Multiagent Resource Allocation with k-additive Utility Functions

Yann Chevaleyre*, Ulle Endriss†, Sylvia Estivie*, Nicolas Maudet*

Abstract

We briefly review previous work on the *welfare engineering* framework where autonomous software agents negotiate on the allocation of a number of discrete resources, and point out connections to combinatorial optimisation problems, including combinatorial auctions, that shed light on the computational complexity of the framework. We give particular consideration to scenarios where the preferences of agents are modelled in terms of k-additive utility functions, *i.e.* scenarios where synergies between different resources are restricted to bundles of at most k items.

Key words: negotiation, representation of utility functions, social welfare, combinatorial optimisation, bidding languages for combinatorial auctions

1 Introduction

Distributed systems in which autonomous software agents interact with each other, in either cooperative or competitive ways, can often be usefully interpreted as *societies of agents*; and we can employ formal tools from microeconomics to analyse such systems. If we model the interests of individual agents in terms of a notion of *individual welfare*, then the overall performance of the system provides us with a measure of *social welfare*.

Individual welfare may be measured either *quantitatively*, typically by defining a utility function mapping "states of affairs" (outcomes of an election, allocations of resources, agreements on a joint plan of action, etc.) to numeric values; or *qualitatively*, by defining

*LAMSADE, Université Paris-Dauphine, Place du Maréchal de Lattre de Tassigny, 75775 Paris cedex 16, France. Email: {chevaley,estivie,maudet}@lamsade.dauphine.fr.

†Department of Computing, Imperial College London, 180 Queen's Gate, London SW7 2AZ, UK. Email: ue@doc.ic.ac.uk.

a preference relation over alternative states. The concept of social welfare, as studied in welfare economics, is an attempt to characterise the well-being of a society in relation to the welfare enjoyed by its individual members [1, 2, 18, 23]. The best known examples (both relying on quantitative measures of individual welfare) are the *utilitarian* programme, according to which social welfare should be interpreted as the sum of individual utilities, and the *egalitarian* programme, which identifies the welfare of society with the welfare of its "poorest" member.

For instance, in an electronic commerce application where users pay a fee to the provider of the infrastructure depending on the personal benefits incurred by using the system, the increase in utilitarian social welfare correctly reflects the profit generated by the provider. The application discussed by Lemaître *et al.* [17], on the other hand, where agents representing different stake-holders repeatedly negotiate over the access to an earth observation satellite (which has been jointly funded by the stake-holders), requires a *fair* treatment of all agents. Here, the respective values of different access schedules may be better modelled by an egalitarian social welfare ordering.

We are particularly interested in applications where negotiation between autonomous agents serves as a means of addressing a resource allocation problem. Recent results in this framework concern the feasibility of reaching an allocation of resources that is optimal from a social point of view [8, 21], as well as (certain aspects of) the complexity of doing so, in terms of both computational costs and the amount of communication required [5, 6, 7].

Multiagent resource allocation is just one of several recent examples for the successful exploitation of ideas from microeconomics in the context of computer science. Other applications include automatic contracting [21], selfish routing in shared networks [11], distributed reinforcement learning [24], and data mining [16]. This area of activity, which we may term *computational microeconomics*, brings together theoretical computer science and microeconomics in new and fruitful ways, benefiting not only these disciplines themselves but also "hot" research topics such as multiagent systems and electronic commerce.

In previous work, we have put forward the framework of *welfare engineering* [8], which addresses the design of suitable rationality criteria for autonomous software agents participating in negotiations over resources in view of different notions of social welfare, as well as the development of such notions of social welfare themselves. In Section 2, we briefly review the underlying multiagent resource allocation system and recall two previous results on the feasibility of reaching a socially optimal allocation of resources from a utilitarian point of view. As we shall see, in cases where the utility functions used by agents to model their preferences over alternative bundles of resources are *additive*, it is sufficient to use very simple negotiation protocols that only cater for deals involving a single resource at a time.

This result suggests to investigate generalisations of the notion of additivity, and hence we consider the case of k-*additive* functions, as studied, for instance, in the context of fuzzy measure theory [14]. The notion of k-additivity suggests an alternative representation of utility functions, which we introduce in Section 3. We show that this representation is as expressive as the "standard" representation (which involves listing the utility values for all possible bundles) and that it often allows for a more succinct representation of utility functions. Nevertheless, it turns out that the positive result on the complexity of deals obtained for additive functions *cannot* be generalised in the expected manner. Counterexamples are given in Section 4.

In Section 5, we discuss connections between our multiagent resource allocation framework and some well-known *combinatorial optimisation* problems (namely, weighted set packing and the independent set problem). These can be used to prove *NP-hardness* results for the decision problem associated with the task of finding a socially optimal resource allocation. We prove complexity results with respect to both the standard representation of utility functions and the representation based on k-additivity. In this context, we also discuss connections of our optimisation problem to the winner determination problem in *combinatorial auctions*. We are going to point out connections between different ways of representing utility functions and different *bidding languages* for such auctions along the way. Our conclusions are presented in Section 6.

