the approximation method and shows that by varying the parameters of the method a trade-off can be made between outcome deviation and computational costs. Section 5 analyzes performance of the proposed bid search algorithm in an experimental setup. Finally, Section 6 concludes the paper.

2. Related work

Klein et al. [12] propose a negotiation protocol based on the mediated single text negotiation [18]. The protocol is specifically designed to handle complex utility spaces with dependencies between issues. In the protocol, a mediator proposes an offer that is initially generated randomly. Each agent then votes to accept or reject the offer. In case both agents vote to accept the offer, the mediator mutates the offer and the new offer is sent back to the agents. If at least one of the agents rejects the offer the mediator mutates the the most recent mutually acceptable contract and sends it to the agents. This procedure is repeated for a fixed number of times. This protocol can be scaled to negotiations with more than two agents.

Two types of agent strategies are used by Klein et al. [12]: “hill-climbers” and “annealers”. The hill-climbers accept an offer from the mediator if it’s utility is higher than that of the most recently mutually accepted offer. With a given probability the annealers can accept contracts worse than the one that is mutually accepted. The probability depends on the utility change between the contracts and the time to the negotiation deadline. The annealers are tuned in such a way that the probability of accepting worse contracts is higher in the beginning of a negotiation and decreases to zero when the negotiation time approaches the deadline.

In the experimental setup the annealers showed better negotiation results than the hill-climber due to the fact that the hill-climbers tend to get stuck in a local optimum while the annealers can explore utility space more extensively by accepting worse offers. However, in a negotiation setting with a mix of hill-climbers and annealers the hill-climbers drag the annealers into a possibly local optimum.

Klein et al. [13] extended the work of Klein et al. [12] by introducing annealing into the mediator. In this protocol, the mediator uses annealing to pursue not only the mutually accepted offers but also the ones that are rejected by the agents. To increase the mediator’s outcome efficiency the agents are required by the protocol to annotate their votes with a form of strength: weak/medium/strong accept (reject). The modified approach overcomes the problem of negotiation settings with a mix of hill-climbers and annealers. However, the protocol remains sensitive to the truthfullness of the agents in annotating the strength of their votes.

Ito et al. [9] extended the idea of using a mediator for negotiations with issue dependencies proposed be Klein et al. [12] by introducing a more advanced negotiation protocol. In the protocol, the agents randomly sample their own utility spaces to find areas of high utility and share them as well as the utilities of the areas with a mediator. The mediator finds overlaps between the reported areas and selects the one with the highest social welfare. The main disadvantage of the proposed protocol is in the necessity of a trusted third-party that would play the role of mediator. The mediator has to be trusted in keeping the shared information about the agent's utility spaces secret and being honest to all agents in maximization of the social welfare when searching for the final negotiation outcome. Furthermore, in the proposed protocol the agents have an incentive to lie about their utilities in order to get a better negotiation outcome for themselves.

The utility spaces that are used by Klein et al. [12,13] and Ito et al. [9] are defined by means of constraints in a multi-dimensional space. Every constraint is defined on a range of values in every dimension and has a single utility value. The utility space is defined as the sum of utilities of all constraints satisfied for a specific offer. The utility spaces defined in such a way are characterized by a "bumpiness" of the utility, i.e. sharp increases and decreases of the utility and a high number of local optimums. Such negotiation circumstances do not seem to be common in negotiation practice (see [11] for a number of examples of utility functions used in negotiation and decision making case studies) where preferences have some structure that is far from such "wild" behaviour. Because of the "wild" behaviour of the preferences according to [12] agents would not be able to reach an efficient agreement by means of a bilateral negotiation without a mediator. This limits applicability of the approach proposed by by Klein et al. [12,13] and Ito et al. [9] because of the need for a trusted third-party that will agree to play the role of a mediator. The approximation technique proposed in this paper enables negotiation between agents having complex preferences without a mediator.

The problem of negotiation with dependencies between issues is also studied in the context of bundling items under negotiation. The utility of a bundle can be non-linear with respect to the presence of certain items in the bundle. Thus the utility of two items in a bundle differs from the sum of utilities of the two items individually. Robu at al. [20] propose a graph-based technique to learn complex opponent's profiles. The authors propose an algorithm of exponential computational complexity for searching through a learned utility space of the opponent. The main goal of [20] is the scalability of a model for representing an opponent's profile, whereas the current paper aims to simplify an agent's profile. The technique of [20] can be only applied to negotiation domains with binary issues that represent the presence of a product in the bundle. The approximation technique proposed in this paper can handle issue with more than two values.

Somefun et al. [23] propose a negotiation algorithm for a shop negotiating with a customer about a bundle of products and a price. The strategy is able to handle non-linear utility functions of bundles. It tries to combine knowledge about preferences of potential customers aggregated from historical negotiations with a preference model of a customer that is learned during the online negotiation. The strategy has so far been tested only on relatively small domains with up to ten products in a bundle (that is $2^{10}-1 = 1023$ possible bundles), whereas the approximation technique proposed in this work was tested on domains with upto 25^{10} possible outcomes.

An interesting approach to approximating non-linear utility functions is proposed by Fatima at al. [6]. The authors split the non-linear utility function in several intervals and then approximate those intervals with linear functions. This can significantly improve the computational tractability of a bid search algorithm. The authors assume that the approximations as well as the corresponding approximation error are given and do not give any recommendations on how to obtain them. The approximation technique proposed in this work has a method to analyze approximation error as well a mechanism to control the error during bidding. Furthermore [6], does not consider the case of dependencies between the issues that is the main focus of our work.

3. Utility of interdependent issues

The overall utility of a set of independent issues can be computed as a weighted sum of the values associated with each of the separate issues. As is common (see e.g. [10,19]), an evaluation function is associated with each issue variable and the utility of a bid then is computed by the following weighted sum of the issue evaluation functions:

$$u(x_1, \ldots, x_n) = \sum_{i=1}^{n} w_i ev_i(x_i) \tag{1}$$

In Eq. (1), the (weighted) contribution of each issue to the overall utility only depends on the value associated with that issue and the contribution of a single issue can be modeled independently from any other issues. Evaluation functions for independent issues thus have exactly the same properties as the utility function associated with the bids that consist of multiple issues: it maps issue values on a closed interval $[0, 1]$. This setup can be used for issue values that are numeric (e.g., price, time) as well as for issue values taken from ordered, discrete sets (e.g., colors, brands).

Bid utility functions that are weighted sums of the contribution of single issue values to the overall utility cannot be used, however, for modeling dependencies between issues. The value of one issue may depend on that of another, thus influencing the utility of a bid that includes both issues. Dependencies between these issues give rise to a generalization of Eq. (1) to:

$$u(x_1, \ldots, x_n) = \sum_{i=1}^{n} w_i ev_i(x_1, \ldots, x_n) \tag{2}$$

The representation of a utility space with non-linear issue dependencies as in Eq. (2) is similar to the model proposed in [12]. The main difference is that instead of considering only binary issue values, we allow multi-valued, discrete, as well as continuous issue ranges.

The complexity of a utility function determines the computational complexity of the negotiation process. One of the main problems in dependent multi-issue negotiation is the computational complexity associated with searching for appropriate bids in the corresponding utility spaces. In case a utility function of multiple issues is non-linear in these issues, i.e. there are issue dependencies, finding a particular bid in the utility space is intractable. Computationally simple and efficient approaches covered in [14] mostly rely on the independence of issues to determine their next bid and are not applicable.[1]

For the non-linear dependencies of the employment contract, Eq. (1) does not suffice. The dependencies, however, can be modelled by Eq. (2). To make the example concrete, the candidate's preferences are modelled using the following evaluation functions:

$$ev_1(x_1, x_2) = 0.01x_1^2 + 0.03x_1x_2 + 0.028x_2^2 \tag{3}$$

$$ev_2(x_1, x_2) = -0.04x_1^2 + 0.13x_1x_2 - 0.11x_2^2 + 1 \tag{4}$$

Figure 1 shows the utility space of the candidate employee defined by the evaluation functions (3) and (4) and weights $w_1 = w_2 = 0.5$.

[1] As we discuss below, however, the approach can be adapted by using exhaustive search through the utility space, but becomes intractable and in practice works only for small utility spaces.

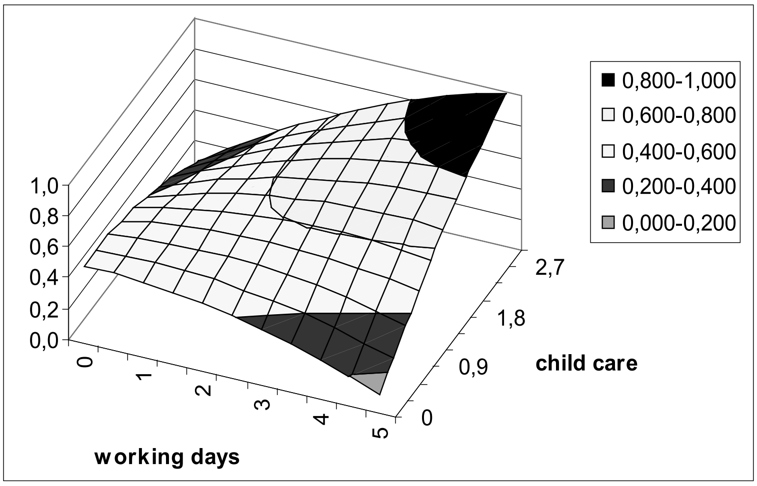

Fig. 1. Utility space of the candidate employee with issue dependencies.

