

Tractable Negotiation in Tree-structured Domains

Yann Chevaleyre¹, Ulle Endriss² and Nicolas Maudet¹

¹LAMSADE, Université Paris-Dauphine, France

²ILLC, University of Amsterdam, The Netherlands

¹{chevaley,maudet}@lamsade.dauphine.fr, ²ulle@ilic.uva.nl

ABSTRACT

Multiagent resource allocation is a timely and exciting area of research at the interface of Computer Science and Economics. One of the main challenges in this area is the high complexity of negotiation. In particular, the complexity of the task of identifying rational deals, *i.e.* deals that are beneficial for all participants, often hinders the successful transfer of theoretical results to practical applications. To address this issue, we propose several protocols designed to tame the complexity of negotiation by exploiting structural properties of the utility functions used by agents to model their preferences over alternative bundles of resources. In particular, we consider domains where utility functions are *k-additive* (that is, synergies between different resources are restricted to bundles of at most *k* items) and *tree-structured* in the sense that the bundles for which there are synergies do not overlap. We show how protocols exploiting these properties can allow for drastically simplified negotiation processes.

Categories and Subject Descriptors

I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence—*Multiagent systems*; J.4 [Social and Behavioral Sciences]: Economics

General Terms

Economics, Theory

Keywords

Multiagent resource allocation, Negotiation

1. INTRODUCTION

Multiagent resource allocation is a timely and exciting area of research at the interface of Computer Science and Economics. Applications range from electronic commerce [19], over logistics [12], to the exploitation of earth observation satellites [13]. In this paper, we are going to be concerned

with a framework where agents negotiate over a finite number of indivisible resources in a distributed manner [9, 18]. Agents are assumed to only negotiate deals that are beneficial for each of the parties involved (which may be more than two). We are interested in negotiation histories that lead to allocations that are optimal from a social point of view; and we consider an allocation optimal if it maximises the sum of utilities of the agents in the system, *i.e.* if it maximises (utilitarian) social welfare [1].

Unfortunately, the complexity associated with the task of identifying rational deals, *i.e.* deals that are beneficial for all participants, is typically very high and often hinders the successful transfer of theoretical results to practical applications. Common approaches to tame this type of complexity are to restrict the range of utility functions agents can use to model their preferences and to exploit the structure of utility functions during negotiation [4, 15, 17]. This is also the approach we are going to follow in this paper. Concretely, we are going to discuss so-called *tree-structured domains*, where the utility functions of agents are subject to certain structural restrictions. We use a compact representation of utility functions based on the notion of *k-additivity*, which restricts the synergies between different resources to bundles of at most *k* items each [11]. Tree-structured domains are sets of *k-additive* utility functions where the bundles for which there are synergies do not overlap.

We then propose two protocols that restrict negotiation to deals involving only the smallest bundles first, and then incrementally bigger bundles. However, these guided negotiation processes are not guaranteed to converge to an allocation with maximal social welfare, *i.e.* the protocols need to be enhanced further. The first protocol we propose is made *intrusive* in the sense that it modifies the structure of agents' utility functions. The second protocol has a centralised flavour and introduces a *system agent* that can *compensate* for a temporary loss in utility. This combination of *guidance* and *social compensation* allows us to restore the convergence properties of the negotiation framework.

Paper Overview

In the remainder of this introduction we briefly review the multilateral negotiation framework we take as the basis for our work [9, 18]. We then exemplify the high complexity of the framework, in Section 2, by analysing the number of deals that agents need to consider at each stage during a negotiation process. The class of tree-structured utilities is introduced in Section 3. Section 4 then presents a number of protocols exploiting these structural properties of util-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AAMAS'06 May 8–12 2006, Hakodate, Hokkaido, Japan.
Copyright 2006 ACM 1-59593-303-4/06/0005 ...\$5.00.

ity functions, and shows how they reduce the complexity of the negotiation process. Our theoretical results are complemented by a brief experimental analysis of the protocols. Section 5 concludes with a discussion of the benefits and drawbacks of the approach advocated in this paper.

Multilateral Negotiation over Indivisible Resources

We briefly review the negotiation framework originally introduced by Sandholm [18], albeit in the context of task allocation problems (see [9] for a detailed introduction). Let \mathcal{A} be a finite set of *agents* and let \mathcal{R} be a finite set of indivisible *resources* that cannot be shared. An *allocation* A is a partitioning of the resources in \mathcal{R} amongst the agents in \mathcal{A} . Each agent $i \in \mathcal{A}$ is equipped with a *utility function* $u_i : 2^{\mathcal{R}} \rightarrow \mathbb{R}$. We write $u_i(A) = u_i(A(i))$ for the utility assigned by agent i to the bundle it receives in allocation A . A *deal* may involve the reallocation of any number of resources between any number of agents. We therefore specify deals $\delta = (A, A')$ simply as pairs of allocations (with $A \neq A'$). A deal may be accompanied by a monetary *side payment*. We write p_i for the amount paid by agent i (negative in case i receives money). The sum of side payments associated with any given deal has to add up to 0. A deal $\delta = (A, A')$ with side payments p is called *individually rational* iff $u_i(A') - u_i(A) - p_i > 0$ for each agent i involved in the deal (and $p_i = 0$ otherwise). Agents are assumed to only negotiate such individually rational deals.

To assess the overall quality of an allocation, we use the concept of (utilitarian) *social welfare* [1]: $sw(A) = \sum_{i \in \mathcal{A}} u_i(A)$. It is well-known that a deal is individually rational iff it increases social welfare [9]. It is also known that any sequence of individually rational deals will eventually result in an allocation with maximal social welfare [18]. However, this *convergence result* only applies in case agents can negotiate deals involving any number of agents and resources. A recent result has identified the class of *k-deals* (involving at most k resources each) as sufficient to guarantee convergence to an optimal allocation in case all agents use *additively k-separable* utility functions to model their preferences, *i.e.* in case \mathcal{R} can be divided into several preferentially independent bundles of at most k items each [4].