2 Resource Allocation by Negotiation

An instance of our negotiation framework consists of a finite set of (at least two) *agents* $\mathcal{A}$ and a finite set of non-divisible *resources* $\mathcal{R}$. A resource *allocation* A is a partitioning of the set $\mathcal{R}$ amongst the agents in $\mathcal{A}$. For instance, given an allocation A with $A(i) = \{r_3, r_7\}$, agent i would own resources r_3 and r_7. Given a particular allocation of resources, agents may agree on a (multilateral) *deal* to exchange some of the resources they currently hold. In general, a single deal may involve any number of resources and any number of agents. It transforms an allocation of resources A into a new allocation A'; that is, we can define a deal as a pair $\delta = (A, A')$ of allocations (with $A \neq A'$).

Each agent $i \in \mathcal{A}$ is equipped with a *utility function* u_i mapping bundles of resources (subsets of $\mathcal{R}$) to rational numbers. We abbreviate $u_i(A) = u_i(A(i))$ for the utility value assigned by agent i to the set of resources it holds for allocation A. While individual agents may have their own interests, as a system designer, we are interested in the *social welfare* associated with a given allocation. According to the aforementioned *utilitarian* programme, the social welfare of an allocation A is given by the sum of utilities exhibited by all the agents in the system:

$$sw(A) \quad = \quad \sum_{i \in \mathcal{A}} u_i(A)$$

That is, any deal that results in a higher sum of utilities (or equivalently, in higher average utility) would be considered socially beneficial. One of the main questions we are interested in in the welfare engineering framework is under what circumstances negotiation between agents will result in an improvement, and eventually an optimisation, with respect to such a notion of social welfare.

A deal may be coupled with a number of monetary side payments to compensate some of the agents involved for an otherwise disadvantageous deal. We call a deal *rational* iff it results in a gain in utility (or money) that strictly outweighs a possible loss in money (or utility) for each of the agent involved in that deal.

As shown in previous work [9], a deal is rational iff it results in an increase in utilitarian social welfare. Given this connection between the "local" notion of rationality and the "global" notion of social welfare, we can prove the following result on the sufficiency of rational deals to negotiate socially optimal allocations [9, 21]:

Theorem 1 (Maximal social welfare) *Any sequence of rational deals with side payments will eventually result in an allocation of resources with maximal utilitarian social welfare.*

This means that (i) there can be no infinite sequence of deals all of which are rational, and (ii) once no more rational deals are possible the agent society must have reached an allocation that has maximal social welfare. The crucial aspect of this result is that *any* sequence of deals satisfying the rationality condition will cause the system to converge to an optimal allocation. That is, whatever deals are agreed on in the early stages of the negotiation, the system will never get stuck in a local optimum and finding an optimal allocation remains an option throughout.

A drawback of the general framework is that the above result only holds if deals involving any number of resources and agents are admissible [9, 21]. In some cases this problem can be alleviated by putting suitable restrictions on the utility functions agents may use to model their preferences. Interesting special classes of utility functions to consider include, for instance, *non-negative* functions (where an agent may not assign a negative utility to any bundle) and *monotonic* functions (where the utility of a set of resources cannot be lower than the utility assigned to any of its subsets).

A particularly simple class is the class of additive functions. A utility function is called *additive* iff the value ascribed to a set of resources is always the sum of the values of its members. As has been shown in an earlier paper [9], in scenarios where utility functions may be assumed to be additive, it is possible to guarantee optimal outcomes even when agents only negotiate deals involving a single resource and a pair of agents at a time (so-called *one-resource-at-a-time deals*):

Theorem 2 (Additive scenarios) *If all utility functions are additive, then any sequence of rational one-resource-at-a-time deals with side payments will eventually result in an allocation of resources with maximal utilitarian social welfare.*

This result is of great practical relevance, because it shows that it is sufficient to design negotiation protocols for pairs of agents (rather than larger groups) and single resources (rather than sets) for applications in which preferences can be modelled in terms of additive utility functions. In the next section, we are going to introduce a generalisation of this notion of additivity.

3 Representations of Utility Functions

An agent's utility function may be represented in different ways. This situation is similar, for instance, to the case of combinatorial auctions, where one can use different *bidding languages* to express the preferences of the participating agents [19, 22]. Maybe the most intuitive representation of a utility function is the *bundle form*, which amounts to listing all bundles of resources to which the agent assigns a non-zero value. Clearly, this approach can soon become problematic, as there may be up to 2^n such bundles in the worst case.

An alternative representation is based on the notion of k-additive functions, which have been studied in the context of fuzzy measure theory [14]. Given a natural number k, a utility function is called *k-additive* iff the utility assigned to a bundle of resources R can be represented as the sum of basic utilities ascribed to subsets of R with cardinality $\leq k$. More formally, a k-additive utility function can be written as follows:

$$u_i(R) \;=\; \sum_{T \subseteq \mathcal{R},\, |T| \leq k} \alpha_i^T \times I_R(T) \quad \text{with } I_R(T) = \begin{cases} 1 & \text{if } T \subseteq R \\ 0 & \text{otherwise} \end{cases}$$

That is, the utility function of agent i is characterised by the coefficients α_i^T for bundles of resources $T \subseteq \mathcal{R}$ with at most k elements. Agent i enjoys an increase in utility of α_i^T when it owns all the items in T *together*, *i.e.* α_T^i represents the synergetic value of this bundle. An example for a 2-additive utility function would be $u_i(R) = 3 \times I_R(\{r_1\}) - 2 \times I_R(\{r_2, r_3\})$. For the sake of simplicity, we are going to omit the indicator function I_R as well as the explicit mentioning of the bundle variable R when defining concrete k-additive utility functions. Using this simplified notation, the above function becomes $u_i = 3.r_1 - 2.r_2.r_3$.