4. Weighted approximation method

Due to the inherent computational complexity and the limited number of negotiation strategies that can be used to handle issue dependencies in negotiations, it would be beneficial to have methods that simplify the negotiation process of dependent issues without using a mediator. One particularly interesting option is to investigate the complexity of the utility space itself and try to eliminate the dependencies between issues. In case issue dependencies can be eliminated, various alternatives for efficient negotiation become available: Searching through the utility space of multi-issue bids becomes feasible and negotiation strategies for independent issues can be applied.

In this section, a weighted approximation method is proposed to eliminate issue dependencies (see Fig. 2). Weighted approximation is an averaging technique with some safety guards built in that are based on some general observations about negotiation (the m-point and the weighting function that will be explained later) and which can take available knowledge about a negotiation domain into account. In particular, knowledge about the relative importance of bids and about outcomes which reasonably can be expected are part of the weighted averaging method.

Although elimination of issue dependencies implies a loss of information and accuracy with regard to utility, this paper shows that if the influence of one issue on the associated value of another issue is “reasonable” (i.e., the utility space is not too wild) a good approximation of the complex utility space can be obtained.

The approximated utility can be used by a negotiation strategy to find an offer given some criteria, e.g. find an offer with a certain utility. The strategy can adopt a bid search wrapper algorithm as proposed in this paper. Due to the approximation error the found offer can have a significant deviation of utility in the original utility space. This deviation can be easily calculated due to the fact that calculating the utility of an offer even using a non-linear utility space is computationally cheap. If the deviation is unacceptable the search algorithm would propose another offer. The procedure is repeated until an offer with acceptable utility deviation is found.

The averaging technique proposed in this paper for eliminating dependencies is valid for utility spaces that have a certain “smooth” structure. The technique averages the values of bids close to each other. Therefore, utilities should not fluctuate too much from one bid to another within the proximity range set

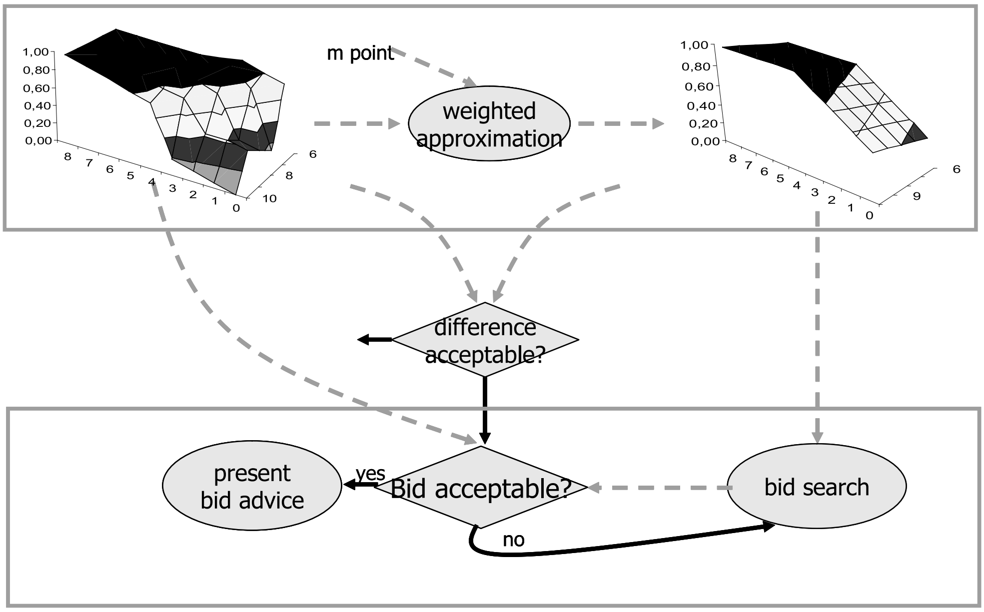

Fig. 2. The WAID method overview.

by the technique. In real life, common negotiations, this limitation on the applicability of the method is not seen as a problem considering that it is cognitively hard to make sense of wildly fluctuating utility spaces. As an indication, we think that the techniques are applicable to utility functions that can be modeled by polynomial functions of modest power. If the nature of the utility space is not clear, the applicability of the proposed techniques has to be tested for that case. A case study illustrates that the elimination of dependencies does not result in significant changes of the negotiation outcome. Additionally, a method for analyzing and assessing the difference between the original and approximated utility space is provided. This method to analyze and assess the results can always be applied to arbitrary utility spaces.

Our main objective thus is to find and present a method for transforming a utility space $u(x_1, \ldots, x_n)$ based on dependent issues that can be represented by Eq. (2) to a utility space $u'(x_1, \ldots, x_n)$ without such dependencies that can be represented by Eq. (1). There exist various techniques to transform complex (utility) spaces with non-linear functional dependencies between variables to spaces which are linear combinations of functions in a single variable [18]. For our purposes, we are particularly interested in the linear separability of non-linear evaluation functions of dependent issues.[2] The main idea is to transform a utility space $u(x_1, \ldots, x_n)$ into an approximation $u'(x_1, \ldots, x_n)$ of that space by approximating each of the evaluation functions $ev_i(x_1, \ldots, x_n)$ by a function $ev'_i(x_i)$ in which the influence of the values of other issues x_j, $j \neq i$, on the associated value $ev_i(x_1, \ldots, x_n)$ have been eliminated. Mathematically, the idea is to "average out" in a specific way the influence of other issues on a particular issue.

The WAID method takes as input a utility space based on non-linear issue dependencies (i.e. issues cannot be linearly separated and transforms it into a utility space that can be defined as a weighted sum

[2]In geometry, when two sets of points in a two-dimensional graph can be completely separated by a single line, they are said to be linearly separable. In general, two groups are linearly separable in n-dimensional space if they can be separated by an $n-1$ dimensional hyperplane.

of evaluation functions of single issues (i.e. issues are independent). The WAID method consists of the following steps:

1. As a first step, estimate the utility of an expected outcome that is reasonable (given available knowledge). This estimate is called the "m-point" and is used to define a region of utility space where the actual outcome is expected to be.
2. Select a type of weighting function. The selection of a weighting function is based on the amount of uncertainty about the estimated m-point (expected outcome) in the previous step.
3. Calculate an approximation of the original utility space based on non-linear issue dependencies using the m-point and the weighting function determined in the previous step. The result of this step is a utility space that can be defined as a weighted sum of evaluations of independent issues (a function of the form of Eq. (1)).[3]
4. Perform an analysis of the difference of the original and approximated utility space by means of a Δ-function to assess the range of the error for any given utility level. In this final step, based on the assessment, thresholds for breaking off the negotiation or accepting opponent's bids can be reconsidered.

Finally, the results of the WAID method can be used in combination with a particular negotiation strategy. In section 5.1, we study the results of using an approximated utility space for the child care example in a negotiation strategy and compare the results with an approach based on the original utility space. The sections below explain each of the steps in more detail and illustrate how these steps achieve the objective of eliminating issue dependencies.

4.1. Estimate an expected outcome

Any approach based on using uniform arithmetical averaging methods has the effect of discarding information uniformly. Such an approach does not take the final goal of negotiation into consideration: the negotiation outcome. A uniform averaging method is indifferent to the fact that even before negotiation starts it can be assumed that certain regions of the utility space are more relevant to the negotiation than others. Some general observations about the structure of utility spaces that can be associated with negotiations taken from actual practice provide additional insight that can be used to increase the effectiveness of an approximation technique.

Consider, to make clear what we mean, a worst case scenario in which two agents A and B associate completely opposite utilities with bids. In other words, what is valuable for agent A is of no value for agent B. Formally, we can express this opposition in terms of utility functions as follows:

$$u_A(x_1, \ldots, x_n) = 1 - u_B(x_1, \ldots, x_n) \tag{5}$$

Given these utility functions, it is easy to see that the Nash product is 0.25 with associated utility values $u_A(x_1, \ldots, x_n) = u_B(x_1, \ldots, x_n) = 0.5$ and the same point within the utility space is an efficient negotiation outcome when using Kalai-Smorodinsky criteria, that is, a Pareto-optimal outcome with equal utilities for both parties. Assuming such opposite interests, none of the agents would ever accept a bid which has a utility below 0.5.

[3] In the more general case of more than two issues, an evaluation function may depend on more than two issues and one of those issues has to be selected to be separated from the other issues.

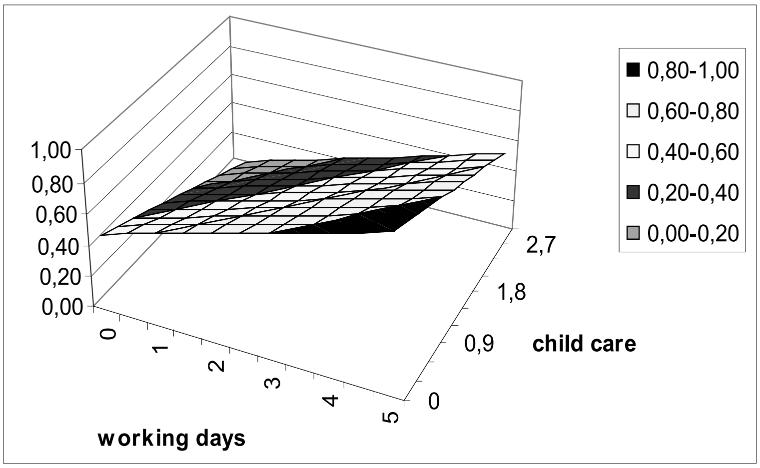

Fig. 3. Utility space of the candidate employee with issue dependencies.