2. NUMBER OF BOUNDED DEALS

Results that identify conditions under which the convergence to a socially optimal allocation remains possible when agents only negotiate deals involving at most k items each significantly reduce the complexity of negotiation (at least in cases where k is small). In particular, the restriction to k -deals significantly reduces the number of deals to be considered at any given stage. In this sense, the quoted result on additively k -separable domains [4] is a first step towards a realistic multilateral negotiation system. However, even in such favourable cases it may be infeasible for agents to identify a k -deal that is acceptable to all parties involved. In this section, we are going to demonstrate this problem by analysing the number of bounded deals that could be implemented in a given allocation.

Recall that $|\mathcal{A}|$ is the number of agents and $|\mathcal{R}|$ is the number of resources in the system. This means that there are $|\mathcal{A}|^{|\mathcal{R}|}$ possible allocations overall (for each resource, we can choose which agent to give it to). Hence, in any given situation there are $|\mathcal{A}|^{|\mathcal{R}|} - 1$ potential deals to consider (we

could move to any alternative allocation). This number can, of course, be somewhat reduced, for instance by discounting any previous allocations, but the remaining number of potential deals that need to be checked for their mutual acceptability would still be prohibitively high.

Bounded Number of Resources

The situation improves significantly if we restrict ourselves to bounded deals involving at most k distinct resources. There are $\binom{|\mathcal{R}|}{k}$ ways of selecting k items from the full set \mathcal{R} . Then each of the chosen k items can be moved to any of the $|\mathcal{A}|$ agents in the system. If the agent receiving a given item is the same agent where that item resided before the deal, then we take this as an encoding for the item not being moved at all, *i.e.* we are also capturing those deals that involve fewer than k resources. Only one of these cases needs to be excluded, namely the case where not a single of the k chosen resources gets moved (as this would not count as a valid deal). Hence, we end up with the following overall number of possible k -deals for any given allocation A :

$$\binom{|\mathcal{R}|}{k} \cdot (|\mathcal{A}|^k - 1) \quad (1)$$

This is polynomial in both the number of agents and the number of resources, while it has been exponential in the number of resources for deals without structural restrictions. In the case of $k = 1$, for instance, we only have to consider $|\mathcal{R}| \cdot (|\mathcal{A}| - 1)$ potential deals from any given allocation.

Bounded Number of Agents

In case we restrict the *number of agents* involved in a deal, the number of possible deals is not constant anymore, but varies depending on the current distribution of goods. To see this, we first compute the number of deals that are possible within a subgroup $S \subseteq \mathcal{A}$ of agents for a given allocation A . For any given agent $i \in S$, and any resource $r \in A(i)$ currently held by i , i can choose to which agent within the group S the item r should go (or whether it should remain with i). This makes $|S|^{|A(i)|}$ choices for agent i . The same is true for every agent in S (and in the end we still need to subtract 1 to account for the case where not a single resource is moving). We end up with the following formula:

$$|S|^{\sum_{i \in S} |A(i)|} - 1 \quad (2)$$

This number is constant as long as we do not change the overall number of resources held by the agents in S . For a group of two agents $S = \{i, j\}$, for instance, the above formula simplifies to $2^{|A(i)| + |A(j)|} - 1$. Hence, the number of *bilateral* deals (involving only two agents) that are implementable given allocation A can be computed as follows:

$$\sum_{i \neq j \in \mathcal{A}} (2^{|A(i)| + |A(j)|} - 1) \quad (3)$$

Now this number is not constant anymore; it depends on the distribution of resources in allocation A . Take a scenario with 3 agents and 9 resources. If each agent holds 3 resources, then the above formula shows that there are $3 \cdot (2^{3+3} - 1) = 189$ potential bilateral deals. If, on the other hand, we start with an allocation where one of the agents holds all 9 resources to begin with, then there are $2 \cdot (2^{9+0} - 1) + (2^{0+0} - 1) = 1022$ bilateral deals. Indeed, formula (3) shows that the number of possible deals increases as

resources become less evenly distributed amongst the agents in the system. It also shows that, even if we only allow for bilateral deals, the number of deals to be considered is exponential in $|\mathcal{R}|$, the overall number of resources, in case the distribution of resources is highly unbalanced. If the distribution of resources is reasonably well balanced, on the other hand, then the number of possible bilateral deals is “only” exponential in $\frac{|\mathcal{R}|}{|\mathcal{A}|}$, which may be manageable in cases where the number of resources is of a similar order of magnitude as the number of agents.

In conclusion, our discussion in this section has given some insight into the factors that determine the number of deals to be considered in each negotiation round and we have seen that the restriction to deals involving only a limited number of resources can result in a significant reduction in complexity. If we restrict deals to those between a bounded number of agents, the positive effect is considerably less significant. In either case, the number of deals to be considered typically remains very high and makes the design of general strategies for multilateral negotiation particularly challenging.

3. TREE-STRUCTURED DOMAINS

In this section, we are going to introduce a restriction on utility functions which we call *tree-structured* utilities. This is a natural restriction for many applications, which will allow us to drastically reduce the search space without losing the convergence properties.