While the bundle form corresponds to the so-called XOR-language for expressing bids in combinatorial auctions [19, 22], there appears to be no counterpart to the k-additive form in the literature on bidding languages. The connections between our framework and combinatorial auctions will be explored further in Section 5.

Utility functions that are k-additive with $k = 1$ are like the additive functions discussed in the previous section (except that they also allow for a non-zero utility value to be assigned to the empty set). Hence, the notion of k-additivity is a generalisation of the familiar notion of additivity. In fact, as we are going to show next, k-additive utility functions cover a whole range of utility function, from the very simple additive functions to the most general utility functions without any restrictions.

Proposition 1 (Expressive power of k-additive utility functions) *Any utility function can be represented as a k-additive function with $k = |\mathcal{R}|$.*

Proof. Let u_i be any utility function mapping subsets of $\mathcal{R}$ to rational numbers. We recursively define coefficients α_i^T for $T \subseteq \mathcal{R}$ as follows:

$$\begin{array}{rcl} \alpha_i^{\{\,\}} & = & u_i(\{\,\}) \\ \alpha_i^R & = & u_i(R) - \sum\limits_{T \subset R} \alpha_i^T \quad \text{for all } R \subseteq \mathcal{R} \text{ with } R \neq \{\,\} \end{array}$$

Hence, $u_i(R) = \sum_{T \subseteq R} \alpha_i^T = \sum_{T \subseteq \mathcal{R}} \alpha_i^T \times I_R(T)$. This is a k-additive utility function for $k = |\mathcal{R}|$. $\square$

Clearly, the bundle form is also fully expressive, *i.e.* our two ways of representing utility functions are equivalent in the sense that they can both express any utility function over the set of resources $\mathcal{R}$. Besides expressive power, another important consideration concerns the succinctness of a representation. It turns out that neither of the two representations is more succinct in all cases. In fact, as we are going to see next, there are cases where translating a utility function given in k-additive form into the bundle form results in an exponential blow-up of the representation, and vice versa.[1]

Proposition 2 (Efficiency of the k-additive form) *The bundle form cannot polynomially simulate the k-additive form of representing utility functions.*

Proof. We prove the claim by giving an example for a utility function with a representation that is linear in the size of $\mathcal{R}$ for the k-additive form, but exponential for the bundle form. Consider a utility function that maps a bundle of resources to the number of elements in that bundle. This is a 1-additive function, which requires the specification of exactly $|\mathcal{R}|$ coefficients in the k-additive form (namely $\alpha_1^T = 1$ for all T with $|T| = 1$). For the bundle form, however, the specification of a utility value for each of the $2^{|\mathcal{R}|} - 1$ non-empty bundles is required. $\square$

[1] Nisan [19] proves a number of similar separation results for different types of bidding languages for combinatorial auctions.

Proposition 3 (Efficiency of the bundle form) *The k-additive form cannot polynomially simulate the bundle form of representing utility functions.*

Proof. We give an example for a utility function with a representation that is linear in the size of $\mathcal{R}$ for the bundle form, but exponential for the k-additive form. Consider a utility function u_i that assigns 1 to any bundle consisting of a single resource and 0 to any other bundle. In the bundle form, u_i requires the specification of a utility value for exactly $|\mathcal{R}|$ bundles (namely those with just a single element). In the k-additive form, on the other hand, it requires the specification of $2^{|\mathcal{R}|} - 1$ coefficients: We certainly have $\alpha_i^T = 1$ for any bundle T with $|T| = 1$. To ensure that $u_i(R) = 0$ for any R with $|R| = 2$ we require $\alpha_i^T = -2$ for any T with two elements. For a bundle with three elements, the sum of the coefficients for all its subsets is $3 \times 1 + 3 \times (-2) = -3$, *i.e.* we have to set $\alpha_i^T = 3$ whenever $|T| = 3$, and so on. In general, we have to choose $\alpha_i^T = |T| \times (-1)^{|T|+1}$ (which is different from 0 for any of the $2^{|\mathcal{R}|} - 1$ subsets T of $\mathcal{R}$ with $T \neq \{\,\}$) to be able to represent u_i as a k-additive function. □

The examples given in the proofs of Propositions 2 and 3 are extreme cases, where one form of representation is exponentially more succinct than the other. While the difference is not always going to be this strong, choosing the right representation for a given problem domain is still important. Broadly speaking, the k-additive form will typically be more succinct in cases where there are only limited synergies between different items. This is likely to be the case for many application domains, which makes this a useful language for expressing utilities in practice.

