Nash Social Welfare in Multiagent Resource Allocation

MSc Thesis (*Afstudeerscriptie*)

written by

Sara Ramezani Khorshid Doost
(born March 25th, 1982 in Rasht, Iran)

under the supervision of **Dr Ulle Endriss**, and submitted to the Board of Examiners in partial fulfillment of the requirements for the degree of

MSc in Logic

at the *Universiteit van Amsterdam.*

Date of the public defense:
November 7th, 2008

Members of the Thesis Committee:
Dr Ulle Endriss
Prof Dr Peter van Emde Boas
Prof Dr Krzysztof Apt
Prof Dr Han LaPoutré
Dr Stéphane Airiau
Drs Joel Uckelman

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

Contents

Acknowledgments

First I thank God for giving me this opportunity and the strength to fulfill it. All praise is due to him.

I am grateful to Dr. Ulle Endriss, my advisor. Ulle was a great example of what an advisor should be, keeping steady contact with me all the time I worked on this thesis and having helpful suggestions at every step. He was even more than that, extremely responsible and going out of his way to accommodate for the time limitations I had for completing this work, I really appreciate that.

The best part of my studies in Amsterdam have been the many friends I have found here, each coloring a part of my life here with their unique hue. I am grateful to all of them, and would particularly like to thank a few here:

Inés, for always being there when I needed her, caring and understanding, as a sister is.

Thomas & Hendrike, for being the ideal neighbors and friends, and generous enough to lend me their "jukebox" for such a long time!

Tunahan, for his genuine "goodness", and deep understanding of friendship; and for the many nights he spent with Vahid and I over dinner and endless cups of tea talking and talking, that we missed so much after he left.

Vanessa, for all of the summer days the two of us spent working on our theses in the MOL room until it was dark, and the friendship that crept in somewhere in between.

I would also like to thank Joel Uckelman, for generously providing me with his source code, and some fruitful conversations. Dr. Benedikt Löwe for his useful help and advice, especially when it was most needed. Tanja, for always being so helpful, and her contagious calmness. Niels, for not only helping me through the labyrinth of Dutch bureaucracy, but never failing to provide an entertaining conversation while doing so. The Beth Foundation, for financially supporting my first year of studies at the ILLC.

I am also grateful to members of my thesis committee who have accepted to read this and attend my defense session, I hope they find it worthwhile.

Finally, I thank Vahid for loving me, no matter what.

The time that I have spent studying in Amsterdam has been particularly challenging, not empty of its ups and downs, joys and especially hardships. But I have a feeling that wherever fate takes me from here, I will not only take memories of Amsterdam with me, but also a dream. A dream of cycling in a narrow alley alongside a canal with the cool breeze on my face that will wake me every once in a while with a unique feeling of utter freedom.

Amsterdam Sara Ramezani
October 25, 2008.

Abstract

Multiagent resource allocation studies the distribution of resources among agents in different ways depending on the criteria that are to be satisfied. The allocation of resources can be carried out in a centralized or distributed manner. The resources may be discrete or continuous, sharable or not, and the criteria may range over a wide array of different requirements, for instance optimizing various social welfare functions or fairness criteria. Many such problems have been studied extensively in the literature.

The Nash social welfare function (also referred to as the Nash collective utility function) is the product of the utilities of individual agents. Because of its mathematical structure, increasing NSW gives a balance between increasing the utilitarian welfare of the society, which is the sum of the utilities of agents, and fairness among agents. This dissertation aims at studying multiagent resource allocation with indivisible unsharable goods with regard to Nash social welfare. We study various properties of the Nash collective utility function in this context such as convergence of agent negotiation, communication and computational complexity. We also devise and implement a new heuristic algorithm for solving the problem of optimizing Nash social welfare, carry out some experiments in this regard and analyze their results.

Chapter 1

Introduction

This thesis is a study in multiagent resource allocation focusing on optimizing Nash social welfare. Multiagent resource allocation (MARA) is concerned with distributing a set of goods or resources among a society of agents in a way that satisfies specific criteria. It can involve a wide array of problems that fall into different fields, usually combining concepts from different disciplines in any single approach. It is particularly becoming more popular in recent years due to the various applications increasingly being found for distributed multiagent software systems and particularly automatic negotiation procedures.

Nash social welfare is a concept of collective welfare inspired by the famous solution of John Nash for the bargaining problem [44]. It proposes the product of utilities of individuals as a measure for collective welfare of a society. It postulates that a state of society in which the product of utilities of agents is maximized is the best in terms of collective welfare [52, 42]. The appeal of this approach is that it encourages the increase of the average utility of the agents while also being sensitive to fairness. In this regard it can be considered as a compromise between two of the major approaches to collective welfare, namely the utilitarian and egalitarian collective utility functions. The former only considers the sum of utilities of the agents as and the latter the minimum utility of all agents as the criterion for evaluation of collective welfare. Utilitarian social welfare is completely insensitive to fairness, and egalitarian social welfare is unconcerned with the average welfare of society as long as their is a single poor agent. Nash social welfare combines both aspects in a rather elegant way. We shall compare these three approaches thoroughly in Chapter 3.

Different aspects of MARA are studied in this work. But there is a common theme throughout, a unifying thread that stitches these seemingly mismatched pieces together, and this is Nash social welfare. All of the problems addressed and all of the results presented are oriented at optimizing Nash social welfare, an approach that has been totally new in most of the contexts we have investigated, and has proven to be interesting in many of them.

The rest of this thesis is organized as follows. There has been a conscious effort to keep the text self-contained, as well as concise, therefore any general concept is briefly introduced before diving into it. Chapter 2 is a brief but rather comprehensive introduction to multiagent resource allocation. In this chapter, the basic components of MARA problems such as resources, agents, preferences etc . are defined and their variations are discussed. Problems that arise in various aspects of MARA are also introduced.

The next chapter involves the theory of social welfare. In Sections 3.1 and 3.2 the main concepts of the theory and the ideas behind them are introduced. Then we go on to define three important concepts of social welfare, namely utilitarian, egalitarian, and Nash social welfare, and compare them and present their properties. In the last section some axiomatic properties of the proposed notions of collective welfare are studied, and an interesting formulation of the three main social welfare orderings discussed in the previous section is presented.

Chapter 4 contains most of the theoretical results of our work. We first introduce the distributed negotiation framework for MARA problems. Then we go on to define a particular kind of deal that increases Nash social welfare, and prove some results on the convergence of sequences of such deals and the possible necessity of particular kinds of deals. In Section 4.3, upper bounds on the number of deals in a negotiation sequence are proved.

Chapter 5 contains some results on the computational complexity of computing Nash social welfare in MARA, namely some formulations of the problem are proven to be NP-complete.

In Chapter 6, a heuristic algorithm for the computation of states optimal in Nash social welfare is presented in the context of centralized MARA (also known as *combinatorial auctions*). The algorithm is designed for cases where a particular type of logic-based languages are used to represent the agents' preferences. A brief report of some of the results of experimenting this algorithm are also presented. The last chapter concludes.

Chapter 2

Multiagent Resource Allocation

The problem of distributing resources among a set of agents has been studied extensively in recent years [31, 62, 63, 41]. The various aspects of it involve notions, methods and perspectives from a variety of different fields including economics (more particularly microeconomics), computer science and artificial intelligence, decision theory and social choice theory. While the problems and questions that arise in each of these fields and their approaches may differ, they unify in the effort to gain insight and solve various aspects of the same general problem of *multiagent resource allocation* (MARA).

The computer scientist may be interested in finding the complexity of computing a particular optimal allocation or devising algorithms for finding such allocations, the AI scientist may also design algorithms with a focus on experimentation and simulating results, the economist or social choice theorist may be interested in the effects of using different social welfare orderings in different contexts, and the game theorist in designing efficient mechanisms to control the behavior of agents in a particular way. This makes MARA have a lot to offer for each of these fields, and also causes interdisciplinary synergy among them.

One reason for the growing research in MARA is it's usefulness in many real world applications. These applications are particularly increasing with the emergence of internet bargaining, negotiation, and e-commerce procedures in general [28]. Among other applications that make use of resource allocation procedures are network routing [57], manufacturing and scheduling [61, 47, 56], fair exploitation of earth observation satellites [36, 37], public transportation [3], logistics [20, 49], airport traffic management [32], crisis management [40], and grid computing [41, 27].