The k -additive Form

An important parameter in the specification of any multi-agent resource allocation problem is the way we represent the preferences of individual agents over alternative bundles of resources. A natural and compact language is the so-called *k -additive form* of representing utility functions. A utility function u_i is k -additive iff there exists a set of *coefficients* $\{\alpha_i^T \mid T \subseteq \mathcal{R}\}$ such that $\alpha_i^T = 0$ whenever $|T| > k$, and the following equality holds for all bundles $R \subseteq \mathcal{R}$:

$$u_i(R) = \sum_{T \subseteq R} \alpha_i^T \quad (4)$$

That is, agent i enjoys an increase in utility of α_i^T when it owns all the items in T together, *i.e.* α_i^T represents the synergetic value of this bundle. If a utility function is defined in terms of such coefficients, we say that it is given in k -additive form. When describing examples, we are going to use a simplified notation. For instance, $u_i = 5.r_1 + 3.r_2.r_3$ represents a 2-additive utility function with two non-zero coefficients: $\alpha_i^{\{r_1\}} = 5$ and $\alpha_i^{\{r_2, r_3\}} = 3$.

The k -additive form goes back to the work of Grabisch [11] in fuzzy measure theory and has recently been advocated as a useful preference representation language in the context of both combinatorial auctions [5] and distributed negotiation schemes [3]. Similar preference languages have been used in Artificial Intelligence for some time [2]; what is special about the k -additive form is that it parametrises the degree of synergies in a natural manner. If k can be chosen freely, then *any* utility function can be written in k -additive form, and the k -additive form is typically significantly more *concise* than simply enumerating the utilities for all bundles [3]. For ease of presentation, we are going to assume that the coefficient for the empty bundle $\alpha_i^{\{\}} = 0$ for all agents $i \in \mathcal{A}$

(but the applicability of our results is not affected by this assumption).

Terms and Trees

To define tree-structured domains, we require utilities to be represented in k -additive form. The set of *terms* of a utility function u_i is the set of bundles T with a non-zero coefficient α_i^T in the k -additive form. For instance, the utility function $u_i = 16.r_2 + 5.r_1.r_2.r_4 - 2.r_1.r_3$ uses the terms r_2 , $r_1.r_2.r_4$, and $r_1.r_3$. Let $\{u_1, \dots, u_n\}$ be the set of utility functions of the agents in the system. \mathcal{T} will denote the set of all terms appearing in the representation of any of the utility functions u_1, \dots, u_n , and α_i^T will denote the coefficient of term T in u_i . Finally, \mathcal{T}^l denotes the set of terms in \mathcal{T} consisting of exactly l resources, while $\mathcal{T}^{\leq l}$ denotes the set of terms in \mathcal{T} with at most l resources each.

Intuitively, tree-structured utilities are k -additive functions in which there are no “overlapping” terms.

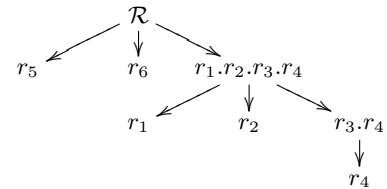
DEFINITION 1. A set of utility functions $\{u_1, \dots, u_n\}$ is called *tree-structured* iff, when represented in k -additive form, it is the case that for all terms $T_1, T_2 \in \mathcal{T}$ we have either $T_1 \subseteq T_2$, or $T_1 \supseteq T_2$, or $T_1 \cap T_2 = \{\}$.

It is also helpful to note that a set of utility functions is tree-structured iff the terms of \mathcal{T} can be represented by a tree, in which \mathcal{R} is the root, and each term is a node. Branches of the tree represent the \subseteq relation. The following example illustrates this representation.

EXAMPLE 1. Consider the set $\{u_1, u_2, u_3\}$ of utility functions defined as follows:

$$\begin{aligned} u_1 &= r_2 + 3.r_5 \\ u_2 &= 3.r_1 + 10.r_1.r_2.r_3.r_4 + 8.r_5 + 4.r_6 \\ u_3 &= r_6 - r_4 + 8.r_3.r_4 \end{aligned}$$

These functions are 4-additive as well as tree-structured. The set of terms $\mathcal{T} = \{r_1, r_2, r_4, r_5, r_6, r_3.r_4, r_1.r_2.r_3.r_4\}$ can also be represented by the following tree:



The idea of tree-structured domains has also been used by Rothkopf *et al.* [17] in the context of combinatorial auctions. In their work, bids are restricted to bundles that belong to a given tree structure. They show that this restriction makes the winner determination problem tractable. We are going to exploit this reduced level of complexity as well, although by designing a negotiation protocol rather than an optimisation algorithm.

We note here that any set of k -additive tree-structured utilities is also additively k -separable. Hence, the convergence towards an allocation with maximal social welfare in case agents only negotiate individually rational k -deals is guaranteed by the aforementioned result [4].

\mathcal{T} -deals

Still, as argued in Section 2, even finding a rational k -deal can often be computationally infeasible. Our aim is therefore to exploit this new restriction on utility functions to

come up with a new type of deal that is less complex, while still ensuring the same convergence properties.

DEFINITION 2. Let \mathcal{T} be a set of terms. A \mathcal{T} -deal is a deal involving the reallocation of the resources of a single term in \mathcal{T} from any number of senders to a single receiver.

For our example above, for instance, moving all of r_1 , r_2 , r_3 and r_4 from whichever agent currently holds them to, say, agent 1 would be a valid \mathcal{T} -deal. Only moving r_3 , on the other hand, would not constitute a \mathcal{T} -deal (because r_3 does not occur as a term in the representation of the utility function any of our three agents).