4 Complexity of Deals with k-additive Utilities

Recall that Theorem 2 has shown that it is always possible to negotiate a socially optimal allocation of resources by means of rational deals involving only a single resource at a time whenever the utilities of all the agents involved are additive (*i.e.* 1-additive). Intuitively, we could have expected a similar result for k-additive utilities with $k \geq 2$ (*i.e.* a result that states that rational deals involving at most k resources at a time are sufficient to reach optimal allocations whenever all utility functions are k-additive). However, as we are going to show next, this turns out not to be the case. The deals required to reach allocations with maximal social welfare in the k-additive case are much more complex.

Proposition 4 (Necessity of complete deals) *Even if all utility functions are k-additive for some $k \geq 2$, a deal involving the complete set of resources may be necessary to reach an allocation with maximal utilitarian social welfare by means of a sequence of rational deals with side payments.*

Proof. To prove the claim, we construct an example with 2-additive utility functions in which a deal involving all resources in $\mathcal{R}$ is needed. Consider two agents sharing n resources $\mathcal{R} = \{r_1, r_2, \ldots, r_n\}$, with the following 2-additive utility functions:

$$\begin{aligned} u_1 &= 0 \\ u_2 &= r_1 - r_1.r_2 - r_1.r_3 - r_1.r_4 - \ldots - r_1.r_n \end{aligned}$$

Let A_{init} be the initial allocation of resources describing which agent owns which resource before negotiation commences, and let A_{opt} be the allocation maximising utilitarian social welfare:

	A_{init}	A_{opt}
Agent 1	$\{r_1\}$	$\{r_2, r_3, \ldots, r_n\}$
Agent 2	$\{r_2, r_3, \ldots, r_n\}$	$\{r_1\}$

Here, $sw(A_{init}) = 0$ and $sw(A_{opt}) = 1$. In fact, the *only* allocation which has a social welfare greater than $sw(A_{init})$ is A_{opt}. Recall that a deal increases social welfare iff it is rational with side payments (the proof may be found in [9]). Thus, the only rational deal here is $\delta = (A_{init}, A_{opt})$, which is a bilateral deal involving all n resources at the same time. □

A possible objection to the example used in our proof may be that it is rather artificial. Utility functions that also have some additional properties, such as being monotonic, besides being k-additive may be more relevant in practice. To show that the problem of requiring complex deals persists even when we make such additional assumptions, we give a further, similarly simple, example that demonstrates that also for k-additive functions that are monotonic, rational deals involving no more than k resources do not always suffice to negotiate socially optimal allocations. Consider the case of three agents and four resources with the following utility functions:

$$\begin{aligned} u_1 &= 4.r_1.r_3 \\ u_2 &= 3.r_1.r_2 \\ u_3 &= 2.r_3.r_4 \end{aligned}$$

Let A_{init} be the initial allocation and let A_{opt} be the optimal allocation with maximal utilitarian social welfare:

	A_{init}	A_{opt}
Agent 1	$\{r_1, r_3\}$	$\{\,\}$
Agent 2	$\{r_2, r_4\}$	$\{r_1, r_2\}$
Agent 3	$\{\,\}$	$\{r_3, r_4\}$

We have $sw(A_{init}) = 4$ and $sw(A_{opt}) = 5$. Clearly, the only rational deal with side payments (*i.e.* the only deal increasing social welfare) is $\delta = (A_{init}, A_{opt})$, which is a deal involving 3 (rather than just 2) resources at the same time.

In summary, our results show, differently from what one might have expected, that the restriction to utility functions that are k-additive for a given value of k does not, in general, reduce the complexity of deals required to reach a socially optimal allocation of resources in an agent society whose members follow a simple rational negotiation strategy.

5 Connections to Combinatorial Optimisation

In Section 3, we have already mentioned the connection between different representations of utility functions (in our case the bundle form and the k-additive form) in our negotiation framework and different bidding languages in combinatorial auctions. In what follows, we explore a further connection between the two areas.

If we view the problem of finding an allocation with maximal social welfare as an algorithmic problem faced by a central authority (rather than as a problem of designing suitable negotiation mechanisms), then we can observe an immediate relation to the so-called *winner determination problem* in combinatorial auctions [19, 20, 22]. In a combinatorial auction, bidders can put in bids for different *bundles* of items (rather than just single items). After all bids have been received, the auctioneer has to find an allocation for the items on auction amongst the bidders in a way that maximises his revenue. If we interpret the price offered for a particular bundle of items as the utility the agent in question assigns to that set, then maximising revenue (*i.e.* the sum of prices associated with winning bids) is equivalent to finding an allocation with maximal utilitarian social welfare. This equivalence holds, at least, in cases where the optimal allocation of items in an auction is such that *all* of the items on auction are in fact being sold (so-called *free disposal*).