Typically, however, negotiations do not fit such worst case scenarios and there is something to gain for both parties. Formally, this means that there exist acceptable negotiation outcomes, i.e. bids, with associated utilities that are higher than 0.5. In such cases, the utility spaces of the negotiating opponents are not completely opposite as expressed by (5). This line of reasoning makes clear that in general we may assume that the expected outcome of the negotiation is located somewhere in the open utility interval $(0.5; 1)$ and this region in the utility space is generally of more importance in a negotiation.

It follows from the previous considerations that some regions within the utility space are more important for obtaining a good negotiation outcome than others and in the WAID method proposed should be approximated as good as is possible. As a first step to identify these regions, an agent can estimate an expected outcome which would identify with some probability one of the more relevant points in the utility space. We call this point the "m-point".

An agent will be able to estimate an expected outcome with reasonable exactness only if it has some knowledge about the opponent's profile. In that case, as we illustrate below, the m-point can be computed in two steps. But even if an agent lacks any information whatsoever about its opponent an m-point can be based on considerations of the agent's own utility space. In the latter case, we propose that the m-point can be identified with the average of the break-off point (an agent breaks off a negotiation in case any utility with a lower utility is proposed) and the maximum utility in the utility space. In the childcare example, the break-off point equals 0.37, which is equal to the minimum utility that still satisfies the candidate employee's childcare constraint.

A second, more informed method to determine an expected outcome can be used when the agent does have some information, e.g. based on previous experience, concerning the opponent's profile. In the childcare example, assuming that the employer will take the child care request seriously into consideration, but will try to minimize his contribution in this regard, bids with 1–2 child care days are reasonable to expect. Additionally, it may be more or less certain that the employer prefers the employee to work as much as possible and that these issues are independent from the other. Then, as an estimated model of the opponent's profile, the following evaluation functions can be used, which, using equal weights of 0.5, result in the utility space depicted in Fig. 3.

$$ev_1(x_1) = x_1/5 \tag{6}$$

$$ev_2(x_2) = (3 - x_2)/3 \tag{7}$$

An estimate of the expected outcome can now be computed from the agent's own utility space and the educated guess of the opponent's utility space using Kalai-Smorodinsky criteria, which ensures that a Pareto-optimal outcome is selected and the expected outcome is not strongly biased in favour of either one of the parties (see Fig. 3). Calculating the utility in our example yields $m = 0.74$. This estimate may still be quite uncertain, but we will discuss this issue more extensively below. The estimated outcome only defines one parameter of the approach.

4.2. Select weighting function

As discussed above, not all points within the utility space are equally important for obtaining a good negotiation outcome. To take into account the relative importance of certain regions within the utility space, we introduce a weighting function associating a weight with each point (its "importance") in the utility space. In general, there are two useful considerations that can be made which provide clues for constructing an appropriate weighting function.

The first consideration is that a certain range of utility values are of particular interest in the negotiation. Also, certain bids may be more "appropriate" than others in a negotiation. As an example, bids with utility values below a break-off point are less significant than other bids and do not have to be approximated as well as others. In the childcare example, provided with the relevant domain knowledge, it is moreover unreasonable for our candidate employee to propose to do no work and at the same time to request 5 childcare days.

The first consideration concerning the approximation of the utility space can be given a formal interpretation by associating the highest weight with the expected outcome (the "m-point" identified above, located within the $(0.5; 1)$ interval).

The second consideration is the fact that an agent may be more or less uncertain about its estimate of the utility of the negotiation outcome. To take this into account, we propose to use two different functions depending on the level of uncertainty that the agent has about the estimate of the m-parameter. In case the agent does not have information about the opponent, nor any past experience with the particular negotiation domain and is quite uncertain about the most probable outcome, a relatively broad range of utility values around the expected outcome should be assigned a high weight. As a consequence, bids in a rather wide neighborhood of the m-point are equally important for the negotiation and only extreme points (with utilities close to one or zero) do not have to be approximated very accurately. Given a relatively large uncertainty, we propose to use a polynomial function of the second order, which is rather flat near the m-point and declines closer to the extreme utilities (see Fig. 4(a)). The corresponding weighting function ψ then can be computed as follows:

$$\psi(x_1, \ldots, x_n) = \frac{2}{m} u(x_1, \ldots, x_n) - \frac{1}{m^2} u^2(x_1, \ldots, x_n) \tag{8}$$

In the case the agent is reasonably certain about the estimate, for example, when the most probable region of the negotiation outcome is well defined on the basis of domain knowledge, knowledge about the opponent or experience gained in previous negotiations, a weighting function with a stronger differentiation of utilities values can be used. In that case, a Gaussian function that is defined in terms of a maximum point m and spread σ can be used that assigns high weights only to bids with a utility close to the expected outcome m (see Fig. 4(b)):

$$\psi(x_1, \ldots, x_n) = e^{-\frac{(u(x_1, \ldots, x_n) - m)^2}{\sigma^2}} \tag{9}$$

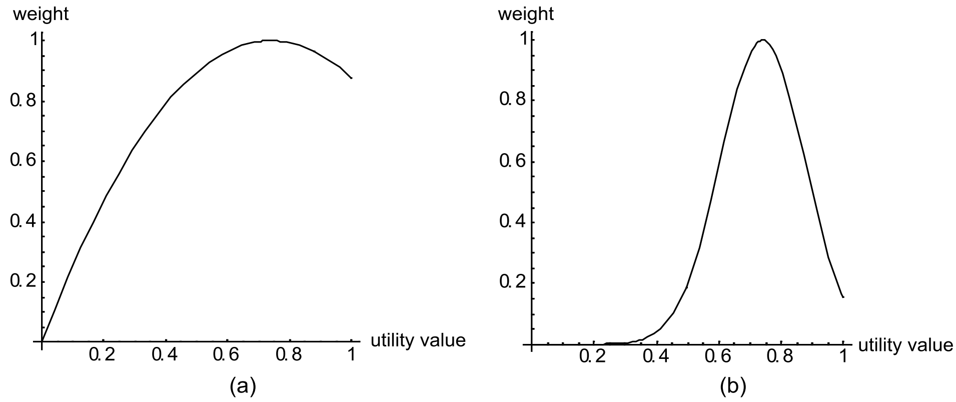

Fig. 4. Example of ψ function for $m = 0.74$.

The spread parameter σ provides an indication of the agent's certainty about expected outcome. In both cases, the m-parameter represents the expected outcome and is a point in the interval $(0.5; 1)$; ψ assigns the m-point the maximal weight of 1.0.

In our example, an educated guess of the opponent's profile could be made and therefore a Gaussian weighting function is selected and a value for the "spread" σ needs to be determined. To this end, we use the 3σ rule (or "Empirical rule"), which says that (most likely) 99.7% of all outcomes will be in the interval $(m - 3\sigma, m + 3\sigma)$, which gives us $\sigma = (0.37 + 0.74)/(2 * 3) = 0.19$.

4.3. *Compute approximation of utility space*

Using the weighting function ψ a weighted approximation technique can be defined. The weighted approximation technique proposed here first multiplies each evaluation value with its corresponding weight and then averages the resulting space by integration. In the equation below, a function ω is introduced instead of ψ since the weighting must be normalized over the interval of integration. The range of integration is identical to the range of the integrated issue.[4] In case a negotiation involves n issues with interdependencies between these issues, and evaluation functions $ev_i(x_1, x_2, \ldots, x_N)$ for the ith issue are given:

$$ev'_i(x_i) = \frac{\int_V \psi_i(x_1, \ldots, x_n) ev_i(x_1, \ldots, x_n) dV}{\int_V \psi_i(x_1, \ldots, x_n)} \tag{10}$$

Here V is a volume of $n - 1$ dimensionality build on the dimensions $x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$. Of course, not all issues have to depend on all others. The approximation technique can be applied sequentially for each evaluation function in the negotiation setup, which involves dependencies between issues.

As an illustration, we apply the weighted averaging technique to our employment contract negotiation. Figure 5 shows the ψ-functions for the original utility space using a polynomial function (8) for the left chart and a Gaussian function (9) for the right one. The flat section in the middle of the left chart represents a rather wide neighborhood of the m-point: this corresponds to the expected outcome and

[4]If the issue has discrete values, integration simply means summation over all these values.

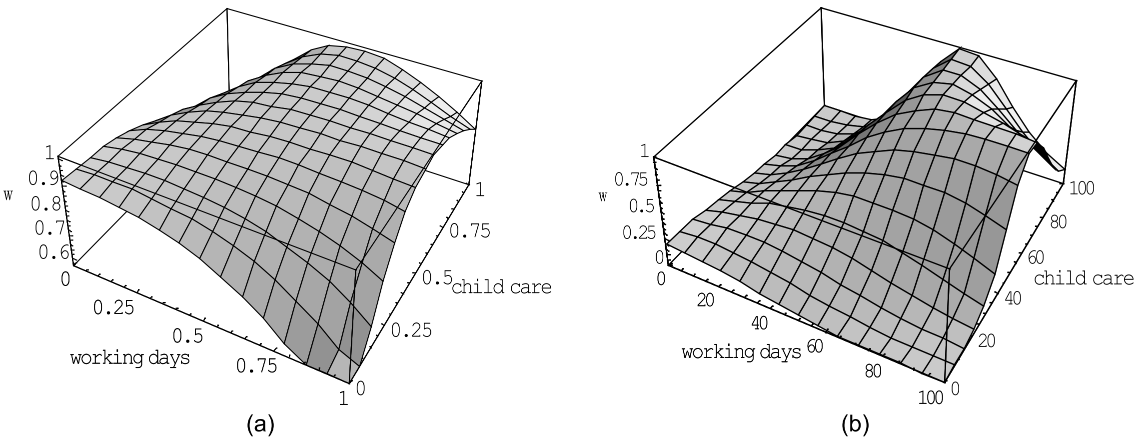

Fig. 5. Examples of ψ-functions for the employee's utility space: (a) polynomial function with $m = 0.74$; (b) Gaussian function with $m = 0.74$ and $\sigma = 0.19$.

weights in its neighborhood are high. Outside this region the weighting function slowly declines to zero. For the Gaussian function (right chart) we obtain a different picture: the function has high values (close to 1) for the small band of bids with utility values close to the m-point and declines rapidly for the remainder of the utility space.