How many \mathcal{T} -deals are possible from a given allocation? The exact number will depend on the particular allocation in question, but $|\mathcal{T}| \cdot |\mathcal{A}|$ will certainly be an upper bound: for any term in \mathcal{T} we can choose which of the agents in \mathcal{A} should receive that term in its entirety. How many terms can we have in \mathcal{T} in the worst case? Firstly, the number of non-overlapping terms of size k is bound by $\lceil \frac{|\mathcal{R}|}{k} \rceil$. Secondly, given a set of k resources, we can use these to construct a subtree with at most $2k - 1$ nodes (to see this, consider that splitting each term/node into two balanced subnodes will give a tree exhibiting the maximal number of nodes). Hence, we end up with the following upper bound for the number of terms in \mathcal{T} for any set of k -additive tree-structured utility functions:

$$|\mathcal{T}| \leq (2k - 1) \cdot \left\lceil \frac{|\mathcal{R}|}{k} \right\rceil \quad (5)$$

Hence, even if we do not put any restrictions on k (*i.e.* in case $k = |\mathcal{R}|$), we end up with an upper bound for the number of possible \mathcal{T} -deals of $(2 \cdot |\mathcal{R}| - 1) \cdot |\mathcal{A}|$. This number is of the same order of magnitude as the number of 1-deals available from any given allocation, and significantly lower than the number of possible k -deals for any $k > 1$. In other words, the complexity of finding rational \mathcal{T} -deals is also very low compared to that of finding rational k -deals.

However, as illustrated by our next example, simply restricting negotiation to individually rational \mathcal{T} -deals will not be sufficient to guarantee that an allocation with maximal social welfare will be reached eventually in all cases.

EXAMPLE 2. Consider the following utility functions:

$$\begin{aligned} u_1 &= 10.r_1 \\ u_2 &= 10.r_2 \\ u_3 &= 11.r_1.r_2 \end{aligned}$$

Furthermore, let the initial allocation be the allocation assigning all items to agent 3. Then the (only) optimal allocation gives r_1 to agent 1 and r_2 to agent 2. Clearly, there are no rational \mathcal{T} -deals allowing the system to reach this optimal allocation. For any of these deals, the utility of agent 3 would drop from 11 to 0, while the receiving agent (either 1 or 2) would only gain 10 utility points. Hence, it would not be possible to arrange for suitable side payments making such a deal beneficial for everyone involved.

In order to restore the desired convergence properties of the framework, we will need to investigate more complex protocols restricting the negotiation process and exploiting, through \mathcal{T} -deals, the properties of tree-structured utilities.

4. TREE-CLIMBING PROTOCOLS

The basic idea of *tree-climbing protocols* is to allow \mathcal{T} -deals involving only the smallest bundles first, then to incrementally allow bigger bundles, until all possible \mathcal{T} -deals have been tried. The *Naive Tree-Climbing Protocol* is a simple implementation of this principle.

Algorithm 1 Naive Tree-Climbing Protocol

Require: n agents with tree-structured utilities u_1, \dots, u_n (with \mathcal{T} the set of all terms)

- 1: $l \leftarrow 1$.
- 2: **repeat**
- 3: Restrict allowed deals to \mathcal{T}^l -deals.
- 4: Let agents negotiate individually rational \mathcal{T}^l -deals.
- 5: **if** no more deal can be conducted **then**
- 6: $l \leftarrow l + 1$
- 7: **end if**
- 8: **until** $l > |\mathcal{R}|$

Clearly, this protocol is more restrictive than the liberal “protocol” consisting of allowing any \mathcal{T} -deals, *i.e.* it cannot guarantee an optimal outcome either (see Example 2). By inspecting more carefully the scenario described in Example 2, however, we find clues indicating possible improvements of this protocol: During the first step of the protocol (where $l = 1$, and only \mathcal{T}^1 -deals are allowed), the term $r_1.r_2$ in u_3 prevents the resources from moving away from agent 3. If, during this first step, terms containing more than one resource were to be *removed* from the utility functions, then the resources would be able to move towards agents 1 and 2.

4.1 An Intrusive Tree-Climbing Protocol

Based on this idea, we design a new protocol which, during round l , removes from all utility functions any terms involving more than l resources. This protocol is called the *Intrusive Tree-Climbing Protocol* (ITCP). It is intrusive in the sense that it modifies the utility functions of each agent during each round of the protocol. In the specification of the ITCP, $u_i^{\leq l}$ denotes the utility function u_i with all terms of size greater than l having been removed. For instance, if $u_i = 12.r_1 + 3.r_1.r_2 + r_1.r_3 - 2.r_1.r_2.r_4$, then $u_i^{\leq 1} = 12.r_1$, $u_i^{\leq 2} = 12.r_1 + 3.r_1.r_2 + r_1.r_3$, and $u_i^{\leq 3} = u_i$.

Algorithm 2 Intrusive Tree-Climbing Protocol (ITCP)

Require: n agents with tree-structured utilities u_1, \dots, u_n (with \mathcal{T} the set of all terms)

- 1: $l \leftarrow 1$.
- 2: **repeat**
- 3: Restrict allowed deals to \mathcal{T}^l -deals,
- 4: Enforce all agents i to change their utilities to $u_i^{\leq l}$.
- 5: Let agents negotiate individually rational \mathcal{T}^l -deals.
- 6: **if** no more deal can be conducted **then**
- 7: $l \leftarrow l + 1$
- 8: **end if**
- 9: **until** $l > |\mathcal{R}|$

Convergence

As we shall see, the ITCP guarantees convergence to an allocation with maximal social welfare, provided the utility functions of all agents are tree-structured with *positive* coefficients. We are first going to show that each round of the

protocol will lead to an allocation deemed optimal wrt. the truncated utility functions of agents populating the system. The social welfare of an allocation A wrt. truncated utility functions in round l is denoted by $sw^{\leq l}(A) = \sum_{i \in \mathcal{A}} u_i^{\leq l}(A)$.