Winner determination in combinatorial auctions is known to be NP-complete [20].[2] The quoted result applies to the case of the "standard" bidding language, which allows bidders to specify prices for particular bundles and makes the implicit assumption that they are prepared to obtain any number of disjoint bundles for which they have submitted a bid (Nisan [19] calls this the "OR language"). Our languages for expressing utilities are more general than this. Hence, the correspondence to combinatorial auctions suggests that the problem of finding an allocation with maximal utilitarian social welfare is at least NP-hard. We can make this observation more precise by showing how our problem relates to well-known NP-complete "reference problems" [3, 13, 15]. One such problem is MAXIMUM WEIGHTED SET PACKING. We use the schema of Ausiello *et al.* [3] to

[2]More precisely, the *decision problem underlying the winner determination problem, i.e.* the problem of checking whether it is possible to find an allocation that achieves at least a given minimal revenue K is NP-complete. The concept of NP-completeness applies to decision problems rather than optimisation problems [3]. The winner determination problem is still NP-hard in the sense that solving it is at least as hard as solving any NP-complete decision problem.

define combinatorial optimisation problems:

MAXIMUM WEIGHTED SET PACKING
Instance: Collection $\mathcal{C}$ of finite sets, each associated with a positive weight.
Solution: Collection of disjoint sets $\mathcal{C}' \subseteq \mathcal{C}$.
Measure: Sum of the weights associated with the sets in $\mathcal{C}'$.

The *optimisation problem* known as MAXIMUM WEIGHTED SET PACKING is the problem of finding a solution $\mathcal{C}'$ for which the sum of the weights associated with the sets in $\mathcal{C}'$ is maximal. The underlying *decision problem* is the problem of answering the question whether there exists a solution $\mathcal{C}'$ for which the sum of weights exceeds a given threshold K. This decision problem is known to be NP-complete (in the size of the instance, *i.e.* with respect to the number of sets in $\mathcal{C}$) [3].

Intuitively, we are going to interpret the sets in $\mathcal{C}$ as bundles of resources and the weights associated with them as utility values. To make the correspondence complete, however, we require the following generalisation of MAXIMUM WEIGHTED SET PACKING:

MAXIMUM COLOURED WEIGHTED SET PACKING WITH FULL COVERAGE
Instance: Collection $\mathcal{C}$ of coloured finite sets, each associated with a weight.
Solution: Collection of disjoint sets $\mathcal{C}' \subseteq \mathcal{C}$, including exactly one set of each colour, such that $\{x \in S \mid S \in \mathcal{C}'\} = \{x \in S \mid S \in \mathcal{C}\}$.
Measure: Sum of the weights associated with the sets in $\mathcal{C}'$.

There are three differences between the original weighted set packing problem and our extended problem: (i) we have dropped the restriction to *positive* weights; (ii) every set is associated with a *colour* and every colour is required to be represented exactly once in any valid solution; and (iii) all the items occurring in any of the set in $\mathcal{C}$ need to be *covered* by the set packing $\mathcal{C}'$.

Lemma 1 (Complexity of extended WSP) *The decision problem underlying* MAXIMUM COLOURED WEIGHTED SET PACKING WITH FULL COVERAGE *is NP-complete.*

Proof. NP-membership of our problem follows from the fact that all the conditions imposed on valid solutions can be checked in polynomial time.[3] NP-hardness follows from the known NP-hardness result for the decision problem underlying MAXIMUM WEIGHTED SET PACKING. To see that our extended problem is indeed at least as hard

[3]Recall that a decision problem is in NP iff any proposed proof for a positive answer can be checked (although not necessarily found) in polynomial time.

as the original problem, we need to show how the original problem can be reduced to the extended one. Consider the following mapping: Given an instance $\mathcal{C}$ of MAXIMUM WEIGHTED SET PACKING, first add the set $\{x\}$ (with weight 0) for every $x \in S$ for every $S \in \mathcal{C}$ to the collection (unless that set is already present). Then assign a different colour to each set in the extended collection. Finally, also introduce an empty set (with weight 0) for each of the colours. The additional sets ensure that for any solution of the original problem there is a solution of the extended problem such that all elements as well as colours are covered. □

The following theorem has first been proved by Dunne *et al.* [6] by means of a non-trivial reduction from a variant of 3-SAT where the number of clauses in the input formula is equal to the number of propositional variables occurring in that formula.[4] Having established the complexity of our extended set packing problem, we are in a position to give a much simpler proof.

Theorem 3 (Complexity wrt. bundle form) *The decision problem underlying the problem of finding an allocation with maximal utilitarian social welfare with utilities represented in bundle form is NP-complete.*

Proof. The problem of finding an allocation with maximal utilitarian social welfare is equivalent to MAXIMUM COLOURED WEIGHTED SET PACKING WITH FULL COVERAGE: sets in the collection correspond to bundles, colours correspond to agents, and the weight associated with a coloured set corresponds to the utility assigned to the respective bundle by the respective agent. NP-completeness then follows from Lemma 1. □

Note that we could have proved the same result using a direct reduction from MAXIMUM WEIGHTED SET PACKING, even from the version without weights, but having a combinatorial optimisation problem that is exactly equivalent to our problem of finding a socially optimal allocation of resources in the language familiar from the literature on combinatorial optimisation is interesting in its own right.

Our next aim is to establish the complexity of the same decision problem, but this time with respect to the k-additive form rather than the bundle form of representing utility functions. As the k-additive form may be exponentially more succinct than the bundle form, NP-hardness with respect to the later does not necessarily imply NP-hardness with respect to the former. Nevertheless, as we are going to see, deciding whether there exists an allocation of resources with a utilitarian social welfare that exceeds a given threshold

[4] Fargier *et al.* [10] also prove a very similar result. In their resource allocation framework agents can, by default, share individual resources, but if a particular resource can only be owned by one agent at a time this can be specified by giving additional constraints.

is also NP-complete. This time, we are going use a reduction from another well-known combinatorial optimisation problem:

MAXIMUM INDEPENDENT SET
Instance: Graph $G = (V, E)$.
Solution: Set $V' \subseteq V$ s.t. no two vertices in V' are joined by and edge in E.
Measure: Cardinality $|V'|$.