The basic issue in MARA is distributing some resources among a number of agents. This distribution is usually aimed to be so as to satisfy particular criteria. For instance we may aim at optimizing a particular notion of social welfare based on the agents' preferences, the most common of which is the utilitarian social welfare. In this dissertation we concentrate on the Nash social welfare ordering

[52, 42] and study some MARA problems while focusing on optimizing social welfare in this sense. We will present the concept of social welfare orderings and collective utility functions, and particularly Nash social welfare in Chapter 3.

The rest of this chapter is a brief introduction to MARA. We define and discuss the different parameters that are influential in MARA problems and study the effect that various modeling choices may have on the problems. The structure of the following sections follow [4] to some extent.

2.1 Resources

The resources that are distributed may be *divisible* (like a bulk product that can be divided by weight) or indivisible. They also may be *sharable* by multiple agents (e.g. a network connection or earth observation satellites) or not. Throughout this thesis we assume that we have a finite set of indivisible unsharable resources. It is also assumed that the resources are *single-unit*, i.e. there is only one resource of each kind, and that they are *static*, do not change (do not perish, are not consumed, etc.) in the process of allocation.

It is also assumed that there are a finite number of agents. Generally a partitioning of the resources (goods) among the agents is called an *allocation*. If some of the resources are not allocated to any agent the result would be a *partial allocation*. The set of resources allocated to an agent is commonly referred to as a *bundle*.

2.2 Preferences

2.2.1 Cardinal and Ordinal Preferences

In order to be able to assess the desirability of each possible distribution of resources among the agents, they must somehow announce the appeal that each alternative has for them. This is usually referred to as the preference of the agents. It is common for agents to represent their preferences via a *preference structure*. Preference structures are mathematical models used to represent preferences, preference structures are generally either *cardinal* or *ordinal*.

When using ordinal preferences, agents only specify between any two states, which one they prefer. In other words each agent has a reflexive and transitive binary relation $\preceq$ that is defined on the set of alternative states. The agent prefers a to b or is indifferent between them if $b \preceq a$, strictly prefers a to b (or $b \prec a$) if $b \preceq a$ but not $a \preceq b$, and is indifferent between them or $a \sim b$ if $b \preceq a$ and $a \preceq b$. The relation may be required to also be complete depending on the model.

A cardinal preference structure on the other hand assigns a numeric or qualitative value to each of the alternatives, so it is in the form of a function from the set of alternatives to a set of numeric values such as the real or natural numbers (or linguistic valuations such as good, bad, etc. if qualitative values are being used). Values used in this way to express preferences are usually called utilities. An extensive theory exists on the use of utilities, how they should be defined, their benefits and drawbacks [1, 33, 22].

An ordinal preference can be induced from any cardinal preference structure, by setting $a \preceq b$ whenever the utility of b is higher than a and $a \sim b$ whenever they are the same. So cardinal preferences may be considered to convey more information than ordinal preferences in the sense that they can express the intensity of preferences that can lead to a more precise evaluation of alternatives. It also can allow for *interpersonal comparisons* e.g. comparing two different agents' happiness with their status in a particular allocation. This may be used in some models to compensate agents with side-payments in the process of negotiation. Although whether this is possible depends on the properties of the certain model, since the agents' preferences may for instance be represented in different metrics or currencies, and thus be incomparable in this sense. We will see in the next chapter that the Nash collective utility function is basically the only collective utility function that can overcome this issue of scalability.

In the rest of this work we will generally use cardinal preferences with numeric values for utilities. It is also assumed that an agent's utility in any particular allocation depends only on the items that he owns and not by what is owned by others, or in other words it is free of *allocative externalities*. This means that the utilities of agents can be defined on the sets of resources, so an agent would be indifferent with respect to two allocations in which it owns the same set of goods (note that in some scenarios we consider, there may also be monetary transfers between agents, so their utility may additionally depend on the amount of money they have in a particular state, but it will still not depend on other agents' resources).

Note that although defining the utility function on the set of bundles simplifies the problem to some extent, the number of bundles is still exponential in the number of resources and thus is a complex problem. This complexity has a combinatorial nature, the agents can have any possible utility function, for instance they may value only bundles with a particular set of items, or only bundles without a particular set of items, or bundles with one of a particular set of items, or with some item and without another item, or even completely random utilities with no observable patterns, etc.. The number of different possibilities increase exponentially with the number of resources.

Another important choice for any model is the way in which agents represent their preferences, we will discuss this in Section 2.2.3.

2.2.2 Some Properties

Before we go on to discuss preference representation, some useful properties that utility functions may have are defined in this section. In the following it is assumed that $\mathcal{R}$ is a finite set of resources and $u : 2^{\mathcal{R}} \rightarrow \mathbb{R}$ is a utility function with real numbers as values, defined on the bundles of resources.

2.2.1. Definition. A utility function u is:

- a *non-negative* utility function if and only if $u(X) \geq 0$ for all $X \in 2^{\mathcal{R}}$,
- *normalized* if and only if $u(\{\}) = 0$,
- *monotonic* if and only if for any $X, Y \in 2^{\mathcal{R}}$, if $X \subseteq Y$ then $u(X) \leq u(Y)$,
- *dichotomous* if and only if for all $X \in 2^{\mathcal{R}}$, either $u(X) = 0$ or $u(X) = 1$,
- is *additive* if and only if for all $X \in 2^{\mathcal{R}}$, $u(X) = \sum_{x \in X} u(\{x\})$,
- a *0-1 function* if and only if it is additive an the utility of each single-item bundle is either 0 or 1,
- *modular* if and only if $u(X \cup Y) = u(X) + u(Y) - u(X \cap Y)$ for all $X, Y \in 2^{\mathcal{R}}$,
- *submodular* if and only if $u(X \cup Y) \leq u(X) + u(Y) - u(X \cap Y)$ for all $X, Y \in 2^{\mathcal{R}}$,
- is *supermodular* if and only if $u(X \cup Y) \geq u(X) + u(Y) - u(X \cap Y)$ for all $X, Y \in 2^{\mathcal{R}}$.

Note that a utility function is additive if and only if it is modular and normalized.

2.2.3 Preference Representation

Agents may use different languages for identifying their preferences, the choice of an appropriate language is very much dependent on the application at hand. A MARA problem in a particular setting may be best represented by a specific language, but it is not always obvious which language would be a better one. In this section we briefly introduce some representation languages used in the literature along with some of their properties.

Bundle Enumeration

The simplest way to represent the preferences would be for agents to list all of the possible bundles along with the utility they assign to them. We shall call this representation the *bundle enumeration form* or simply *bundle form*. In general such a representation could be exponential in the number of resources (the total number of possible bundles is two to the power of the number of resources).

In this representation usually only the bundles with non-zero utilities would be required to be listed. So it is actually not always as bad as it seems, since in many common applications agents do not have a specific non-zero value for each of the exponential possibilities. This is because it is cognitively very hard for humans to actually have preferences over such a large number of alternatives. It is often the case though that preferences can be represented more succinctly in other languages that may have a more complex structure. For instance in order to represent a preference structure that assigns the number of items in a bundle as its utility, all bundles would need to be listed in this language. It is possible to represent this utility function much more succinctly as we shall see below. Bundle enumeration is a fully expressive language, i.e. any utility function may be represented in this manner.

k-Additive Representation

In many application domains agents can easily specify a value for each single item or small sets of items, but their value for larger bundles is simply the sum of the values assigned to its subsets. This is in conformation with the principle stated previously that agents may not be able to evaluate many bundles precisely, particularly if they are large. This observation has lead to the use of *k-additive* functions in MARA. The idea originates from fuzzy measure theory [26] and was first used in the MARA context by Chevaleyre et al. [5] and independently in combinatorial auctions by Contizer et al. [9].

A utility function is k-additive if the utility of each bundle can be computed as the sum of corresponding coefficients of its subsets with k or less members. To represent such functions, suppose the coefficient corresponding to the subset T of bundles is α^T, then the utility assigned to a bundle can be computed via the following formula:

$$u(A) = \sum_{T \subseteq A} \alpha^T.$$

So there is a constant coefficient corresponding to each bundle of at most k goods, and the function can be fully specified in this way. The size of the representation would be polynomial (in the number of resources) if the size of k is constant and not dependent on the number of items, or if k is not small but there are many redundant bundles with zero coefficients.