LEMMA 1. *For any set of tree-structured utilities with positive coefficients, if at the end of round l of the ITCP an optimal allocation wrt. $sw^{\leq l}$ has been reached, then at the end of round $l+1$ an optimal allocation wrt. $sw^{\leq l+1}$ will be reached.*

PROOF. Assume after round l the allocation A^l has been reached and that allocation is optimal wrt. $sw^{\leq l}$. Let A^{l+1} be (any of) the allocation(s) that are optimal wrt. $sw^{\leq l+1}$. Because of the tree structure, we can consider the optimal allocation of resources belonging to distinct terms in \mathcal{T}^{l+1} separately. Let $T \in \mathcal{T}^{l+1}$. We can distinguish three cases: (1) If T is owned by a single agent in A^{l+1} , then we are done, because this can be achieved using a \mathcal{T}^{l+1} -deal. (2) If not, and if T is also not owned by a single agent in A^l , then we are done, because the optimal allocation will be the same as in A^l . (3) If it were the case that T is owned by a single agent in A^l but not in A^{l+1} , then that would pose a problem, because the necessary reallocation would not be a \mathcal{T}^{l+1} -deal. However, this situation is impossible due to our restriction to positive coefficients. \square

We are now in a position to prove that the ITCP does indeed guarantee the intended convergence property.

THEOREM 1. *For any set of tree-structured utilities with positive coefficients, following the ITCP will result in an allocation of resources with maximal social welfare.*

PROOF. The proof is by induction over the number of rounds. In the first round, all agents have additive utility functions, and it has been shown elsewhere that negotiation with individually rational deals involving only a single resource each will converge to an optimal allocation in additive domains [9]. Lemma 1 shows that the property of having maximal social welfare wrt. truncated utility functions persists from one round to the next. Hence, after the final round, the allocation reached will have maximal social welfare wrt. the full utility functions. \square

Negative Coefficients

Observe that our proof of Lemma 1 shows why we need to restrict ourselves to k -additive functions with *positive* coefficients. If there are negative coefficients, then also deals that “destroy” an existing term can increase utility. This is also demonstrated by the following example.

EXAMPLE 3. *Consider the following utilities:*

$$\begin{aligned} u_1 &= 2.r_1.r_2 + 10.r_3 - 100.r_1.r_2.r_3 \\ u_2 &= 5.r_3 \end{aligned}$$

After round 2 of the ITCP, agent 1 would own all three resources, and the social welfare relative to terms of length ≤ 2 would be 12. In round 3, the only individually rational deal involving a term with exactly 3 resources would be to move $r_1.r_2.r_3$ from agent 1 to agent 2 (increasing social welfare from -88 to 5). But then the deal of moving r_3 back to agent 1 would increase social welfare even further, namely from 5 to 10 (which is the maximal social welfare attainable for this resource allocation problem). However, this deal would not be admissible in the ITCP during round 3.

A potential way around this problem would be to widen the class of \mathcal{T} -deals, and to also allow for deals that involve a term from \mathcal{T} being removed from a *single sender* (while our actual definition of \mathcal{T} -deal foresees a single receiver). With that extension, the ITCP could be proved to converge to an allocation with maximal social welfare also for tree-structured utilities with arbitrary coefficients, although it would also lose much of its appeal by increasing the range of available deals at each stage significantly.

Note that in the work of Rothkopf *et al.* [17] this problem does not occur, because the standard bidding language for combinatorial auctions (the so-called “OR language”) can only express superadditive preferences in the first place [14].

Communication Complexity

The *communication complexity* of a negotiation scheme assesses the length of negotiation processes and the amount of information that agents exchange. The “first level” of communication complexity is concerned with the *number of deals* required to reach an optimal allocation [8]. Here we can distinguish two types of upper bounds, namely the length of the longest and that of the shortest path to the optimum. For the ITCP we obtain the following bounds:

- The ITCP requires agents to implement a maximum of $|\mathcal{T}| \cdot (|\mathcal{A}| - 1)$ deals (*longest path*), after which the optimal allocation will be reached.
- In the ITCP, agents can always reach an optimal allocation by negotiating at most $|\mathcal{T}|$ deals (*shortest path*).

The shortest path applies if agents always implement the “ideal” deal for every term in \mathcal{T} . The longest path applies if utility functions are such that we can move each term to every agent before it finally reaches its eventual owner.

Discussion

A possible interpretation of the ITCP would be that it works with a sequence of progressively more fine-grained *approximations* to the real utility functions in such a way that structurally simple deals become individually rational (wrt. these approximations). Of course, these are approximations wrt. the *representation* of utility functions (in k -additive form), but not necessarily wrt. the values that utility functions assign to bundles of resources.

These two dimensions may, however, coincide in many practical cases. If we assume that the synergetic effect of a bundle of resources reduces as the cardinality of that bundle increases, then the sequence of auxiliary utility functions used in the ITCP will indeed be progressively more accurate in terms of the values they assign to bundles as negotiation develops. In other words, if the coefficients in the k -additive form are high for single resources, and if they get smaller as the size of the respective term increases, then the utility functions used in the ITCP become more accurate in each round. This is a very reasonable assumption to make: the larger a bundle of resources is, the more difficult would it be for an agent to estimate the additional benefit incurred by owning all the resources in that bundle *together* (beyond the benefit incurred by the relevant subsets). Indeed, the very same (cognitive) argument is one of the main reasons why k -additive utilities with low values of k are not only computationally attractive but also highly relevant in practice.

Nevertheless, the ITCP *does* violate the postulate of full individual rationality for all stages of a negotiation process.

Also, because of its “intrusive” aspect, this protocol is not completely satisfactory: it cannot be applied when agents are not under the control of the designer or when they are not willing to temporarily manipulate their own preferences in order to implement the protocol correctly. Clearly, such a protocol would be of little use when agents are humans, or simply when privacy issues prevent the designer from directly accessing the agents’ preferences.