The problem of finding an independent set whose cardinality exceeds a given threshold is known to be NP-complete [13] (although some special cases, e,g. when all vertices have a degree of at most 2, are solvable in polynomial time).

Theorem 4 (Complexity wrt. k-additive form) *The decision problem underlying the problem of finding an allocation with maximal utilitarian social welfare with utilities represented in k-additive form is NP-complete.*

Proof. Firstly, the problem is certainly in NP, because checking whether the social welfare of a given allocation exceeds a given threshold K can be checked in polynomial time. We show NP-hardness by reducing the decision problem underlying MAXIMUM INDEPENDENT SET to our problem. Given a graph $G = (V, E)$ and a rational number K, we want to establish whether the graph has got an independent set V' with cardinality $|V'| > K$. Without loss of generality, we may assume that no vertex in V is joined with itself by and edge in E, because no solution V' would contain such a vertex. We can map this independent set problem to an instance of our decision problem by introducing an agent for every vertex in V and a resource for every edge in E. We define the utility coefficients in the k-additive form for every agent i as follows: Let T be the set of resources corresponding to edges in E that are adjacent to the vertex corresponding to i. We define $\alpha_i^T = 1$ and there are no other utility coefficients for agent i. Now every allocation A corresponds to an independent set V' and the utilitarian social welfare of A equals the cardinality of V'. Hence, there exists an independent set V' with $|V'| > K$ iff there exists an allocation A with $sw(A) > K$. □

Of course, as with MAXIMUM INDEPENDENT SET, there will be special cases where the above problem is not NP-hard anymore. A very simple example would be the case of $k = 1$: It is easy to devise a polynomial algorithm for finding an allocation with maximal utilitarian social welfare in cases where all agents use 1-additive utility functions (simply assign each resource to the agent that values it the highest).

What about $k = 2$ though? In our proof, k directly corresponds to the maximal degree of vertices in the graph used for the reduction. As pointed out already, the decision problem underlying MAXIMUM INDEPENDENT SET is *not* NP-hard anymore if no vertex has got a degree exceeding 2. Hence, our proof of Theorem 4 would not allow us to conclude that our problem remains NP-hard for $k = 2$. This is the objective of our next theorem. It shows that the problem of finding a socially optimal allocation is still NP-hard for $k = 2$. In this sense, our problem is harder than MAXIMUM INDEPENDENT SET (where the transition to NP-hardness only occurs when we move from 2 to 3).

Theorem 5 (Complexity for $k = 2$) *The decision problem underlying the problem of finding an allocation with maximal utilitarian social welfare with utilities represented in k-additive form remains NP-complete for $k = 2$.*

Proof. NP-membership follows from Theorem 4. To prove NP-hardness for $k = 2$, we show how any problem instance with k-additive utility functions for $k \geq 3$ can be transformed into a problem with 2-additive functions in polynomial time. NP-hardness then follows, again, from Theorem 4.

We will show that a 3-additive resource allocation problem can be reformulated as a 2-additive one. This is an adaptation of an idea by Boros and Hammer [4] to our case. Consider n agents having 3-additive utility functions. We will show here that each 3-additive term appearing in the utility functions can be replaced by a set of five 2-additive ones, in a way that leaves the optimal resource allocation unchanged. Let us suppose u_i contains a 3-additive term $\alpha.r_1.r_2.r_3$ which we want to get rid of. To make it 2-additive, we will have to create a new "pseudo-resource" r_{12} which represents the bundle $\{r_1, r_2\}$. Clearly, the integrity constraint $r_{12} = r_1.r_2$ (with both r_{12} and $r_1.r_2$ being equal to either 0 or 1) has to be fulfilled in order to have $\alpha.r_1.r_2.r_3 = \alpha.r_{12}.r_3$.

For this purpose, let us define the following function with M being a big constant ($M = 1 + 2\sum_{i,T} |\alpha_i^T|$ is sufficient):

$$integrity(r_1, r_2, r_{12}) \quad = \quad -M.r_1.r_2 + 2M.r_1.r_{12} + 2M.r_2.r_{12} - 3M.r_{12}$$

This integrity function, which is 2-additive, will be added to the term $\alpha.r_{12}.r_3$ to penalise it in case the constraint is violated:

$$\begin{array}{ll} \textit{integrity}(r_1, r_2, r_{12}) = 0 & \textit{if } r_{12} = r_1.r_2 \\ \textit{integrity}(r_1, r_2, r_{12}) \leq -M & \textit{otherwise} \end{array}$$

Let us now consider the new utility function equal to u_i in which the term $\alpha.r_1.r_2.r_3$ has been replaced by the 2-additive formula $\alpha.r_{12}.r_3 + \textit{integrity}(r_1, r_2, r_{12})$. This change does not affect social welfare in case the integrity constraint is fulfilled. If not, then the social welfare will have a very low value (far from optimal). Up to now, a single 3-additive term was reduced to five 2-additive terms. By iterating this reduction, a set of 3-additive utilities can be reformulated in 2-additive utilities, without changing the optimal allocation. In addition, note that this can be applied $k - 2$ times to transform any k-additive utility function into one that is 2-additive.