It is easy to show that any utility function has a unique representation in the k-additive form if k is large enough. For some functions k would have to be the same as the number of items which can result in listing a non-zero coefficient for all of the possible bundles, an exponentially large representation.

But for many utility functions the k-additive form can actually result in a more succinct representation, as they can be represented via a k-additive function with small k. For example the function described in the previous section in which the utility of each bundle is its cardinality can be represented by a 1-additive (or modular and in this case also additive, since it is normalized) function that has a 1 coefficient for the bundle consisting of each of the single resources. This is linear compared to the exponential representation needed in the bundle enumeration form.

Note that although the k-additive representation may be more succinct than the bundle form in many natural cases, it is not always so. They are actually incomparable in this respect. There are in fact classes of utility functions (especially non-monotonic ones) that are representable in linear space (with respect to the number of resources) in the bundle form, but need an exponential representation in the k-additive representation [5]. The class of utility functions that assign a positive utility to all single-resource bundles, and zero to all others are an example of such functions. In order to represent such functions in the k-additive form, the absolute value of the coefficient corresponding to each bundle must be equal to the product of the number of items in it and the given positive value, positive for bundles containing an odd number of items and negative for bundles of an even number of items. For instance consider the utility function u defined as follows where x is some real number:

$$u(A) = \begin{cases} x & \text{if } |A| = 1 \\ 0 & \text{otherwise} \end{cases}$$

In the k-additive form this utility function would have the following unique representation:

$$\alpha^T(A) = \begin{cases} -x|A| & \text{if } |A| \text{ is even} \\ x|A| & \text{if } |A| \text{ is odd.} \end{cases}$$

OR, XOR and OR* Languages

Some of the languages used in MARA originate from the combinatorial auctions literature. Actually combinatorial auctions can be considered as MARA problems with a centralized allocation method as shall be discussed in Chapter 6. The most common language used in this domain is the OR language. In this language an agent represents her preferences (commonly referred to as bids in the auction settings) by a set of bundle and value pairs. Any set of mutually disjoint bundles of the agents may be accepted by the auctioneer. The utility that the agent assigns to an allocation (the amount he is willing to pay in the auction) is equal the sum of the values of the accepted bids.

For example suppose an agent has the following bid:

$$\langle\{a\}, 3\rangle \; OR \; \langle\{b\}, 1\rangle \; OR \; \langle\{a, b\}, 5\rangle$$

where a and b are single resources. This means that the agent assigns value 3 to a alone, 1 to b, and 5 to both items together. Alternatively it could be said that the agent is willing to pay 5 for the bundle containing a and b. The OR language is not fully expressive, as it cannot represent utility functions that are subadditive [45], for instance in this example there is no way to value both items at a price that is less than the sum of their individual prices.

The XOR language is on the other hand fully expressive, but is usually less succinct than the OR language. In the XOR language there are again bundle and value pairs, but they are connected by XOR this time. So only one of the pairs can be selected for each agent. It is thus possible to list all possible bundles and their values, and hence the expressiveness.

The comparison between the succinctness of these two languages can again be seen considering the example where the value of each bundle is the number of items in it. The OR language can express this language with the one pair of value 1 for each single item bundle, while the XOR language would need an exhaustive listing of all possible bundles.

There have been efforts for combining the succinctness of the OR language with the expressiveness of the XOR language. Some results have been the OR-of-XOR, XOR-of-OR, and OR/XOR languages and the OR* language [45]. The first two consist of the OR of a number of XOR bids (only one pair of each internal XOR bid can be chosen, but many of them can be chosen through the OR as long as they do not overlap), and the XOR of the OR of a number of bids. The OR/XOR language can consist of pairs connected by OR and XOR in any possible way. In the OR* language it is possible to introduce *dummy* items into the model in order to make disjoint bundles overlap by adding the same dummy item to both and thus allowing for subadditive utility functions without losing the succinctness of the OR language.

Weighted Propositional Formulas

Some representation languages in MARA make use of logical formulas explicitly for representing preferences [35]. The most common idea is for agents to use propositional formulas in order to represent their preferences for bundles of resources. For this means, a propositional variable p_a is defined for each resource a for each agent. In any allocation, a propositional variable is true for a particular agent if he owns the corresponding resource. The agent can thus use pairs of propositional formulas and values to represent her preferences, a set of such pairs that an agent uses to express his utilities is called a *goal base*. Different methods with varying complexity have been proposed in this framework. In the simplest case each agent can have a single pair to represent his preferences. Then he assigns utility equal to the value of the pair to any allocation in which the formula is satisfied. The use of propositional formulas allows for the succinct representation of specific varieties of utility functions.

For instance an agent with preference $(p_a \vee p_b, 4)$ would assign value 4 to all bundles containing at least one of the items a or b and zero to all other bundles, Preference $(p_a \wedge p_b, 3)$ would assign 3 to bundles containing both a and b, and $(\neg p_a, 5)$ a value 5 only to all bundles that do not contain a. So it can easily be specified that only one of a subset of items is needed (e.g when one of a set of identical resources is wanted), or that some items are desirable only when they are acquired together (left and right shoe), or that some items are not desirable in any circumstances. Other more complicated preferences of this sort may be constructed by combining the connectives to make more complex formulas.

In the next level of complexity, the each agent's goal base can contain multiple formulas and value pairs. Then an agent's utility in a particular state is usually computed as the sum of the values of the formulas satisfied for the agent. These languages are further discussed in Chapter 6.

Subsets of languages of weighted propositional formulas, with various degrees of expressiveness and succinctness, can be used as the preference representation language depending on the needs of a particular application or model. We shall use a subset of these languages with multiple pair goal bases that use only clauses containing conjunctions of positive literals to represent utilities in the model of Chapter 6.

2.3 Centralized vs. Distributed Allocation

The *allocation procedure* is another important issue in MARA. There have been various methods used in this field. The two main branches are negotiation protocols such as the ones we discuss in Chapter 4 [50, 18, 17], and auction protocols which are mostly different forms of combinatorial auction methods [10]. The main distinction of these two branches is in whether the allocation procedure is *centralized* or *distributed*.

In a centralized method, a central authority is responsible for allocating the items to agents. This authority will receive the agents' preferences in some format and method that is specified by the protocol being used. Then the authority will compute an allocation satisfying whatever measure that should be satisfied by some particular algorithm and assign the goods to the agents based on the computed allocation. Auction protocols are a common example of such procedures. Common protocols are English or Dutch auctions that are open bid, everyone knows everyone else's preferences, and respectively increasing and decreasing. There are also various sealed bid auctions where the agents are not aware of other agents' preferences. Because of the combinatorial nature of the agents' preferences in MARA, the auctions that arise in this field are those normally designated as combinatorial auctions.

Auction protocols are simple and clear in the sense that once a given communication protocol and allocation method are agreed upon, all agents submit their preference to the auctioneer, and he is the one who does all the computing. There are however some drawbacks to these techniques. One is that it is not clear who should act as an auctioneer. There are many applications where it is hard to find an individual entity that can be trusted by all agents as an auctioneer and does not benefit from the outcome in any way. Another problem is that computing a solution in combinatorial auctions can become very hard (it is in fact in many cases intractable as we shall see later), and putting all the burden of computation on the shoulder of one entity may not be the best idea. The third drawback is that considering the complexity of the agents' preferences and the possibly limited amount of communication between the agents and the auctioneer (particularly if we consider systems of automatic agents that communicate via a limited bandwidth which are becoming more and more common), it may be impossible to use a centralized approach in some applications.

A solution to these problems is introducing distributed protocols for resource allocation. In such protocols each agent acts on her own, and depending on the protocol, can propose a deal and accept or reject deals proposed by other agents. Such protocols are commonly known as negotiation protocols. Each agent would then be responsible for part of the computation and could accept or reject the deal if it does not increase her personal welfare or that of society in some sense or the other. Also agents would not be required to send large chunks of data, such as their full preference orderings, on the communication network. The distributed negotiation framework is particularly fit for applications where we are dealing with independent agents that are actually distributed on a network and have limited contact with each other. The actual protocol may depend on how much the agents can know of the preferences of other agents and the topology of the communication network.