4.2 A Non-Intrusive Protocol

To overcome the drawbacks of our intrusive protocol, we design a third protocol in which utilities are not modified, but where a special payment function influences the rationality considerations of each agent. In addition, the system itself also participates in the side payments, by giving money to or taking money from the negotiating agents. For this purpose, we introduce a so-called *system agent*. The trick here is that these side payments have exactly the same effect as truncating the utility functions, but without being intrusive (as shown in Theorem 2). Thus, this protocol could be used in real applications, even with human agents as long as utilities have been elicited.

This new protocol is called the *Omniscient ϵ -Altruistic Tree-Climbing Protocol* (OATCP). In this protocol the system actively mediates the negotiation process. The protocol is called *omniscient*, because the system needs to know the utility functions of all agents. It is called *ϵ -altruistic*, because the system must be prepared to share its money with the agents in order to reach an allocation with maximal social welfare, and because the amount of money it shares depends on a parameter ϵ . For the specification of the OATCP, we define $\Delta_{(A,A')}^{\leq l} = sw^{\leq l}(A') - sw^{\leq l}(A)$. This is the increase in social welfare for the deal $\delta = (A, A')$ wrt. truncated utilities considering only terms of size $\leq l$.

Algorithm 3 Omniscient-Altruistic Tree-Cl. Pr. (OATCP)

Require: n agents with tree-structured utilities u_1, \dots, u_n and a parameter $\epsilon \in (0, 1]$ (with \mathcal{T} the set of all terms)

- 1: All agents transmit their utility functions to the *system agent*.
- 2: $l \leftarrow 1$.
- 3: **repeat**
- 4: Restrict allowed deals to \mathcal{T}^l -deals.
 The side payment for agent i for a \mathcal{T}^l -deal $\delta = (A, A')$ involving n' agents is computed as follows:

$$p_i = u_i(A') - u_i(A) - \frac{\epsilon}{n'} \cdot \Delta_{(A',A)}^{\leq l}.$$
 For the system agent, $p_{sys} = -\sum p_i$.
- 5: Let agents negotiate individually rational deals.
- 6: **if** no more deal can be conducted **then**
- 7: $l \leftarrow l + 1$
- 8: **end if**
- 9: **until** $l > |\mathcal{R}|$

Role of the System Agent

Note that in the OATCP the *system agent* could be viewed as a kind of bank giving or receiving money from agents. The amount of money shared with other agents depends on the difference of satisfaction enjoyed by the whole society before and after the deal and on the value of the parameter ϵ .

- When ϵ is close to 1, the system agent is not guaranteed to earn any money in the long run. In that case,

the system agent is said to be *altruistic*, because the other agents will benefit from this by earning a lot.

- When ϵ is close to 0, then during each deal, the system agent takes from the other agents as much money as possible. Thus, the negotiating agents will earn the minimum amount of money necessary to reach the optimal allocation, and the system agent will earn a lot more (not altruistic at all in that case).

The protocol could be extended in such a way that the money earned by the system agent would later be redistributed. One option would be to simply divide the money equally amongst all agents, but how to best set up this redistribution in view of both fairness considerations and strategic issues is a question that warrants further investigation.

Convergence

The following results show that this protocol also converges to an allocation with maximal social welfare, just as the ITCP does. We first establish that a deal is “rational” with the special payment function (*i.e.* utility gains outweigh payments for each of the agents except the system agent) iff it increases social welfare wrt. truncated utilities.

LEMMA 2. *During round l of the OATCP, a given deal $\delta = (A, A')$ is rational iff $sw^{\leq l}(A) < sw^{\leq l}(A')$.*

PROOF. ‘ \Rightarrow ’: Suppose the deal is rational. That is, for each agent i involved, $u_i(A') - u_i(A) - p_i > 0$. Thus,

$$sw^{\leq l}(A') - sw^{\leq l}(A) = \frac{1}{\epsilon} \cdot \sum (u_i(A') - u_i(A) - p_i) > 0$$

‘ \Leftarrow ’: Now suppose $sw^{\leq l}(A') > sw^{\leq l}(A)$. Then for each agent i involved in the deal, the rationality criterion is met:

$$u_i(A') - u_i(A) - p_i = \frac{\epsilon}{n'} \cdot \Delta_{(A',A)}^{\leq l} > 0$$

This completes our proof of Lemma 2. \square

This directly leads to the convergence result.

THEOREM 2. *For any set of tree-structured utilities with positive coefficients, following the OATCP will result in an allocation of resources with maximal social welfare.*

PROOF. We show that the deals accepted in this protocol are the same as those accepted in the ITCP protocol. Clearly, during round l , a deal $\delta = (A, A')$ is accepted in the ITCP iff $sw^{\leq l}(A) < sw^{\leq l}(A')$. Lemma 2 shows that the OATCP behaves in exactly the same way. The claim then follows from Theorem 1. \square

Given this operational equivalence between the two protocols, the communication complexity of the OATCP (*i.e.* the number of deals required to reach an optimal allocation following the protocol) is the same as for the ITCP.

Money Earned by the System Agent

A critical point of the OATCP is that it requires an external provider of money. However, the following result shows that the system agent will actually *earn* money over the entire duration of the protocol, as long as we choose a sufficiently small value for ϵ .