It follows that finding a socially optimal resource allocation with 2-additive utility functions is as hard as finding it for k-additive functions with $k > 2$ (modula the polynomial reduction described). Hence, the problem remains NP-hard for $k = 2$. □

As a final complexity result, we are going to show that the problem of *verifying* that a given allocation is socially optimal is co-NP-complete. This holds for both the bundle form and the k-additive form of representing utility functions and is a simple corollary to Theorems 3 and 4.

Corollary 1 (Complexity of verifying optimality) *The problem of verifying that a given allocation has got maximal utilitarian social welfare is co-NP-complete (for both representations of utility functions).*

Proof. Checking that an allocation A is *not* optimal involves firstly computing $sw(A)$, which can be done in polynomial time, and then solving the decision problem "is there an allocation A' with $sw(A') > sw(A)$?". The latter is NP-complete according to Theorem 3 (Theorem 4) for the bundle (k-additive) form. Hence, the complementary problem must be co-NP-complete. □

Related to this result, Dunne *et al.* [6] have shown that the problem of checking whether a given allocation of resources is Pareto optimal is also co-NP-complete.[5]

What is the practical relevance of the connections between our negotiation framework and the combinatorial optimisation problems discussed in this section? In the proof of Theorem 4, for instance, we have reduced MAXIMUM INDEPENDENT SET to a very specific class of instances of the problem of finding a socially optimal allocation of resources, namely those where the utility functions of all agents can be represented as k-additive functions with only a single non-zero coefficient. While this reduction has been useful to establish our NP-hardness result, it does not provide us with much useful information on how to find an optimal allocation in practice. Here, the opposite direction, *i.e.* reductions *from* resource allocation problems *to* standard combinatorial optimisation problems may be more attractive. Such a reduction would allow us to exploit existing algorithms, including highly optimised approximation algorithms [3], to find optimal (or near-optimal) allocations of resources.

[5]An allocation of resources is called *Pareto optimal* iff there is no other allocation that would be better for at least one of the agents without being worse for any of the others. For further results on negotiating Pareto optimal allocations we refer to [9].

In the case of utility functions in k-additive form, the resource allocation problem can be reduced to the weighted variant of MAXIMUM INDEPENDENT SET [3], provided all utility coefficients are positive and all agents value the empty bundle at 0. The mapping firstly involves introducing a vertex for each coefficient (and using the coefficient itself as the weight associated with that vertex). Then we introduce an edge for every possible "conflict": any two vertices α_i^T and $\alpha_j^{T'}$ with $i \neq j$ and $T \cap T' \neq \{\,\}$ are joined together by an edge. The independent set yielding the highest overall weight then corresponds to the optimal allocation.

In the case of the bundle form, we already have established a on-to-one correspondence to MAXIMUM COLOURED WEIGHTED SET PACKING WITH FULL COVERAGE. However, to exploit existing algorithms, we require a reduction to the standard problem of MAXIMUM WEIGHTED SET PACKING. This is possible whenever a resource allocation problem meets the following conditions: (i) all utility functions are non-negative; (ii) all agents value the empty bundle at 0; and (iii) we can assume *free disposal*, *i.e.* for every incomplete allocation (not covering all resources) there is always a complete one that is not worse.[6] The proposed mapping would involve creating a set for every pair of an agent i and a bundle R with $u_i(R) \neq 0$. Here, we consider both the resources and the agent as elements of that set. The weight associated with the set would be $u_i(R)$. It is then not difficult to see that allocations with maximal social welfare correspond to set packings with maximal overall weight. Hence, we can reuse existing algorithms for MAXIMUM WEIGHTED SET PACKING to find optimal allocations of resources.

Finally, we should stress that this would be a methodology for a *centralised* approach to finding optimal resource allocations. It is not immediately applicable to negotiation, which is a *distributed* process. Nevertheless, the techniques used to design optimisation and approximation algorithms may still inspire useful mechanisms for distributed resource allocation. We hope to address this issue in our furture work.

6 Conclusion

In this paper, we have given a brief overview of recent work on multiagent resource allocation in the context of the welfare engineering framework, and we have further analysed the properties of this framework for the case of k-additive utility functions. Our results presented in Section 4 show that, despite the positive expectations raised by the previous result on negotiation in additive domains (Theorem 2), the complexity of the negotiation protocol required to agree on a socially optimal allocation does not necessarily decrease for problems with k-additive utility functions when k gets smaller (as long as $k > 1$).

[6] This may be achieved, for instance, by adding an agent i to the system with $u_i(R) = 0$ for all $R \subseteq \mathcal{R}$, or by having at least one agent with a monotonic utility function.

On the other hand, as we have seen in Section 3, representing utility functions in the k-additive form rather than the bundle form can be significantly more succinct, particularly in cases where a representation with a small value for k is possible.