Many of the problems arising in MARA may have similar or even identical solutions in particular centralized and distributed cases (e.g. the complexity of finding an allocation that is optimal in some sense). In the following we consider a distributed negotiation framework in Chapter 4. Chapter 6 that introduces a heuristic algorithm in the combinatorial auctions framework. Some of the theoretical solutions are valid in both settings, namely the complexity results in Chapter 5.

The choice of criteria for evaluating an allocation, or a social welfare criteria is another important aspect in MARA, although as mentioned before in many cases the utilitarian concept of social welfare is taken as default. As we specifically focus on the choice of a social welfare criteria, we shall discuss social welfare in a separate chapter which follows.

Chapter 3

Social Welfare

This thesis is an effort in investigating multiagent resource allocation with the aim of optimizing Nash social welfare in mind. Nash social welfare is a social welfare ordering designed so that it balances total utilitarianism [29] (increase of agents' average utilities without taking fairness into account) and fairness or egalitarianism. It favors states where the average utilities are higher as long as they are not "too" unfair, i.e. as long as they have a higher product of utilities.

So we are interested in social welfare in the context of MARA, which is not the usual perspective in this field, as utilitarian social welfare is the criterion usually considered and there are rarely studies that take fairness issues into account, some exceptions include [6, 2, 39]. In this section we study social welfare theory and define the most common welfare and fairness criteria in this theory and compare their properties. We specifically focus on Nash social welfare since it is the criterion that we shall focus on in the next chapters. The terminology of this chapter follows that of Moulin [42, 43]. Also, in this chapter the presentation shall not be restricted to the case of MARA. The problem of finding a feasible state that is best in terms of the collective welfare of the individuals of a society and the various solutions to it will be generally studied. It is assumed that the individuals (which we may still call agents) have utilities for each of the possible states. To restrict this model to MARA it suffices to suppose that the states are just the allocations.

3.1 Social Welfare Orderings

The central concepts of social welfare theory are fairness and justice. Fairness is not always easy to achieve, even its meaning in different situations is not obvious. The notion of equality though is a central notion when considering justice, it has been postulated since ancient times that justice is nothing but treating equals equally and unequals unequally. The approach of social welfare theory [52, 54, 42, 43] can be considered as opposed to the utilitarian program in which it is supposed that individuals should be left to do whatever increases their individual utilities, and the state that arises should be a desirable one. This utilitarian approach has obvious shortcomings, the most important one being that, as we shall see, it is insensitive to fairness.

While utilitarianism can somewhat be ensured to emerge in a society with self-interested agents, there may not be obvious incentives for agents to observe rules that ensure fairness, unless we suppose that they are interested in the overall welfare of society. Thus in many cases in the context of welfarism it is supposed that a *benevolent dictator* is in charge of deciding the state of the world, and all members of the society will follow whatever is decided. Although, in different applications, individuals would prefer outcomes adhering to different definitions of welfare, so a central authority is not always needed, and distributed methods can also be used. Also, the quest for fairness is strongly rooted in human nature and thus it is not always hard to find incentives for its implementation.

We should also take into account that agents are not always humans or self-interested agents. Thus achieving a particular notion of social welfare may be the optimum state for all agents in an application, i.e. any agent may prefer the socially optimum state even thought there is another state where her utility is higher.

Generally the aim is to find a state where the collective welfare of the society is optimum in some sense or the other. In order to do so there should be a means for comparing different states. Since the satisfaction of the individuals with each state is solely dependent on their utility functions, we must be able to compare different utility profiles. A *utility profile* is a listing of the utilities of all agents in a particular state; a vector $u(x) = (u_1(x), \cdots, u_n(x)) \in \mathbb{R}^n$ where $u_i(x)$ is the utility of agent i in the particular state x (the states in the parentheses may be omitted when it is not relevant in the computations).

3.1.1. Definition. A *social welfare ordering* (SWO) is a transitive and complete binary preference relation $\preceq$ on utility profiles. Complete means that for any two utility profiles u and u', either u is preferred to u' ($u' \prec u$), or u' is preferred to u ($u \prec u'$), or they are indifferent. Transitive means that for any utility profiles u, u' and u'', if $u' \preceq u$ and $u'' \preceq u'$, then $u'' \preceq u$.

We shall define different notions of social welfare in the form of social welfare orderings and their cardinal partners, *collective utility functions*.

One of the main concepts in social welfare is Pareto optimality, named after an Italian economist of the 19th and early 20th century. When comparing two utility profiles, one is said to be *Pareto superior* to the other if no agent prefers the latter to the former, and at least one strictly prefers the former. More formally:

3.1.2. Definition. A utility profile u is Pareto superior to another utility profile u' if and only if $u_i \geq u'_i$ for any agent i and $u_j \geq u'_j$ for some agent j . A utility profile is *Pareto optimal* if no other utility profile is Pareto superior to it. A social welfare ordering $\preceq$ is said to have the *Pareto optimality property* if whenever a utility profile u is Pareto superior to another profile u', $u \prec u'$.

In other words a state is Pareto optimal if it is not possible to find another state in which the utility of one agent increases without the utility of any other agent decreasing. Pareto optimality is usually a very basic property that should be satisfied by any preferred state, since if there is a Pareto superior state, there would be no plausible reason for not choosing it. Pareto optimality is also commonly referred to as efficiency or fitness. As stated above satisfying the Pareto optimality property can be considered as a requirement for any reasonable SWO, but note that it is not itself a social welfare ordering since it is transitive but not complete, e.g. of the utility profiles $(4, 2)$ and $(3, 5)$, neither is Pareto superior to the other.

Other basic properties that should arguably be satisfied by any SWO are *monotonicity* and *symmetry*.

3.1.3. Definition. A social welfare ordering $\preceq$ is *monotonic* if increasing a single agent's utility ceteris paribus results in a state that is preferred by the SWO. In other words, if u and u' are utility profiles such that $u'_i < u_i$, and for all $j \neq i$, $u'_j = u_j$, then $u' \prec u$.

So monotonicity means that a state should be preferred to another if an agent is better off in it and all agents are the same in the two states. It is similar to Pareto optimality in a way, if an SWO is monotonic, then its maximal elements among any set of feasible utility profiles are always Pareto optimal.

3.1.4. Definition. A social welfare ordering $\preceq$ is *symmetric* if any two utility profiles that are a permutation of one another are be indifferent according to that ordering.

Symmetry ensures that all agents are treated equally. Again it would be very unreasonable to treat agents that are in the same state differently, as it would contrast the most basic concepts of fairness and equality.

3.2 Collective Utility Functions

Collective utility functions (CUF) are in short the cardinal version of social welfare orderings.

3.2.1. Definition. A *collective utility function* f is a function defined from the set of utility profiles to the real numbers $f : \mathbb{R}^n \to \mathbb{R}$.

The value of a CUF for a particular utility profile is a numeric value that is an index of its collective utility. It is possible to define CUFs corresponding to a particular SWO, i.e. a CUF f that yields the same ordering as the SWO $\preceq$, i.e. $u' \prec u$ if and only if $f(u) < f(u)$, and $u' \sim u$ if and only if $f(u) = f(u)$. In fact almost all common SWOs have a naturally defined corresponding CUF, with the exception of the leximin ordering which we shall discuss later.

3.3 Utilitarian Social Welfare

Utilitarian social welfare is one of the most common notions of social welfare.

3.3.1. DEFINITION. The *utilitarian social welfare ordering* is an SWO that prefers a utility profile $u = (u_1, \cdots, u_n)$ to a utility profile $u' = (u'_1, \cdots, u'_n)$ if and only if $\sum_{i=1}^{n} u_i > \sum_{i=1}^{n} u'_i$. Correspondingly the *utilitarian collective utility function* is defined as $sw_U(u) = \sum_{i=1}^{n} u_i$.