THEOREM 3. *If ϵ is sufficiently small, then the system agent will have earned a positive amount of money by the end of any OATCP negotiation.*

PROOF. Let us first compute how much is earned by the system agent each time a deal is conducted. Suppose at step l , a deal $\delta = (A, A')$ is conducted by n' agents. The system agent makes the following payment:

$$\begin{aligned} p_{sys} &= -\sum \left(u_i(A') - u_i(A) - \frac{\epsilon}{n'} \cdot \Delta_{(A',A)}^{\leq l} \right) \\ &= -(sw(A') - sw(A)) + \epsilon \cdot \Delta_{(A',A)}^{\leq l} \end{aligned}$$

Now let d be the number of deals conducted during the whole of the negotiation process. Furthermore, let A_{init} be the initial allocation and let A_{fin} be the final allocation. If p_{sys}^{total} denotes the total amount of money the system agent pays out during the whole negotiation process, then $-p_{sys}^{total}$ is the money earned by the system. We get:

$$\begin{aligned} -p_{sys}^{total} &= \sum_{\delta=(A,A')} \left(sw(A') - sw(A) - \epsilon \cdot \Delta_{(A',A)}^{\leq l} \right) \\ &\geq sw(A_{fin}) - sw(A_{init}) - \epsilon \cdot d \cdot \max_{A,A',l} \Delta_{(A',A)}^{\leq l} \end{aligned}$$

Clearly, we can see that $-p_{sys}^{total}$ converges to the overall social surplus as ϵ decreases:

$$\lim_{\epsilon \rightarrow 0} (-p_{sys}^{total}) = sw(A_{fin}) - sw(A_{init})$$

Hence, as long ϵ is chosen small enough, the system agent can always earn during negotiation. \square

The central idea in the OATCP is that the system agent (*i.e.* the mechanism) can give a “loan” to agents to allow them to accept otherwise disadvantageous deals, but that these agents repay this loan later on when they arrange deals that are overly beneficial for themselves. This general idea would also be applicable in other scenarios than we have considered here, where agents do not necessarily have tree-structured utility functions. As long as the final allocation has higher utilitarian social welfare than the initial allocation and as long as the system agent has sufficient funds to temporarily sustain allocations with very low social welfare, *any* sequence of deals can be made individually rational from the viewpoint of the negotiating agents and would still result in an eventual surplus for the system agent.

4.3 Experimental Analysis

We have shown that both the ITCP and the OATCP perform optimally in case all agents model their preferences using tree-structured utilities with positive coefficients. While this is a reasonably wide class of utility functions that are relevant to real application domains, it is also clear that it will not always be possible to meet these constraints. Still, the ideas implemented in our protocols seem to have a general appeal and may also be useful (albeit not to the degree of guaranteeing optimal outcomes in all cases) in domains that are not tree-structured.

To test this intuition, we have carried out a number of experiments with agents that do not have tree-structured utilities. Figure 1 shows an example. The graph relates the performance of the ITCP/OATCP and of a protocol only implementing 1-deals to the maximal social welfare attainable. In each run, we have generated the utility functions of 4 agents over 10 resources in k -additive form by randomly choosing 8 terms and coefficients each. The size of terms is drawn from $\{1..k_{max}\}$ and the size of coefficients is drawn from $\{1..100\}$ (uniform distribution). The horizontal axis

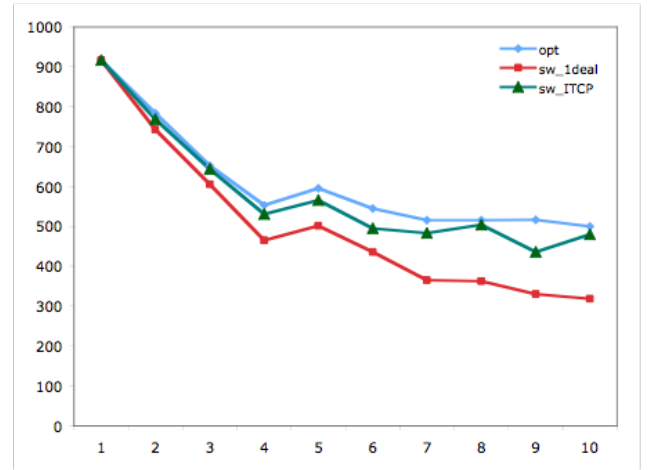


Figure 1: Performance of different protocols

shows the value of k_{max} and the vertical shows the social welfare of the final allocation (averaged over 20 runs). The experiment shows that the ITCP/OATCP clearly outperforms the 1-deal protocol for larger values of k_{max} , and that it typically manages to reach an allocation with over 95% of the maximal social welfare. We have run similar experiments with other distributions (not reported here for lack of space). The ITCP/OATCP generally performs well and is preferable to a 1-deal protocol whenever the agents’ utilities include a significant number of terms of size greater than 1.

5. CONCLUSION

We have further analysed a negotiation framework previously studied by several authors [7, 8, 9, 18]. While most work on negotiation in multiagent systems has addressed either bilateral negotiation [16] or auctions [6], this framework is multilateral, *i.e.* deals may involve any number of agents and any number of resources. The requirement for full multilateral negotiation stems from the fact that agents are assumed to be both rational and myopic. Recent results have identified situations in which deals involving only a limited number of resources each are required to reach allocations that are optimal from a social point of view [4].

However, the task of identifying rational deals, *i.e.* deals that are beneficial for all participants remains highly complex. In this paper, we have shown that if all agents model their preferences by means of *tree-structured* utility functions, then finding an allocation with maximal social welfare does become feasible by implementing adequate protocols. Two protocols, clearly designed to exploit the specific structure of these functions, have been proposed in this paper. The crucial contribution in our proposal, we believe, is the *combination of two ideas*:

- The central idea in the ITCP has been to provide a protocol that *guides* negotiation. In each round, there are only a relatively small number of options as to which deal to choose (and the complexity of these deals increases stepwise, from round to round). The particular choices made in the definition of the ITCP mean that deals are not too far from truly rational deals (certainly if we make some additional assumptions on the nature of utility functions, as discussed earlier).

- The additional idea in the OATCP has been to introduce a system agent that can *compensate* for a temporary loss in social welfare (provided that such a loss is due to a deal sanctioned by the guidelines implemented in the ITCP). It thereby allows for a wider range of deal sequences leading to the optimal allocation, including sequences of structurally simple deals.