We apply expression (10) to the evaluation functions of our employment contract negotiation example to derive an approximated utility space without interdependencies from the original utility space. Figure 6 shows the original (a) and approximated utility spaces obtained by approximation with a polynomial weighting function (b) and obtained by using a Gaussian weighting function (c).

The utility spaces obtained by approximation with the polynomial and Gaussian weighting functions have a similar structure. However, the Gaussian weighting function due to its stronger utility discrimination power makes it more precise in the vicinity of the m-point. This is explained in more detail in the next section.

4.4. Analyze difference Δ with original utility space

The technique presented approximates the original utility space and consequently, introduces an error in the utility associated with bids. To obtain a measure for the distance of the values of bids in the original utility space compared to the bids in the approximated utility space, a difference function Δ can be defined as follows:

$$\Delta(x_1, \ldots, x_i) = |u(x_1, \ldots, x_N) - u'(x_1, \ldots, x_N)| \tag{11}$$

As is to be expected, the Δ-values for the approximation using the Gaussian weighting function shift the utility considerably for some bids. For certain bids in the childcare example, the difference is almost 0.5. However, this only is the case for bids that are unreasonable and are not relevant for reaching a negotiation outcome. In particular, this shift in utility occurs for bids that involve more days of child care than working days. Approximations of the utility of bids that are close to the m-point are very good and close to zero.

To see the effect of the weighted averaging method near the m-point we take a section in the original utility space for the m-point ($m = 0.74$ for our negotiation example). By fixing the utility to 0.74, an expression can be obtained for the value of one of the issues as a function of another one:

$$u(x_1, x_2) = 0.74 \Rightarrow x_1 = f(x_2) \tag{12}$$

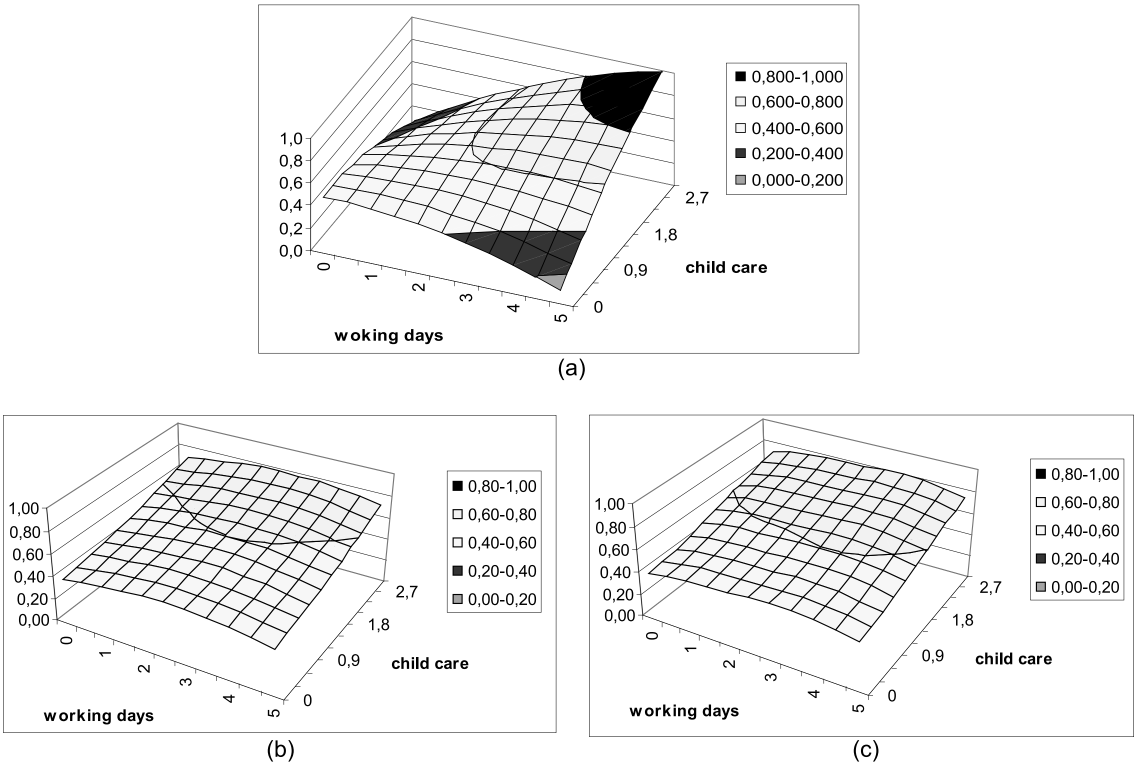

Fig. 6. The original utility space (a) and new utility spaces of the employee obtained by (b) the weighted averaging method using a polynomial weighting function with $m = 0.74$ and (c) the Gaussian weighting function with $m = 0.74$, $\sigma = 0.19$.

The function thus obtained can be substituted into the expression of the delta function (11). This provides us with the values of Δ for a fixed utility as a function of only one of the issues, and can be obtained for other utility values in a similar way.

The Δ-values obtained by weighted averaging with the polynomial weighting function and the Gaussian weighting function for utility equal to 0.74 are rather small for both (see Fig. 7(b)), but weighted averaging with a Gaussian function produces smaller approximation errors: it is almost twice as good. For bids with utilities of 0.9 the Δ-values (see Fig. 7(c)) rise in comparison with that of 0.7, however, the Gaussian weighting function still gives a better result. For bids with a utility of 0.5 (see Fig. 7(a)) the Δ-values are quite similar.

In Fig. 8, a worst case analysis is illustrated. It presents the utilities for extreme values of childcare (Fig. 8(a)) and for the number of working days (Fig. 8(b)) that run through the maximum Δ-value, corresponding to the bid with 0 working days and 3 days of childcare. It shows that the evaluation function associated with 0 days of child care (0 working days) is almost mirrored with respect to the evaluation function associated with 3 days of child care (5 working days). In effect, this shows that our child care example presents a serious test for our WAID method that somehow has to average these differences.

4.5. Bid search wrapper algorithm

The approximation of a preference profile by means of WAID allows an agent to more efficiently compute counter bids during negotiation. The WAID method ensures that a bid with a utility that is

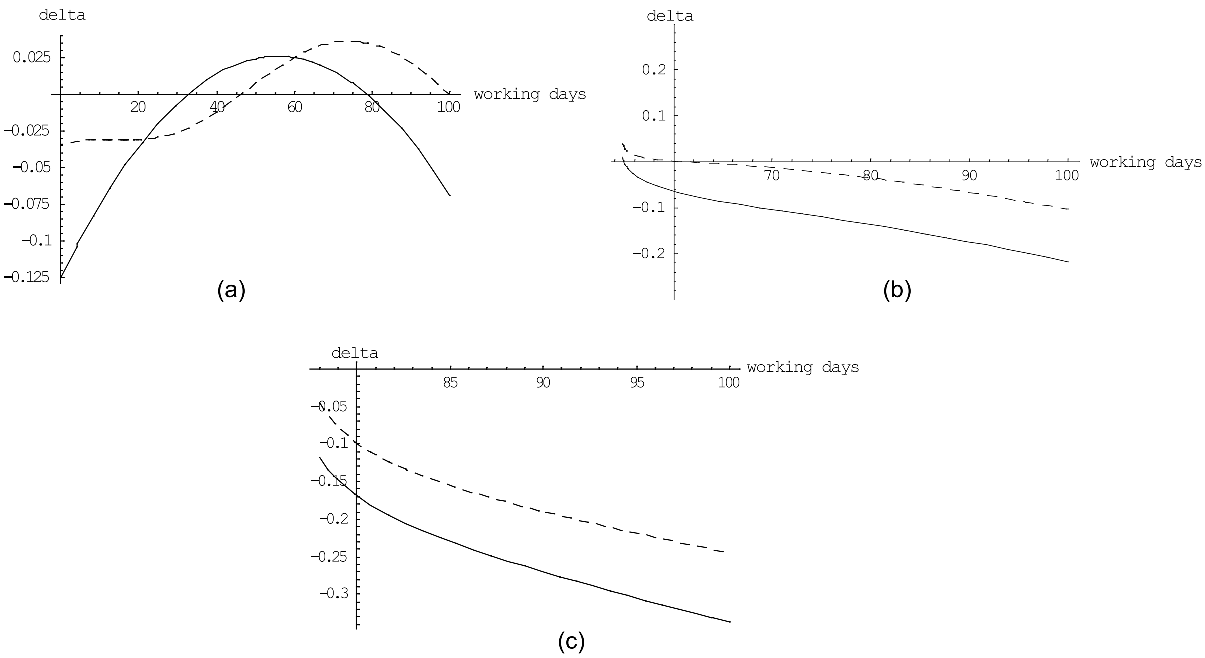

Fig. 7. Graphs depicting values of the Δ functions for utility equal to (a) – 0.5, (b) – 0.7, (c) – 0.9 in the original space based on a polynomial weighting function (solid line), and a Gaussian weighting function (dashed line).