The utilitarian SWO thus ranks utility profiles based on the sum of utilities of individuals (or the average utility of an individual since profiles with the same number of individuals are always compared). It is the embodiment of the utilitarian philosophy that postulates that the overall utility of society should be the only determining factor in social welfare. It is thus totally indifferent when it comes to fairness, e.g. the two utility profiles $(0, 100)$ and $(50, 50)$ are equivalent in terms of utilitarian social welfare, and $(0, 101)$ would actually be preferred to $(50, 50)$, a small raise in the total utility is worth the extreme unfairness, from the utilitarian point of view.

3.4 Egalitarian Social Welfare

Let the result of reordering a utility profile $u = (u_1, \cdots, u_n)$ in increasing order be denoted by $u^* = (u^*_1, \cdots, u^*_n)$.

3.4.1. DEFINITION. The *egalitarian social welfare ordering* is an SWO that prefers a utility profile $u = (u_1, \cdots, u_n)$ to a utility profile $u' = (u'_1, \cdots, u'_n)$ if and only if $u^*_1 > u'^*_1$, i.e. the egalitarian SWO prefers the profile in which the utility of the worst-off individual is higher. The *egalitarian collective utility function* is defined as $sw_E(u) = u^*_1$ or the utility of the worst-off agent.

The egalitarian SWO [34] contrary to the utilitarian SWO focuses solely on fairness. No matter how high the average utility of the society is, from the egalitarian viewpoint it has no effect as long as there is a single individual with a low utility level. The egalitarian and utilitarian SWOs can be considered to be two extremes in the spectrum of welfare orderings, the former ignoring average utility in favor of the least fortunate, and the latter disregarding unfairness completely for the sake of increasing total social utility.

As an example, consider the utility profiles, $u = (1, 10, 10, 10)$, $v = (1, 1, 1, 1)$, and $w = (1, 1, 1, 10)$. They have the same egalitarian collective utility of 1, although they differ in terms of both average utility levels and equality. The egalitarian SWO takes only the utility of the agent with lowest utility as a determining factor and ceases to distinguish between profiles that have the same lowest utility level. It is easy though to extend the egalitarian idea so that it has a higher degree of resolution and can be a distinguishing factor in more cases, for instance the ones in the example above. This can be done by the leximin SWO [11]:

3.4.2. Definition. The *leximin social welfare ordering* is an SWO that prefers a utility profile $u = (u_1, \cdots, u_n)$ to a utility profile $u' = (u'_1, \cdots, u'_n)$ if and only if u^* is lexicographically superior to u'^*, i.e. if for some $1 \leq k \leq n$ we have $u^*_i = u'^*_i$ for all $i = 1 \cdots k-1$, and $u^*_k > u'^*_k$.

It is obvious that the leximin ordering is compatible with the egalitarian SWO, but can distinguish between a much larger number of pairs of utility profiles; it will actually strictly prefer one of them up to symmetry, i.e. if they are not permutations of each other, while the egalitarian SWO will only strictly prefer one if their lowest utility level differs. The leximin ordering is sometimes referred to as *practical egalitarianism*. In the example above, the leximin ordering would prefer u to w to v, which would be reasonable as it prefers the state where there are only a few poor to one where a minority monopolizes most of the utility, and still prefers that to the equal division of poverty.

The leximin SWO does not have an obvious natural CUF corresponding to it, in fact it can be shown that *no* collective utility function can represent the leximin SWO [43].

3.5 Nash Social Welfare

The Nash SWO strikes a balance between focusing exclusively on fairness and only on total utility.

3.5.1. Definition. The *Nash social welfare ordering* is an SWO that prefers a non-negative utility profile $u = (u_1, \cdots, u_n)$ to a non-negative utility profile $u' = (u'_1, \cdots, u'_n)$ if and only if $\prod_{i=1}^n u_i > \prod_{i=1}^n u'_i$. The *Nash collective utility function* corresponding to this SWO is defined as $sw_N(u) = \prod_{i=1}^n u_i$.

The idea of using the product of utilities is appealing for various reasons. Using the product ensures that the average utility of society does not decrease too much since the product will also decrease if it does. On the other hand it encourages fairness since in a constant average utility level, the profile with equal utilities for all is always the one with highest product of utilities. Intuitively this is similar to the mathematical fact that a square is the largest (in terms of area) rectangle you can make with any given piece of string.

Since the Nash utility function is defined as the product of utilities, it would only be meaningful to define it on non-negative utility profiles (profiles in which the utilities of all individuals are non-negative). This is because if some utilities are negative, the outcome of the Nash CUF would not be continuous, it would actually fluctuate unreasonably between positive and negative values depending on whether the number of individuals with negative utilities is even or odd. In many cases it would even be better (and even required in some of the scenarios we investigate) for all the utilities to be strictly positive. A single zero utility in the profile would make the CUF insensitive to changes in the rest of the utilities in the profile, e.g. the profiles $(0, 2, 6)$, $(3, 5, 0)$, and $(4, 6, 0)$ would have the same collective utility value. This is not a good idea since the second profile is obviously more fair than the first one and the third profile is Pareto superior to the second one.

These observations can be reflected in the Nash CUF if we substitute a very small positive value ϵ instead of the zeros (which will make the list increasing in Nash collective utility), but is lost when zeros are used. This is a drawback in many cases where there is for instance no feasible state where all agents have a non-zero utility but a comparison based on Nash CUF could still be useful. Also substituting ϵ where the utility may be zero otherwise could be compatible with the spirit of the Nash CUF in the sense that not rewarding an agent with any utility would result in a drastic decrease in the collective utility, also having multiple agents with ϵ utility would decrease the utility very fast.

On the other hand though, the use of zero utilities may be justified in applications where it is known that there are always feasible profiles where all agents have strictly positive utilities and leaving an agent with no utility would be so unacceptable that it deserves such penalizing (although using an ϵ that is sufficiently small will still work for these cases also). In concluding this discussion note that as we have seen in the example above, allowing for zero values results in the Nash CUF not satisfying the monotonicity and even Pareto optimality properties (in the strict sense at least, i.e. a Pareto superior state would not be least preferred in the Nash SWO, but equivalent; the same goes for the monotonicity property).

The following example due to [43] illustrates the advantage the Nash SWO can have in comparison with the utilitarian and egalitarian viewpoints.

3.5.2. Example. There is a common workspace in which the radio is always on, and the different employees' taste for radio channels differ, or similarly roommates that can not agree on a TV channel. It is possible to share time between the channels. The question is how should the time be distributed among the channels so that it is fair towards all agents?

Let's formulate the problem as follows. There are n individuals and k channels, the fraction of time each channel i is broadcast is t_i so that $t_1 + \cdots + t_k = 1$. Each person likes only one channel to be broadcast, and the her utility is equal to the fraction of time that channel is broadcast. Also suppose that the number of people who prefer channel i is n_i (note that only channels with at least one fan are considered so $n_i \neq 0$ for all i).

From the viewpoint of utilitarian social welfare, one of the channels that has the highest number of fans should be always broadcast (if there are more than one of these channels, the time can be shared between them also). This solution totally disregards all agents that prefer a less popular channel in order to increase the overall utility, since allowing a channel with fewer fans to be broadcast will result in a lower average utility when such channels are being broadcast.

The egalitarian solution to this problem would be to broadcast each channel $\frac{1}{k}$ of the time, regardless of the number of individuals that prefer it. This is because in any other kind of division, one of the channels would be broadcast less than $\frac{1}{k}$ of the time, and thus the resulting profile would be worst in terms of egalitarian social welfare.

None of these two proposed solutions are very appealing. In the first case for instance, if one channel has two fans and all other channels have a single fan, the channel with two fans will be always broadcast no matter how large n is, leaving all but two particular agents always unhappy. In the egalitarian case, if one agent likes one channel and all other agents like another one, each will be broadcast half of the time leaving all but one specific agent unhappy half of the time. Neither of these solutions makes use of the relative values of n_i in computing the solution, and this seems to be the main problem here.

Now the solution that the Nash SWO prefers is to broadcast each of the stations $\frac{n_k}{n}$ of the time; each channel is broadcast in an amount of time proportional to its popularity. So if a channel is more popular, it simply gets more broadcast time. This solution has neither of the problems explained above and seems to be somewhere in between the two approaches, combining the merits of each, while avoiding their drawbacks. It also makes meaningful use of the relative popularity of the channels and is generally appealing in terms of fairness.