This combination of *guidance* and *social compensation* makes our approach work. While our formal results are restricted to tree-structured domains, we believe that the central ideas are general enough to be applied to other scenarios as well. This view is supported by our experimental results with utility functions that are not tree-structured.

Our proposal may be considered a hybrid between distributed and centralised approaches to resource allocation. On the one hand, allocations emerge in a distributed manner as a result of a sequence of local negotiation steps. On the other hand, the concept of the system agent introduces a centralised element. One of the main arguments usually given in favour of distributed approaches over centralised ones (such as combinatorial auctions), is that it may not always be possible to find an agent that could act as the central authority (the auctioneer) executing the allocation procedure.¹ To some degree, the same reservations may be made against our proposal. However, we believe that there is at least one important difference. In a combinatorial auction, the auctioneer is endowed with the (significant) computational burden of deciding the final allocation (the so-called winner determination problem [6]). In our approach, on the other hand, there are no computationally hard tasks associated with the role of the system agent. Of course, in the context of the tree-structured domains studied here, the centralised task is not prohibitively hard in itself. Still, our work shows that it can also be distributed safely amongst agents by employing our protocols. Importantly, our experiments suggest that the ideas underlying the design of these protocols also have a more general appeal.

A potential problem with our approach is that agents could stop participating in negotiation before the optimal allocation has been reached. Indeed, it may be rational for them to do so, because our protocols allow certain agents to temporarily enjoy higher utility than they would enjoy in the socially optimal allocation. In this sense, the framework loses the “anytime character” of the original system [18], where every new allocation would be guaranteed to have higher social welfare than its predecessor. This anytime character is particularly important because of the prohibitively high complexity of the general framework [3, 7, 8]. If optimal allocations cannot be guaranteed to be found within a reasonable amount of time, then the original framework does at least guarantee some improvement over the initial allocation when time runs out. In our proposal, the anytime character of the framework is far less important, because it dramatically reduces the complexity of negotiation and thereby makes convergence to an optimal allocation within a reasonable amount of time a realistic option. Nevertheless, the issue of incentive compatibility in the OATCP protocol remains an interesting topic for future work.

¹Such arguments for distributed negotiation schemes are similar to those often given to motivate the interest in *distributed constraint optimisation problems* [10]. It would be interesting to explore whether algorithms developed in this area could be applied to our negotiation setting as well.

6. REFERENCES

- [1] K. J. Arrow, A. K. Sen, and K. Suzumura, editors. *Handbook of Social Choice and Welfare*. North-Holland, 2002.
- [2] U. Chajewska and D. Koller. Utilities as random variables: Density estimation and structure discovery. In *Proc. UAI-2000*. Morgan Kaufmann, 2000.
- [3] Y. Chevaleyre, U. Endriss, S. Estivie, and N. Maudet. Multiagent resource allocation with k -additive utility functions. In *Proc. DIMACS-LAMSADE Workshop on Computer Science and Decision Theory*, 2004.
- [4] Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet. Negotiating over small bundles of resources. In *Proc. AAMAS-2005*. ACM Press, 2005.
- [5] V. Conitzer, T. W. Sandholm, and P. Santi. Combinatorial auctions with k -wise dependent valuations. In *Proc. AAAI-2005*. AAAI Press, 2005.
- [6] P. Cramton, Y. Shoham, and R. Steinberg, editors. *Combinatorial Auctions*. MIT Press, 2006.
- [7] P. E. Dunne, M. Wooldridge, and M. Laurence. The complexity of contract negotiation. *Artificial Intelligence*, 164(1–2):23–46, 2005.
- [8] U. Endriss and N. Maudet. On the communication complexity of multilateral trading: Extended report. *Journal of Autonomous Agents and Multiagent Systems*, 11(1):91–107, 2005.
- [9] U. Endriss, N. Maudet, F. Sadri, and F. Toni. Negotiating socially optimal allocations of resources. *Journal of Artificial Intelligence Research*, 2006. To appear.
- [10] B. Faltings and M. Yokoo. Introduction: Special issue on distributed constraint satisfaction. *Artificial Intelligence*, 161(1–2):1–5, 2005.
- [11] M. Grabisch. k -order additive discrete fuzzy measures and their representation. *Fuzzy Sets and Systems*, 92:167–189, 1997.
- [12] P. J. ’t Hoen and J. A. La Poutré. A decommitment strategy in a competitive multi-agent transportation setting. In *AMEC V*. Springer-Verlag, 2004.
- [13] M. Lemaître, G. Verfaillie, F. Jouhaud, J.-M. Lachiver, and N. Bataille. Selecting and scheduling observations of agile satellites. *Aerospace Sciences and Technology*, 6:367–381, 2002.
- [14] N. Nisan. Bidding languages for combinatorial auctions. In P. Cramton et al., editors, *Combinatorial Auctions*. MIT Press, 2006.
- [15] V. Robu, D. Somefun, and J. A. La Poutré. Modelling complex multi-issue negotiations using utility graphs. In *Proc. AAMAS-2005*. ACM Press, 2005.
- [16] J. S. Rosenschein and G. Zlotkin. *Rules of Encounter*. MIT Press, 1994.
- [17] M. H. Rothkopf, A. Pekeč, and R. M. Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131–1147, 1998.
- [18] T. W. Sandholm. Contract types for satisficing task allocation: I Theoretical results. In *Proc. AAAI Spring Symposium: Satisficing Models*, 1998.
- [19] P. R. Wurman, W. E. Walsh, and M. P. Wellman. Flexible double auctions for electronic commerce: Theory and implementation. *Decision Support Systems*, 24(1):17–27, 1998.