We have also explored connections to well-known combinatorial optimisation problems, which allowed us to establish complexity results for the problem of finding a socially optimal allocation with respect to different representations of utility functions (Section 5). In this context, we have also briefly discussed the relation of our negotiation framework to combinatorial auctions for different kinds of bidding languages. While our negotiation framework is clearly *not* and auction (it is, for instance, not concerned with the aspect of agreeing on the price for a set of items), the abstract "centralised" problem of finding a socially optimal allocation (which is not itself a problem faced by the agents participating in a negotiation process) directly corresponds to the winner determination problem in combinatorial auctions. Under this view, the languages used to represent utility functions correspond to bidding languages for such auctions. However, it appears that the bidding language corresponding to our k-additive form has not yet been exploited by auction designers.

Finally, we would like to stress that the high complexity of of our negotiation framework does not, at least not necessarily, mean that it cannot be usefully applied in practice. This view is supported by the fact that, in recent years, several algorithms for winner determination in combinatorial auctions (a problem of comparable complexity to the problems arising in the context of welfare engineering) have been proposed and applied successfully [12, 20, 22].

We see the work presented in this paper as part of a wider research trend, which brings together ideas from different areas including microeconomics, game theory, complexity theory, and algorithm design. Some further examples of this kind of interdisciplinary research are cited in the introductory section.

Acknowledgements. We would like to thank Jérôme Lang for many fruitful discussions on the topics addressed in this paper, and in particular for suggesting to consider k-additive utility functions. We would also like to thank Patrick Baier, Marco Gavanelli, and Jerôme Monnot for insightful discussions on NP-completeness, as well as Cristina Bazgan for comments on a previous version of this paper.

References

[1] K. J. Arrow. *Social Choice and Individual Values*. John Wiley and Sons, 2nd edition, 1963.

[2] K. J. Arrow, A. K. Sen, and K. Suzumura, editors. *Handbook of Social Choice and Welfare*, volume 1. North-Holland, 2002.

[3] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation: Combinatorial Optimization Problems and their Approximability Properties*. Springer-Verlag, 1999.

[4] E. Boros and P. L. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123(1–3):155–225, 2002.

[5] P. E. Dunne. Extremal behaviour in multiagent contract negotiation. *Journal of Artificial Intelligence Research*, 2004. To appear.

[6] P. E. Dunne, M. Wooldridge, and M. Laurence. The complexity of contract negotiation. Technical Report ULCS-03-002, Department of Computer Science, University of Liverpool, 2003.

[7] U. Endriss and N. Maudet. On the communication complexity of multilateral trading. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2004)*. ACM Press, 2004.

[8] U. Endriss and N. Maudet. Welfare engineering in multiagent systems. In *Engineering Societies in the Agents World IV*, volume 3071 of *LNAI*. Springer-Verlag, 2004.

[9] U. Endriss, N. Maudet, F. Sadri, and F. Toni. On optimal outcomes of negotiations over resources. In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2003)*. ACM Press, 2003.

[10] H. Fargier, J. Lang, M. Lemaître, and G. Verfaillie. Partage équitable de ressources communes: (2) Éléments de complexité et d'algorithmique. *Technique et Science Informatique*, 2003. To appear.

[11] R. Feldmann, M. Gairing, T. Lücking, B. Monien, and M. Rode. Selfish routing in non-cooperative networks: A survey. In *Proceedings of the 28th International Symposium on Mathematical Foundations of Computer Science (MFCS-2003)*, volume 2747 of *LNCS*. Springer-Verlag, 2003.

[12] Y. Fujishima, K. Leyton-Brown, and Y. Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-1999)*. Morgan Kaufman, 1999.

[13] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.

[14] M. Grabisch. k-order additive discrete fuzzy measures and their representation. *Fuzzy Sets and Systems*, 92:167–189, 1997.

[15] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*. Plenum Press, 1972.

[16] J. M. Kleinberg, C. H. Papadimitriou, and P. Raghavan. Segmentation problems. In *Proceedings on 30th Annual ACM Symposium on Theory of Computing (STOC-1998)*. ACM, 1998.

[17] M. Lemaître, G. Verfaillie, and N. Bataille. Exploiting a common property resource under a fairness constraint: A case study. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-1999)*. Morgan Kaufmann, 1999.

[18] H. Moulin. *Axioms of Cooperative Decision Making*. Cambridge University Press, 1988.

[19] N. Nisan. Bidding and allocation in combinatorial auctions. In *Proceedings of the ACM Conference on Electronic Commerce (EC-2000)*, 2000.

[20] M. H. Rothkopf, A. Pekeč, and R. M. Harstad. Computationally manageable combinational auctions. *Management Science*, 44(8):1131–1147, 1998.

[21] T. W. Sandholm. Contract types for satisficing task allocation: I Theoretical results. In *Proceedings of the AAAI Spring Symposium: Satisficing Models*, 1998.

[22] T. W. Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135:1–54, 2002.

[23] A. K. Sen. *Collective Choice and Social Welfare*. Holden Day, 1970.

[24] D. H. Wolpert and K. Tumer. An introduction to collective intelligence. Technical Report NASA-ARC-IC-99-63, NASA Ames Research Center, 1999.