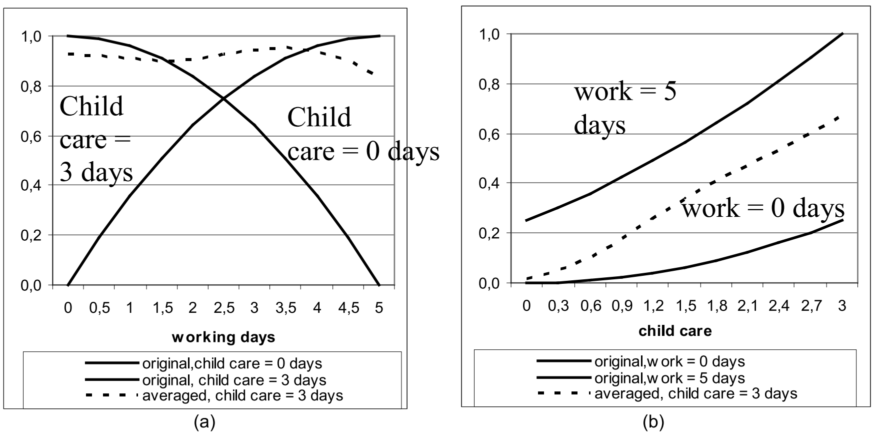

Fig. 8. Original and averaged utility values running through maximum Δ-point.

reasonably close to a particular utility value will be obtained. The method needs to be combined with a negotiation strategy to apply it in a bilateral negotiation. It does not by itself provide a guarantee that against arbitrary opponents a good negotiation outcome will be reached.

Although a transformation of the utility space may have an effect on the negotiation process as well as on the negotiation outcome, whenever the approximations computed by WAID are close enough to those computed by means of exhaustive search, it is immaterial with which strategy WAID is combined. WAID can be combined with any negotiation strategy that determines a next bid by searching for a bid with a particular utility. Such negotiation strategies include the Time-dependent and Behaviour-dependent

1 **Evaluate bid $bid_A(i)$ received from opponent A:**
Accept and end negotiation if $u_b(bid_A(i)) > u_b(bid_B(i))$
2 **Compute concession and target utility:**
Concession $\gamma = \beta * (1 - \mu / u_B(bid_B(i))) * (u_B(bid_A(i)) - u_B(bid_B(i))))$
Target utility: $\tau = u_B(bid_B(i)) + \gamma$
3 **Determine a next bid:**
3a **Find a bid with target utility**
Find a bid $bid_B(i+1)$ such that $u'_B(bid_b(i+1)) \approx \tau$
3b **Compare bid utility in approximated and original space**
Check whether $|u_B(bid_B(i+1)) - u'_B(bid_B(i+1))| < \delta$
3c **If not, find next candidate for the bid and repeat step (3b)**:
Find next candidate bid $bid_B(i+1)$ such that $u'_B(bid_B(i+1)) \approx \tau$
and utility with previous bid only differs minimally
4 **Else, send bid to opponent**

Fig. 9. Negotiation algorithm with bid search procedure.

tactics of [4], the Trade-off strategy of [5], the Bayesian learning agent of [8], the ABMP strategy of [10], and others. Here, we have used the ABMP-strategy proposed in [10].

To assess the impact of the WAID method, a negotiation strategy is applied to the employment contract example. The ABMP strategy, outlined in Fig. 9, is a concession-oriented negotiation strategy. It selects counter-offers without taking opponent knowledge into account. The ABMP strategy decides on a negotiation move based on considerations derived from the agent's own utility space only. It calculates a utility of a next offer based on the current utility gap between the last opponent's offer and the last own offer. The ABMP-strategy determines a bid in two steps: the strategy first (a) determines the target utility for the next bid, and then (b) determines a bid that has that target utility. The (b) part of the strategy is very efficient for independent utility spaces. For the purpose of comparison, however, we can use exhaustive search through the complete utility space to find a bid in the second step, provided that the space is discretized in a suitable manner (using small enough steps). In this way, the first step (a) in the ABMP-strategy followed by the second step (b) using exhaustive search can be applied to the original utility space whereas the original ABMP strategy can be applied to its approximation.

An additional check is incorporated into the strategy when the approximated utility space is used to avoid the risk of accepting bids with low utilities in the original space that have much higher utilities in the approximated space. The bids with high Δ-values, that have shifted significantly due to application of the averaging method, can be filtered out in this additional step. When the agent receives a bid from its opponent, the agent has to calculate the associated original utility as well and compare it with the bid acceptance threshold.

We propose a parameterized procedure that can be used to control the probability of large outcome deviations. The parameters of this procedure can moreover be used to influence the tradeoff between the accuracy of the negotiation outcome and the computational efficiency of the negotiation strategy. In the next sections, experimental results are presented that allow the tuning of these parameters.

In the negotiation algorithm the bid selection procedure is the source of the deviation of the negotiation outcome. In particular, in step 3 of the algorithm described in the previous section the approximated space is used instead of the original space which gives rise to outcome deviations. To avoid approximation errors that are too big, we propose to add a checking procedure in this step which compares the utility of a bid in the approximated space with the utility in the original space.

The proposed procedure integrated into the negotiation algorithm can be found in Fig. 9. The step to determine a next bid is refined and an iterative procedure is incorporated to check whether the difference in utility stays below a certain threshold δ. As before, in step 3a a bid is computed that matches a certain target utility. In step 3b, however, now a check has been incorporated that checks whether $\Delta U(bid) < \delta$, that is, whether the absolute approximation error stays below a threshold δ. This additional check itself is computationally cheap, since it involves only a simple calculation using Eq. (2). If $\Delta U(bid) > \delta$, a bid bid', which utility differs minimally from the previously computed bid, is searched for, until $\Delta U(bid') < \delta$. This iterative procedure for finding an appropriate bid is called δ-checking.

The additional check is used to avoid the risk of proposing bids with (very) low utilities in the original space that have (much) higher utilities in the approximated space. The concessions made in step 3 thus are controlled by a parameter δ to ensure that they are not too big.

The δ-checking procedure introduces additional search again into the computation of a bid. Various heuristics could be applied again, however, to minimize the amount of search. For example, a limit on the number of iterations could be introduced for spaces of high dimensionality to ensure a bid would be found within a reasonable amount of time. (The probability of finding an appropriate bid is high in high-dimensional spaces close to the m-point.) The relation of the value of the δ-parameter and the computational cost is analyzed in more detail using experimental results in Section 5.4.

5. Experimental evaluation

In this section, we present experimental results that show how the value of the δ-parameter in the checking procedure relates to the distribution of the outcome deviation. These results show that there is a direct relation between the size of δ and outcome distribution.

5.1. Overall negotiation behaviour in the case study

In this section, the negotiation strategy outlined in Fig. 9 is used to study the bids that an agent will offer during a negotiation using the original as well as the approximated utility space. The negotiation strategy that an agent decides to use should not only fit the agent's personality profile and culture, its experience in general and the current domain and negotiation partner, but it also has to be applicable given the utility space. For this case study we use the ABMP negotiation strategy.

In our experiments, the same profile of the employer was used in the original as well as in the approximated case. The employer's profile that has been used is the same as that introduced above.

Figure 10(a) (left) shows the outcome space as defined by the utilities of the employer and employee per bid. Each point on the chart represents one bid. The coordinates of the bid are the utilities of the opponents (x-coordinate is the employer's utility of the bid, y-coordinate is the employee's utility of the bid). The Nash product representing a bid with the highest utilities simultaneously for both opponents of the original utility space equals 0.53 and corresponds to a bid of 5 working days with 2.5 days of childcare, which satisfies the employee's constraints. The Kalai-Smorodinsky solution is 1.5 days of child care and 5 working days. This bid is found by locating a bid on the Pareto-optimal frontier, which is closest to the line drawn from points with utilities of (0; 0) to points with utilities (1; 1). This bid represents a negotiation outcome where both parties get the same utility. Using the ABMP strategy with exhaustive search for both parties, the negotiation lasts 4 rounds (4 bids from each side, the employer starts) and finishes when the employee accepts a bid of 2 days of childcare with 4.5 working days.