The Nash SWO works exceptionally well in this example, although it might not always so obviously be the best choice in other scenarios. There are many aspects to fairness and so different concepts of social welfare can be useful in different problems. Finding the one that fits a problem at hand is itself a challenge, and should be done very carefully. Moulin provides a good deal of insight into the various aspects of fairness and the choice of an appropriate social welfare measure with many demonstrative examples in his book [43].

3.6 Axiomatic Formalization

In this section we introduce some other properties that can be defined on social welfare orderings. We briefly explain the intuition behind each of these properties, and the SWOs that satisfy each one. These properties can also be seen as axioms on the set of SWOs [43, 42]. Finally we sketch why the three SWOs that we have focused on are the ones significant for investigation in another sense, namely that they are notable elements of a family of CUFs characterized by a set of desirable and reasonable axioms [43]. We mention three SWOs because the egalitarian and leximin SWOs can be considered as the same, since the leximin SWO is an extension of the egalitarian SWO as mentioned before. So, we may use the term *egalitarian* to refer to the broader notion also covering the leximin SWO or practical egalitarianism.

The first property we discuss is a very basic one that should generally be satisfied by any SWO. This property states that if an agent has the same utility in two different states, he should not be influential in the comparison between them, so only agents whose utilities are affected by the change should have an effect on the outcome of the comparison. The idea is that if the agents' utility levels are the only factor in determining the collective welfare, then it will make no sense for the outcome to depend on unconcerned agents.

3.6.1. Definition. A social welfare ordering $\preceq$ over positive utility profiles is *independent of unconcerned agents* if the agents whose utilities are the same in two utility profiles are not influential in their comparison, i.e. for any $u, v, w \in \mathbb{R}^n$ such that $w_i = 0$ wherever $u_i \neq v_i$, if $v \preceq u$ then $v + w \preceq u + w$. This property is also known as *separability*.

If we restrict our choice of collective utility functions to continuous ones, then it can be proven that the set of CUFs that are independent of unconcerned agents are exactly those that are *additive* –could be represented as $f(u) = \sum_i g(u_i)$, for an increasing function $g : \mathbb{R} \to \mathbb{R}$. This result is known as the Debreu-Gorman Theorem and is originally due to [12] and [25], also presented in [43]. (Note that the definition of an additive CUF is different from that of the additive utility function defined in the previous chapter).

The next property is the *Pigou-Dalton transfer principle*. It is the most basic property for ensuring fairness among agents. Based on this axiom, a change in the utility profile is desirable as long as the difference in utility of the agents involved in it decreases, hence they become more equal.

3.6.2. Definition. A social welfare function $\preceq$ satisfies the *Pigou-Dalton transfer principle* if it prefers or is at least indifferent to any change in the utility profile that is mean-preserving and inequality-reducing and involves only two agents. In other words $v \preceq u$, if $u_k = v_k$ for all $k \neq i, j$: $u_i + u_j = v_i + v_j$, and $|v_i - v_j| \geq |u_i - u_j|$. A shift from one state to another that is preferred based on this principle (e.g. from v to u in the above formulas) is called a *Pigou-Dalton transfer.*

It is easy to see that the Nash social welfare ordering always strictly prefers the state arising after a Pigou-Dalton transfer. The egalitarian and leximin orderings prefer it in some cases where it involves agents that are poor enough, they are indifferent in other cases. The utilitarian SWO does not strictly prefer such a transfer but is actually indifferent towards it as it is mean-preserving. An SWO that does not satisfy this property would in fact have a bias *against* fairness.

Now if we restrict CUFs to those that satisfy the Pigou-Dalton property and are independent of unconcerned agents (additive), the resulting CUF would be an additive function $f(u) = \sum_i g(u_i)$ such that g is concave [43, 53].

The next two properties involve the effect of changing the scale of the utility profiles on the SWO.

3.6.3. DEFINITION. A social welfare ordering $\preceq$ is *independent of common scale* if re-scaling the utility functions of all agents with the same factor does not change their underlying ordering, i.e. for any two positive utility profiles u and v, and any positive scalar λ, if $v \preceq u$ then $\lambda v \preceq \lambda u$.

This property means that it is irrelevant for the SWO with which metric the utilities of the agents are expressed, as long as they all use the same scale. For instance the utilities of the agents can be expressed in different monetary units, without an effect on their ordering. All of the SWOs we have discussed have this property. Continuing our previous discussion, if we further require the CUFs to be independent of common scale as well as satisfying the independence of unconcerned agents and Pigou-Dalton principles, it can be shown [43] that the resulting CUFs would be (up to a multiplicative constant) of one of the following forms :

$$W_p(u) = \sum_i (u_i)^p, \qquad 0 < p \leq 1$$

$$W_0(u) = \sum_i \log u_i$$

$$W^q(u) = -\sum_i (u_i)^{-q}, \qquad 0 < q < +\infty.$$

Note that this family of functions is actually continuous since W_0 is the limit of W_p and W^q as p and q go towards zero respectively (this is easy to see using the $z^p = e^{p \log z} \simeq 1 + p \log z$ approximation that is valid when $p \to 0$). It is also obvious that the utilitarian CUF is equivalent to W_1.

Furthermore, W_0 is actually an alternative representation of the Nash SWO:

$$\begin{aligned} sw_N(u) > sw_N(u') &\iff \prod_i u_i > \prod_i u'_i \\ &\iff \log(\prod_i u_i) > \log(\prod_i u'_i) \\ &\iff \sum_i \log(u_i) > \sum_i \log(u'_i) \iff W_0(u_i) > W_0(u'_i). \end{aligned}$$

So although the Nash CUF –which is used more commonly than this logarithmic version– not additive, the Nash SWO can be represented by an additive CUF.

The last significant observation about this family of CUFs is that the social welfare ordering represented by W^q as $q \to +\infty$ is in fact equivalent to the leximin ordering.

Therefore the three main notions of social welfare that we have introduced, and are basically the ones most studied in the literature are all represented in this significant family of collective utility functions. A family that characterizes three major axioms, independence of unconcerned agents, Pigou-Dalton transfer principle, and independence of common scale.

Finally, we consider a property similar to Definition 3.6.3 but more restrictive. It requires for the ordering to remain the same even if only a single agent changes the scaling of her utility function. It is formalized as follows.

3.6.4. Definition. A social welfare function over positive utility profiles $\preceq$ is *independent of individual scale of utilities* whenever for any $u, v, w \in (\mathbb{R}^+)^n$, if $v \preceq u$ then $v \cdot w \preceq u \cdot w$ ($\cdot$ denotes the inner product operator, i.e. $u \cdot w = (u_1 w_1, \cdots, u_n w_n)$).

It turns out that the Nash collective utility function is uniquely characterized by this property [43]: it is not only independent of individual utility scales, but any SWO that has this property must be represented by the Nash CUF (to see this note that the scalars multiplied in the utilities of the agents result in the same number –product of all scalings, $w_1 \times \cdots \times w_n$ in the definition– being multiplied in the collective utility in all cases and hence does not change the ordering). This can in many cases be an advantage. The outcome of the egalitarian and utilitarian CUFs can be manipulated by agents announcing their preferences in a scale that makes their utilities look smaller or larger respectively, but the Nash CUF would not be affected in this way. This can be seen in the following example that has been designed after a similar example from [43]:

3.6.5. Example. Consider a problem with two agents 1 and 2, and three states A, B, C with the following utility functions:

	A	B	C
1	15	10	5
2	3	5	6

In this case the utilitarian, egalitarian and Nash SWOs would select A, C or B, and B respectively as the optimal state.

If the utilitarian SWO is used as the measure of social welfare, agent 2 has incentive to change the outcome, since the state that is worst for him will be chosen. If the framework allows for agents to change the individual scales, agent 2 can just scale up his utility values by a factor of 4 so that the utilities are:

	A	B	C
1	15	10	5
2	12	20	24

Now the utilitarian SWO would prefer state C with welfare 29 instead of A with welfare 27, and agent 2 will have it as he pleases.

Similarly if the egalitarian CUF is used as the index for assessment of welfare, agent 1 would prefer to change the outcome and could change the outcome by scaling down her utility function by a factor of 5, resulting in the following utility values:

	A	B	C
1	3	2	1
2	3	5	6

The optimum egalitarian choice would then be A, as the first agent had intended.