Figure 10(b) presents the result using the original ABMP strategy for both parties, where the profile of the employee has been approximated. The bids in the utility space are now concentrated around the employees original and approximated utility level of 0.7 (the m-point) with some spread towards

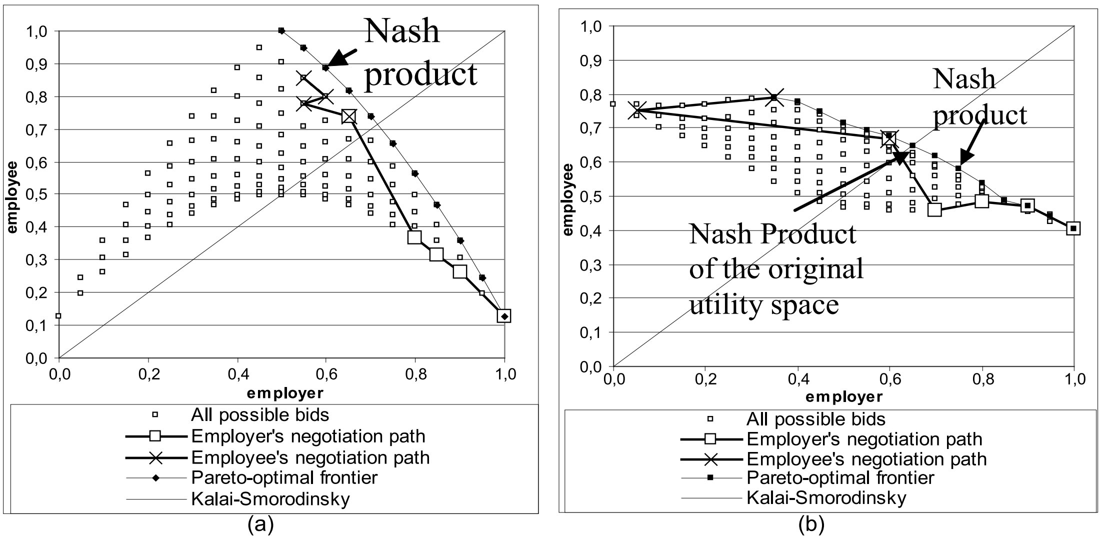

Fig. 10. Outcome space, optimality criteria, and negotiation paths (a) for the original utility space of the employee, and (b) for the approximated utility space of the employee.

lower utilities. The Nash product shifts to the bid of 5 working days and 1.5 days of childcare and the Kalai-Smorodinsky solution now is 4 working days and 1.5 days of childcare.

The original outcome space and the approximated one are significantly different. However, the difference is not critical for the negotiation itself due to the fact that most of the bids for which the difference is significant will not be used in a negotiation and we basically aim for the efficient solutions (Kalai-Somorindinsky point, and Nash Product). Also note that the bids are shifted only on the vertical axis (employee's utility), because the employer's profile remains the same. The negotiation performed for the same setup but using the approximated employee's utility space is also finished in 4 rounds as in the previous experiment and also results in a deal of 4.5 working days and 2 days of childcare. This example shows that the approximation procedure leads to some shifts in the efficient outcomes of the negotiation with respect to Nash and Kalai-Smorodinski. However, it also confirms that these bids and those around them preserve their meaning for the negotiator. Negotiation outcomes for both utility spaces are rather close even though the negotiation paths are different.

5.2. *Impact on outcome deviation*

To analyze the impact of the WAID method on the negotiation outcome and computation costs a probabilistic experimental setup has been used. The negotiation outcomes obtained by using the WAID are compared with those obtained using the original utility space. The experimental results are obtained from utility spaces modeled by multivariate quadratic polynomials. Such polynomials are widely-used in decision making theory to represent preferences of a decision maker, e.g. see [15]. They are more expressive then the classic multilinear or multiplicative forms studied by Keeney and Raiffa in [11] and, thus, can cover wide range of domains including scheduling, assignment, quality control, facility layout, computer-aided process planning, and others (see [15]). On the other hand, efficient methods for preference elicitation using polynomial representation exist and studied in [15]. These methods are based on the fact that the polynomial can be easily differentiate and the decision maker can use gradients of a polynomial function to assess its utilities.

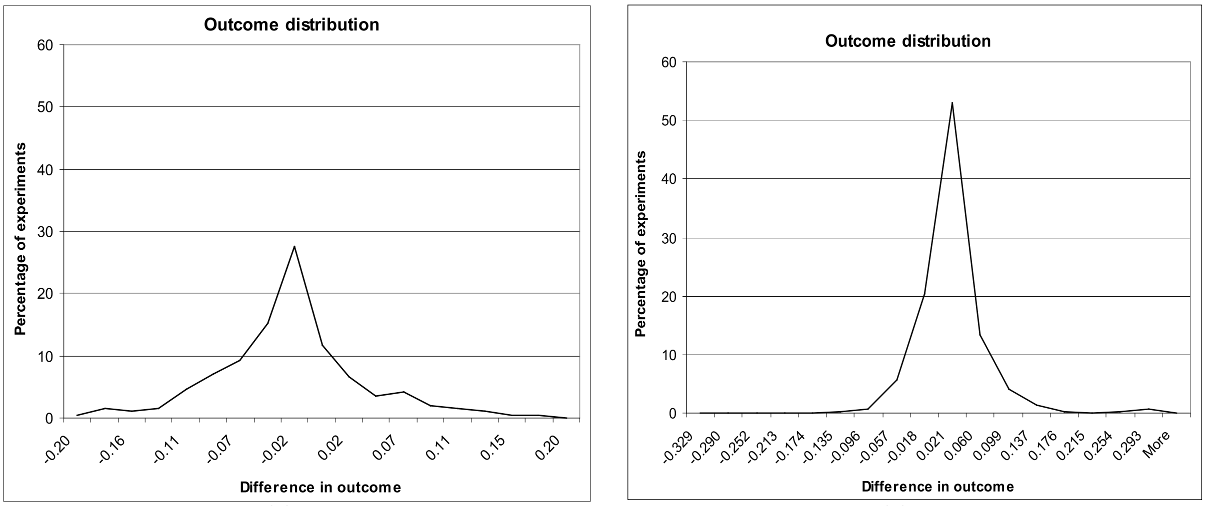

Fig. 11. Distribution of negotiation outcome deviation without checking procedure (a) and with the procedure (b) for approximated spaces vs. original spaces for 4 issues ($k = 15$).

These polynomials may have multiplicative terms $x_i x_j$ which represent issues:

$$u(x_1, x_2, \ldots, x_n) = \sum_{l}^{n} w_l \sum_{i=0}^{n} \sum_{j=0}^{n} a_{i,j} x_i x_j, \text{where } x_0 = 1 \tag{13}$$

Values of the coefficients $a_{i,j}$ are generated randomly from interval $[-1; 1]$. Then, the evaluation functions are normalized to interval of $[0; 1]$ using a Monte-Carlo method. It is well-known that solving such quadratic programming problems is NP-hard, see e.g. [7]. The m-point parameter that has to be fixed in order to apply the WAID-method is determined for each utility space by a Monte-Carlo method.

The ABMP negotiation algorithm [10] (see Fig. 9) is used to assess the outcome deviation that may occur when an approximated space is used instead of the original space during a negotiation. In the experiments that were performed agent A also uses a variant of the ABMP strategy but does not approximate any issue dependencies in its utility space. Instead it uses exhaustive search through its utility space in step 3 to determine a next bid given a suitable discretization of this space (i.e. using small enough steps).To compare outcomes for utility spaces of medium size, the same negotiation is performed again with agent B using exhaustive search in step 3. Of course, exhaustive search can only be used for utility spaces of medium size due to exponential time costs and memory limitations. It is, however, imperative to use it if we want to calculate outcome deviation. In the experiments, spaces with up to a number of 5 issues and a number of discretization steps of at most 25 have been used (see also Sections 5.2 and 5.3).

The main result of the experiments performed shows that the distribution of negotiation outcome deviations is similar to a normal distribution with a mean value close to zero. Figure 11 (left) presents the distribution of outcome deviations for a negotiation about 4 issues. The deviation is a result of using the approximated space in the negotiation strategy instead of performing an exhaustive search to find a good bid in the original space. We use absolute values of the deviation in terms of utility instead of percentages here to have uniform presentation of the parameters and results throughout the paper. Note, that evaluation and utility functions are normalized into interval of $[0; 1]$. The bell-shaped distribution on the figure (average $= -0.02$; std.dev. $= 0.09$) means that the negotiation over the approximated space tends to produce the same result as the negotiation over the original space using exhaustive search. This demonstrates that one may expect to obtain reasonable outcomes when negotiating with approximated spaces instead of non-approximated spaces.

Even though this result shows that approximating the original utility space to remove issue dependencies may result in quite reasonable outcomes compared to those obtained otherwise, it also shows that there is quite a high chance of deviating significantly. In fact, for the 4 issue case Fig. 11(a) shows that there is a quite high probability of obtaining outcomes that are worse by up to 20%. Additionally, the curve is not really symmetrical and shows a tendency towards negative deviations. As an illustration, the probability of obtaining a result that is worse than 10% equals 0.196. It is clear that in many domains such a high risk will be unacceptable.

The impact of adding the δ-checking procedure to the negotiation algorithm on the outcome distribution is significant, as is shown by Fig. 11(b). The experimental setup is exactly the same but the negotiation algorithm used by agent B now includes the checking procedure. It shows the outcome distribution for a threshold of $\delta = 0.01$.

Clearly, the outcome distribution of the plot in Fig. 11(b) is more symmetrical than in Fig. 11(a) and more clustered around the mean; it has a mean $= -0.00016$ and a standard deviation of 0.045. A more detailed analysis of the relation between Δ and the outcome deviation is presented in the Section 5.2.

The main conclusion thus is that additional measures need to be taken to reduce this risk. The benefit of using approximated spaces is clear: issues can be negotiated independently which makes the negotiation tractable. Controlled balance has to be found between the computational costs and the risk of significantly deviating negotiation outcomes.

Additionally, we investigated the influence of the discretization per issue under consideration on the outcome distribution. In the experiments we performed, the possible values for each issue were reduced by discretization the space to 10, 15, 20, and 25 values. In order to assess the impact of adding the checking procedure to the negotiation algorithm, we performed experiments with 3, 4, 5, and 6 issues. Finally, for the δ-parameter of the checking procedure we used the values 0.001, 0.005, 0.01, 0.02, 0.03, and 0.05. In total, we performed over 44.000 experiments in which the outcomes were compared with the original space: 12.000 for 3 issues, 12.000 for 4 issues, 12.000 for 5 issues, and 6.000 for 6 issues. Comparisons of negotiation outcome for spaces of higher dimensionality were not feasible. The higher the number of issues n and the higher the discretization parameter k, the longer it takes to do the exhaustive search (it takes kn steps). To investigate the scalability of the proposed approach, we ran in total 500 experiments with 7,8,9,10 and 15 issues for $\delta = 0.02$ and each k-value, so 2000 experiments in total. The results for 10 and 50 issues are presented in Section 5.4.

The experimental results relating the value of δ to the outcome distribution are depicted in Fig. 12. We do not show all results but only those for δ-values of 0.01, 0.02, and 0.03 which most clearly demonstrate the impact of different values on the distribution and also define the turning points where decreasing this parameter further does not have a very big impact anymore (see also Fig. 14) and decreasing it results in significantly worse outcomes.

In Fig. 12, on the x-axis the outcome difference is set out. The outcome deviation may be bigger than the value of the δ-parameter since errors may accumulate over multiple rounds in the negotiation. The y-axis refers to the percentage of experiments having particular outcome differences. The different lines correspond with different values of the discretization parameter k. For each combination of a particular number of issues, δ-value, and k-value, 500 experiments were run.

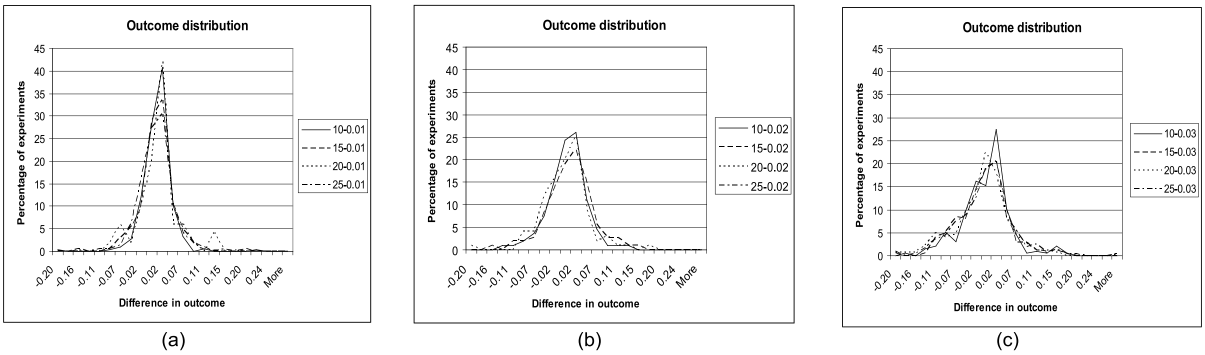

Fig. 12. The distribution of outcome deviations for 5 issues and $\delta = 0.01$ (a), $\delta = 0.02$ (b), $\delta = 0.03$ (c). The various lines relate to different k-values.

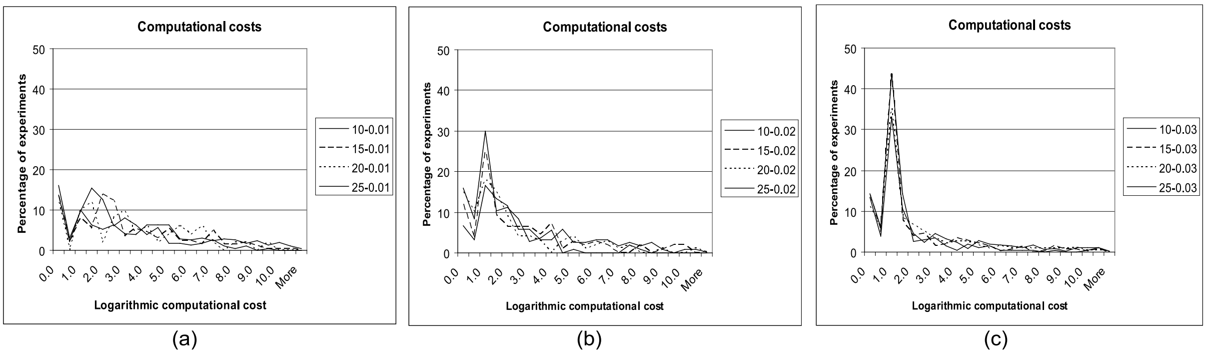

Fig. 13. Computational costs for 5 issues and $\delta = 0.01$ (a), $\delta = 0.02$ (b), $\delta = 0.03$ (c). The various lines relate to different k-values.

In general, as is to be expected since δ is supposed to control the error introduced by the approximation, the experimental findings show that smaller values for δ result in negotiation outcomes that are closer to the outcomes in the original space.

The findings illustrated in Fig. 12 are as follows. For $\delta = 0.01$ (see Fig. 12(a)) the standard deviation ranges from 0.0327 to 0.0442, and the average outcome difference ranges from -0.0066 to 0.0015. For $\delta = 0.02$ (see Fig. 12(b)) the standard deviation ranges from 0.0350 to 0.05806 and the average outcome difference ranges from -0.0142 to 0.0010. Finally, for $\delta = 0.03$ (see Fig. 12(c)) the standard deviation ranges from 0.0499 to 0.0717, and the average outcome difference ranges from -0.0199 to -0.0151.

5.3. *Impact on computational cost*

Including the checking procedure implies that the bid determination part might need iterations to find an appropriate bid. The previous section shows that smaller δ-values lead to better outcome deviations, and it stands to reason that the smaller the value, the higher the number of iterations needed. To get more insights into the frequency with which the need for iterations causes high computational costs, a series of experiments have been performed. The algorithm was tested for 4, 5, 6, and 10 issues, with the discretization value k varying over $\{10, 15, 20, 25\}$ and δ varying over $\{0.005, 0.001, 0.03, 0.02, 0.01\}$. Each test was performed 500 times with randomly generated original utility spaces.

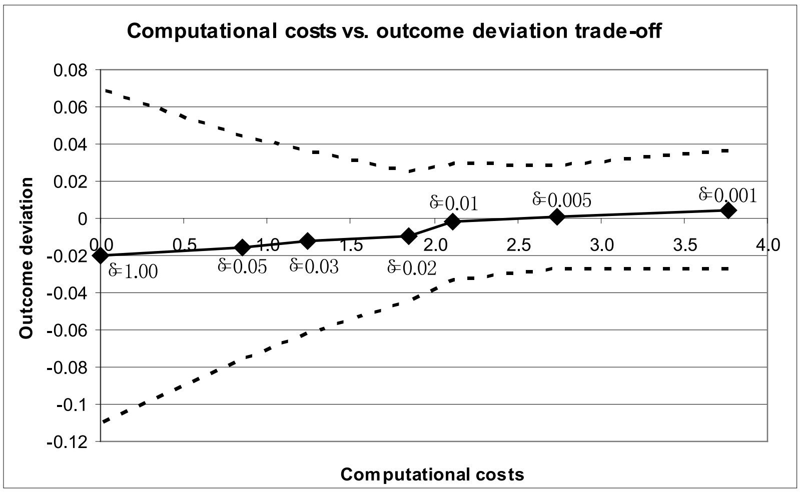

Fig. 14. Computational cost vs. outcome deviation for 5 issues and $k = 10$.

Figure 13 shows the results for 5 issues, the results for other values are not shown, since they do not provide additional insights. In these pictures, on the x-axis the logarithmic costs are set out. The y-axis refers to the frequency with which an experiment had such a logarithmic cost, with respect to the total number of experiments. The different lines refer to different k-values.

The results clearly show the expected increase of high computational costs for higher δ-values: higher percentages for higher computational values. However, when looking at the areas underneath the lines, another interesting observation can be made. In Fig. 13(a), for $\delta = 0.01$, the bulk of the area underneath the lines ends approximately at $ln(x) = 6$. In Fig. 13(b), for $\delta = 0.02$ the bulk ends at $ln(x) = 4$, and in Fig. 13(c), for $\delta = 0.03$ at $ln(x) = 2$. Evidently, the number of iterations needed is bounded.

5.4. Tuning computational costs

Combining the results of the outcome analysis of Section 5.2 and the computational cost analysis of Section 5.3 shows that the need for a small outcome difference has to be balanced against computational costs. In this a setting for the k, and δ parameters is chosen that balances accuracy against efficiency.

To find a good balance between accuracy and cost, an integrated analysis has been performed for the usual combination of parameters: the number of issues ranging over $\{4, 5, 6, 10\}$, k ranging over $\{10, 15, 20, 25\}$ and δ ranging over $\{0.001, 0.005, 0.01, 0.02, 0.03, 0.05, 1\}$. $\delta = 1$ corresponds to a setting without checking procedure.

Figure 14 presents the trade-off between negotiation outcome accuracy and the computational costs. Each point on the solid line of the chart represents the average of a series of experiments where δ varies over $\{0.001, 0.005, 0.01, 0.02, 0.03, 0.05, 1\}$. The top dashed line is an average+std.dev. and bottom dashed line is the average-std.dev.

The results show that a good compromise is a δ-value of 0.02: for $\delta < 0.02$ the costs increase, for $\delta > 0.02$ the outcome approximation gets worse. Furthermore, the standard deviation drops off at this value, but does not decrease further for even smaller δ-values.