The Nash social welfare will remain the same in all three of these circumstances, and thus such tricks could not be carried out if it is used as a measure of social welfare.

This last property distinguishes the Nash CUF from the other ones in the family of CUFs described above, and uniquely characterizes it.

Chapter 4

Distributed Negotiation Framework

In this chapter multiagent resource allocation is studied in a framework where there is no central authority and agents are free to negotiate their preferences of allocations by means of deals in which resources are interchanged among them. The main appeal of such a model is that the agents can be designated as independent entities (e.g. software programs) that make decisions based on their local preferences. The computational costs can also be distributed among the agents in this way.

There are a few different problems that can arise in this context. First is: considering a particular local criteria for the negotiation of agents, is it possible to characterize the properties of the state that would ultimately result? This question can also work in the opposite direction, which will give the problem a taste of mechanism design: is it possible to design protocols for negotiation that ensure a state with particular properties as outcome? Is it possible to guarantee a state that is optimal/suboptimal with respect to a specific definition of social welfare?

Other questions that would follow, particularly in a context where computation is an issue, are those of complexity. What is the complexity of reaching a desired state, in terms of communication (e.g. the number of deals/negotiation steps that are necessary/sufficient)? What is the computational cost of all of this?

Many have studied such problems in this area so far [50, 18]. There has been a particular focus on methods that aim at optimizing the utilitarian CUF [50], which, as we shall see, happens to be additionally appealing since it meets particular notions of *rationality* for individual agents. There has also been some work on other notions of social welfare [18, 6]. Our focus shall be on optimizing the Nash CUF. As far as we know there has been almost no effort in studying these problems with this aim before.

The rest of this chapter is organized as follows. In the next section, the formalities of the framework in which we study the problem are defined. The other two sections each cover one of the questions stated above, briefly reviewing the previous results, and presenting the novel ones. The first such section deals with negotiating optimal allocations of resources. The next one investigates communication complexity of the problem at hand. The problem of computational complexity is studied in the next chapter.

4.1 The Model

We study a model of distributed negotiation that has been employed previously in various studies [50, 18, 63]. In this model there are a finite number of agents who represent their preferences with quantitative utility functions and a finite number of unsharable indivisible resources. It is supposed that the agents start out in an initial allocation, and can agree on deals that result in changes in the allocation. We are interested in studying variations of this problem. For instance the effect of the agents using deals with different kinds of restrictions or agents with a specific type of utility function. The effect that these local properties can have on the overall outcome, and the complexity of the computations involved are some of the topics that we investigate. We do not however get involved with the study of how agents come to agree on a deal. It is generally supposed that if a particular type of deal exists, agents are somehow capable of finding and agreeing on it, this is however usually not an easy task and a topic of research in itself [55].

4.1.1 Distributed Allocation Problems

We define a *distributed allocation problem* as a triple $\langle \mathcal{A}, \mathcal{R}, \mathcal{V} \rangle$. $\mathcal{A}$ is a set of n agents $\mathcal{A} = \{1, 2, \ldots, n\}$, and $\mathcal{R}$ is a set of m resources $\mathcal{R} = \{r_1, \ldots, r_m\}$. $\mathcal{V} = \{v_1, \ldots, v_n\}$ is a set of valuation functions, one for each agent. The valuation functions are functions from the subsets of resources to the set of real numbers $v_i : 2^{\mathcal{R}} \to \mathbb{R}$.

In some cases it is useful to use *monetary side-payments*, or simply side-payments, in the negotiation process. This means that agents are able to adjust their utility levels more precisely by exchanging units of money that have a specific relationship with the utility units of the agents (usually a unit of money is equal to a utility unit). Side-payments may for instance be used to compensate agents that have a utility decrease by those who experience an increase, thus allowing for a larger range of acceptable deals when the agents are self-interested. This is allowable since we are using numeric utility functions that allow for the comparison of the intensity of the satisfaction between agents.

We now proceed with a formal definition of an allocation.

4.1.1. Definition. An *allocation* A is a function from the set of agents to subsets of the resources, $A : \mathcal{A} \to 2^{\mathcal{R}}$, such that $A(i) \cap A(j) = \emptyset$ if $i \neq j$, and $\cup_{i=1}^{n} A(i) = \mathcal{R}$. $A(i)$ represents the resources that agent i owns in allocation A.

So each allocation A is a partitioning of all of the resources among the agents. Provided that there are no monetary side-payments involved, the state of the system in any given time is represented by the triple $\langle \mathcal{A}, \mathcal{R}, \mathcal{V} \rangle$ and the allocation in that state. In such cases, the the *utility functions* of the agents, u_i, would be the same as their valuation functions. Therefore in a state where the current allocation is A, the utility of agent i would be $u_i(A(i)) = v_i(A(i))$, the two functions could thus be used interchangeably. Agents usually do not need to refer to the bundles of other agents (since there are no allocative externalities), so we use the notation $u_i(A)$ for $u_i(A(i))$ wherever it would cause no confusion.

In cases where there are monetary side-payments involved, the state of the system would also consist of a payment balance function $\pi : \mathcal{A} \to \mathbb{R}$ such that $\sum_{i \in \mathcal{A}} \pi(i) = 0$. This function assigns to each agent the amount of monetary deficit that he has in that state. Thus the utility function would thus be defined as a function on the valuation function and a monetary value $u_i : 2^{\mathcal{R}} \times \mathbb{R} \to \mathbb{R}$. It is usually defined as the difference of the value of the current allocation according to the agent and his monetary deficit $u_i(A(i), \pi(i)) = v_i(A(i)) - \pi(i)$, supposing that the value/utility and money are defined in the same unit. Again we may abbreviate this to $u_i(A, \pi)$. Whenever there are no monetary payments involved we use the terms *utility* and *value* and also *utility function* and *valuation function* interchangeably.

4.1.2 Deals

As mentioned, we suppose that the agents start off with some initial allocation and they can then change the state of the system by mutually agreeing on exchanging resources via *deals*.

4.1.2. Definition. A *deal* $\delta = (A, A')$ is a an ordered pair of distinct allocations A and A', it can be seen as taking allocation A to allocation A'. The set of agents *involved in* δ is $\mathcal{A}^{\delta} = \{i \in \mathcal{A} | A(i) \neq A'(i)\}$, the agents whose bundles are changed due to δ.

In cases where we have side-payments, a payment function p similar to π in its definition and condition needs to be specified along with the two allocations in each deal; $p(i)$ specifies the amount that agent i has to pay alongside the resource exchanges in the deal. If it is positive, agent i has to pay $p(i)$, and if it is negative she will receive $-p(i)$. Hence, the payment balance, π, represents for each agent, the sum of all payment functions from the initial state up to that point for that agent. Note that the amount that an agent pays another agent is not relevant, since a payment function can represent any sort of agreement between the agents by just adding up each agent's overall balance.

This definition covers a very wide range of deals, they can be very complex and involve any number of agents and resources. In many cases we would be interested in deals that cannot be further broken down into smaller independent deals. For instance if a deal consists of agent 1 giving r_2 to agent 2, and agent 3 giving r_5 to agent 4, it can be considered as two independent consecutive deals. The following definition captures this property.

4.1.3. Definition. A deal $\delta = (A, A')$ is *independently decomposable* if it can be decomposed into two deals whose sets of involved agents are disjoint, i.e. if there are two deals $\delta_1 = (A, A'')$ and $\delta_2 = (A'', A')$ such that $\mathcal{A}^{\delta_1} \cap \mathcal{A}^{\delta_2} = \emptyset$.

As we have mentioned above, we are interested in the effects of using different types of deals. The restrictions enforced on deals can involve their *structure* [50, 18] (e.g. restricting the number of agents or resources involved) or the welfare of agents (e.g. deals that have a particular effect on the utilities of the agents involved). The latter type of constraints have also been called *rationality* constraints [18].

One of the simplest kinds of deals in terms of structure are *1-deals* [50, 18] which correspond to a particular type of deals first studied in the classical Contract-Net [55].