To analyse the scalability of the modified negotiation algorithm we performed a series of negotiations with 10 issues. Exhaustive search as a benchmark for the negotiation is no longer possible due to the extremely large utility space. Figure 15 shows average of the computational cost depending on the

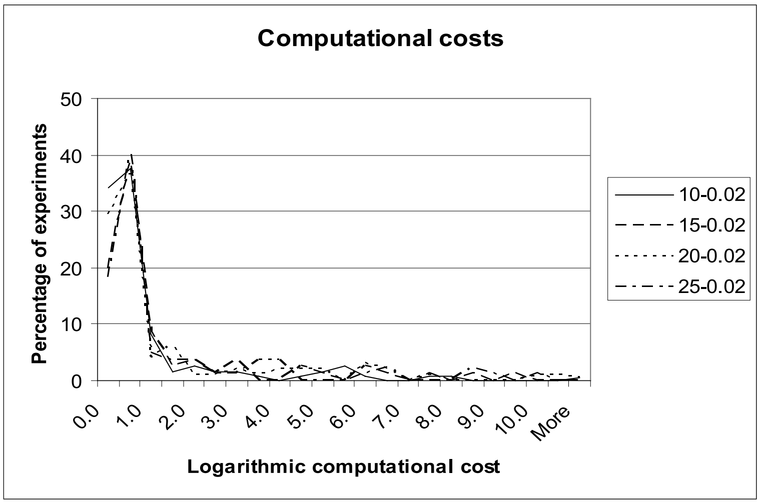

Fig. 15. Computational costs for 10 issues and $\delta = 0.02$. The different lines refer to different k-values.

number of issues for $\delta = 0.02$, $k = 20$. The figure suggests that the most of the randomly generated utility spaces remain tractable for the negotiation algorithm with the δ-checking procedure.

6. Conclusions

In this paper we introduced a new approach that allows agents to deal with complex utility functions in a negotiation environment with dependencies between issues. Instead of representing the negotiation task as an optimization task for nonlinear utility functions we propose an approximation method to simplify the agent's utility. The method is based on the observation that in common negotiation settings the expected negotiation outcome is approximately known. Furthermore, the insight that the nature of utility spaces for such common negotiation settings has enough structure to make our approach applicable. More insight is, however, required to assess the effects of using approximations of real preference profiles. Such an insight can help fine-tune and improve the approximation method by exploiting features of the real preference profiles.

The main advantage of the proposed method is that it enables applicability of a wider range of computational negotiation strategies without introducing a mediator into the negotiation. A number of negotiation strategies for negotiation domains without dependencies between issues have proven its efficiency. Such strategies cannot be directly applied to the negotiation domains with issues dependencies. Instead, the utility space approximated by means of the proposed method can by used.

Available information about the domain and the most probable negotiation outcome can be used to increase the accuracy of the method in the utility area around the expected outcome, which is most important for the negotiation. Such information can be obtained from various sources, such as a negotiation expert, domain knowledge, historical negotiations, and others. The more precise the knowledge about the expected outcome the better approximation of that region of the utiiltiy space.

The method provides means to analyze the impact of the approximation on a particular utility space, thereby making it possible to determine up front, whether or not the approximation is useful in any particular domain. If the approximation error is too big to be neglected another negotiation mechanism should be used, e.g. a mediator with nonlinear optimization techniques to find a fair negotiation outcome.

If the approximation in the region of the expected outcome is acceptable a negotiation strategy for domains with no issues dependencies can be used.

However, even when the approximation error of a particular negotiation space is acceptable in the region of the expected outcome using an approximation always comes with a risk. In the case of multi-issue negotiation, the risk is that a bid is proposed (and accepted by the other party) that seems to have a good utility, but in fact, in the original utility space has a much lower utility. The δ-checking procedure proposed in this paper offers a way to avoid this risk at the cost of additional computations. This check in itself is computationally cheap and ensures reasonable negotiation performance. Experimental results show that a tradeoff can be made between the accuracy of the bids and the computational overhead this entails. If the δ-parameter in the checking procedure is set to 0.02, the utility of the bids made is at most 0.02 away from the real utility, on a scale from 0 to 1. Moreover, using this value for the δ-parameter, the negotiation algorithm including the δ-checking procedure can handle high-dimensional utility spaces. the negotiation outcome obtained in this manner only slightly deviates from the outcome obtained without approximation.

To conclude, in this paper an effective balance is found of accuracy versus efficiency for multi-issue negotiation with issue dependencies in which the dependencies are removed by approximation. The bid search algorithm can be tunned according to the costs of an outcome deviation and availability of computational power for searching an offer.

In future research, we want to identify in more detail which classes of utility functions and types of dependencies can be approximated by weighted averaging sufficiently accurate. Another interesting direction for research would be a modeling experiment with humans, to gain a better understanding of the nature of the complexity of human preferences and the ways in which humans simplify the negotiation task.

References

[1] Y. Bar-Yam, Dynamics of complex systems. Addison-Wesley, 1997.

[2] R. Davies and R.G. Smith, Negotiation as a methaphor for distributed problem solving, *Artificial Intelligence* **20**(1) (1983), 63–109.

[3] J.S. Dyer, P.C. Fishburn, R.E. Steuer, J. Wallenius and S. Zionts, Multiple criteria decision making, multiattribute utility theory: The next ten years, *Management Science* **38**(5) (May 1992), 645–654.

[4] P. Faratin, C. Sierra and N.R. Jennings, Negotiation decision functions for autonomous agents, *Int. Journal of Robotics and Autonomous Systems* **24**(3–4) (1998), 159–182.

[5] P. Faratin, C. Sierra and N.R. Jennings, Using similarity criteria to make negotiation trade-offs, *Journal of Artificial Intelligence* **142**(2) (2003), 205–237.

[6] S. Fatima, M. Wooldridge and N. Jennings, An analysis of feasible solutions for multi-issue negotiation involving nonlinear utility functions. In Sierra Decker, Sichman and Castelfranchi, editors, Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009), May 2009.

[7] P. Favati, G. Lotti and F. Romani, Theoretical and practical e¡ciency measures for symmetric interpolatory quadrature fromulas, *BIT Numerical Mathematics* **34**(4) (1994).

[8] K. Hindriks and D. Tykhonov, Opponent modelling in automated multi-issue negotiation using bayesian learning. In Proceedings of the AA-MAS 2008, 2008.

[9] T. Ito, Hiromitsu Hattori, and Mark Klein. Multi-issue negotiation protocol for agents: Exploring nonlinear utility spaces. In Proc. of IJCAI-2007, 2007, pp. 1347–1352.

[10] C.M. Jonker, V. Robu and J. Treur, An agent architecture for multiattribute negotiation using incomplete preference information, *Journal of Autonomous Agents and Multi-Agent Systems* **15** (2007), 221–252.

[11] R.L. Keeney and H. Raiffa, Decisions with Mutliple Objectives. Cambridge University Press, 1976.

[12] M. Klein, P. Faratin, H. Sayama and Y. Bar-Yam, Negotiating complex contracts. Paper 125 of the Center for eBusines@ MIT. http://ebusiness.mit.edu., 2001.

[13] M. Klein, P. Faratin, H. Sayama and Y. Bar-Yam, Negotiating complex contracts, *Group Decision and Negotiation Journal* **12** (2 November 2003).

[14] G. Lai, C. Li, K. Sycara and J.A. Giampapa, Literature review on multi-attribute negotiations. Technical Report CMU-RI-TR-04-66, Robotics Institute, Carnegie Mellon University, December 2004.
[15] B. Malakooti and S. Subramanian, Generalized polynomial decomposable multiple attribute utility functions for ranking and rating multiple criteria discrete alternatives, *Applied Mathematics and Computation* **106**(1) (December 1999), 69–102.
[16] M.J. Osborne and A. Rubinstein, A Course in Game Theory. MIT Press, 1994.
[17] D.G. Pruitt, Negotiation Behavior. Academic Press, 1981.
[18] H. Raiffa, The Art and Science of Negotiation. Belknap Press of Harvard Univercity Press, Cambridge, Mass., 1982.
[19] H. Raiffa, J. Richardson and D. Metcalfe, Negotiation Analysis: The Science and Art of Collaborative Decision Making. Harvard University Press, 2003.
[20] V. Robu, D.J.A. Somefun and J.A. La Poutre, Complex multi-issue negotiations using utility graphs. In Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AA-MAS'05), Utrecht, The Netherlands, 2005, pp. 280–287.
[21] J.S. Rosenschein and G. Zlotkin, Rules of encounters Designing conventions for automated negotiation among computers. Cambridge, MA: MIT Press, 1994.
[22] C. Sierra, P. Faratin and N.R. Jennings, A service-oriented negotiation model between autonomous agents. in: *Proceedings of the 8th European Workshop on Modelling Autonomous Agents in Multi-Agent World, MAAMAW97*, M. Boman and W. van de Velde, eds, volume 1237 of Lecture Notes in Artificial Intelligence, Springer–Verlag, 1997, pp. 17–35.
[23] K. Somefun, T.B. Klos and H.L. Poutre, Online learning of aggregate knowledge about nonlinear preferences applied to negotiating prices and bundles, in: *Proceedings International Conference on Electronic Commerce*, M. Janssen, H.G. Sol and R.W. Wagenaar, eds, ACM, 2004, pp. 361–370.
[24] L. Thompson, The Mind and Heart of the Negotiator. Prentice Hall, 2004.
[25] J. Wallenius, J. Dyer, P. Firshburn, R. Steuer, S. Zionts and K. Deb, Multiple criteria decision making and multiattribute utility theory: Recent accomplishments and what lies ahead, *Management Science* **54** (2008).