4.1.4. Definition. A deal $\delta = (A, A')$ is a *1-deal* if it only consists of reallocating a single resource from one agent to another: there is only one resource r such that $r \in A(i)$ and $r \in A'(j)$ for $i \neq j$.

The rationality constraints may be different in terms of their locality. For instance in order to only accept the class of deals that increase the utility of the worst-off agent, each agent has to know the utility of all other agents in both allocations involved in the deal (in order to be able to identify the worst-off agent). On the other hand, if each agent only accepts deals that increase her utility level, she would not need to know more than her own utility level in the two allocations (it is supposed here that agents that are not involved in the deal do not have a role in deciding upon it). A particular class of constraints that we are interested in lies in between these two extremes, these are deals in which each agents needs to know no more than the utilities of the other agents that take part in a deal. Following [18] such deals are characterized by the criterion defined below.

4.1.5. Definition. A class of deals Δ can be characterized by a *local rationality criterion* if and only if there exists a predicate Φ defined over $2^{\mathcal{A}\times\mathbb{R}\times\mathbb{R}}$, such that $\delta = (A, A') \in \Delta$ if and only if $\Phi(\{(i, v_i(A),\ v_i(A')) \mid i \in \mathcal{A}^{\delta}\})$ holds.

Note that the predicate is defined only on the valuations of the agents that are involved in the deal. Also, in the explanation above we focused on the information that each agent needs to know, while when defining local rationality criteria, the predicate could be defined on the utilities of the agents involved in the deal. These are equivalent here since each agent would need all of this information in order to decide on the deal. The class of deals in the example above where the agents are self-interested would also count as a local rationality criterion since knowing the valuations of the agents involved would be enough to classify the deal. Another example of such a class of deals could for instance be those that increase the sum of utilities of the agents involved in the deal. The predicate that characterizes these deals can be defined as follows:

$$\Phi_u(\{(i_j, v_j, v'_j) \mid j = 1 \cdots k\}) = \top \iff \sum_{j=1}^{k} v'_j > \sum_{j=1}^{k} v_j.$$

4.1.3 Optimal Allocations

We shall make use of the notions of social welfare defined in the previous chapter throughout this chapter also. In our model, the set of feasible states of the system are those that can be defined by an allocation (and a valid payment balance function when there are side-payments). So the feasible utility profiles are only those that arise from such feasible states. The collective utility functions would therefore also be defined only on the feasible utility profiles.

We shall demonstrate the distributed allocation model and the use of social welfare orderings in this context via an example below.

4.1.6. Example. Consider a distributed negotiation problem without side-payments with $\mathcal{A} = \{1, 2\}$, $\mathcal{R} = \{r_1, r_2\}$, and the following value/utility function:

A	$v_1(A)$	$v_2(A)$
$\emptyset$	1	0
$\{r_1\}$	3	4
$\{r_2\}$	4	7
$\{r_1, r_2\}$	20	16

There are four possible allocations, lets call them A, B, C, and D, and define them as below:

$$\begin{aligned} A(1) &= \{\}, & A(2) &= \{r_1, r_2\} \\ B(1) &= \{r_1\}, & B(2) &= \{r_2\} \\ C(1) &= \{r_2\}, & C(2) &= \{r_1\} \\ D(1) &= \{r_1, r_2\}, & D(2) &= \{\}. \end{aligned}$$

The utilities corresponding to these allocations would thus be:

$$\begin{aligned} u_1(A) &= 1, \quad & u_2(A) &= 16 \\ u_1(B) &= 3, \quad & u_2(B) &= 7 \\ u_1(C) &= 4, \quad & u_2(C) &= 4 \\ u_1(D) &= 20, \quad & u_2(D) &= 0. \end{aligned}$$

Now let's investigate these utility profiles taking our different notions of social welfare into account (note that $u(X) = (u_1(X), u_2(X))$ denotes the utility profile corresponding to allocation X). First, based on utilitarian social welfare the profiles would have the following order: $sw_U(u(D)) = 20 > sw_U(u(A)) = 17 > sw_U(u(B)) = 10 > sw_U(u(C)) = 8$, it favors the two most unequal allocations because they have higher mean utility values. Based on the egalitarian perspective, $sw_E(u(C)) = 4 > sw_E(u(B)) = 3 > sw_E(u(A)) = 1 > sw_E(u(D)) = 0$. This time the two most unequal allocations are favored least due to the low utility of the poorer agent. The resulting ordering is completely reversed in comparison with that of the utilitarian SWO in this example. The leximin ordering will have the same result as the egalitarian SWO here since there are no ties. The ordering imposed by the Nash SWO, on the other hand, would be $sw_N(u(B)) = 21 > sw_N(u(A)) = 16 = sw_N(u(C)) = 16 > sw_N(u(D)) = 0$. Here one of the less equal allocations is ordered as high as the most equal one since the overall welfare is high enough to compensate for the loss in multiplication. The allocation that has the highest mean, but results in a zero utility for one of the agents is the worst, as we have discussed previously about this property of the Nash SWO.

Also, all of the four allocations are Pareto optimal since none of them is Pareto dominated by another one.

4.2 Negotiating Socially Optimal Allocations

4.2.1 Previous Work

Individual Rational Deals and the Utilitarian SWO

As it has been mentioned before, the most common approach to MARA problems, the distributed negotiation setting being no exception, is aiming at optimizing utilitarian social welfare. Another common approach is to suppose that the individual agents make choices that are *individually rational*, i.e. increase their individual utility. One of the most fundamental results in the distributed negotiation setting shows that these two approaches can be united when agents are myopic (here we mean that they do not look forward more than a single deal at a time). This is presented more elaborately in what follows.

First let's suppose that we have a model with side-payments, and the utility of an agent i in a state with allocation A and payment balance function π is $u_i(A, \pi) = v_i(A) - \pi(i)$. Now, the formal definition of individual rationality:

4.2.1. DEFINITION. A deal is *individually rational* if and only if there exists a payment function such that the utility of all agents would strictly increase once the deal is implemented with the corresponding payment function, except possibly for (some) agents whose valuation function does not change, whom can have zero payments.

Considering the definition of the utilities above, this would mean that if we are in a state with allocation A and payment balance π, a deal $\delta = (A, A')$ is individually rational if and only if for all agents i such that $v_i(A) \neq v_i(A')$, $u_i(A, \pi(i)) > u_i(A', \pi(i)+p(i))$, so $v_i(A)-\pi(i) > v_i(A')-\pi(i)-p(i)$, hence $v_i(A) > v_i(A')-p(i)$. So if there exists a payment function p such that $v_i(A')-v_i(A) > p(i)$ for all agents except for agents with $v_i(A') - v_i(A) \geq p(i)$ who may have zero payments, δ is an individually rational deal. In other words we may say that it is individually rational for all agents involved in it.

Also note that in any particular state with allocation A and payment balance π, $sw_U(A, \pi) = \sum_{i \in \mathcal{A}} u_i(A) = \sum_{i \in \mathcal{A}}(v(i)-\pi(i)) = \sum_{i \in \mathcal{A}} v_i(A)$, since the payment function adds up to zero for all agents, it can be disregarded in computing the utilitarian CUF.

Therefore the payment function should offer more than his loss for an agent that has a lower valued bundle in the deal by having the other agents whose value for the new bundle is higher pay, of course the latter group of agents should also not be made to pay more than their increase. It is obviously not possible to define such a payment function for any deal. The class of deals for which such a payment function exists are characterized in the following lemma which was first proposed in this form in [18]:

4.2.2. LEMMA. *A deal is individually rational if and only if it increases the utilitarian social welfare of the society of agents.*

Proof Sketch: If a deal $\delta = (A, A')$ is individually rational, by the definition of individual rationality $v_i(A')-v_i(A) \geq p(i)$ for all i, and there is at least one agent with $v_i(A') - v_i(A) > p(i)$, so by summing up both sides of the inequality for all agents we have

$$\sum_{i \in \mathcal{A}}(v_i(A') - v_i(A)) > \sum_{i \in \mathcal{A}} p(i) \Rightarrow sw_U(A') - sw_U(A) > 0 \Rightarrow sw_U(A') > sw_U(A).$$

For the other side of the proof suppose $sw_U(A') > sw_U(A)$, then using the payment function below, the deal is individually rational.

$$p(i) = v_i(A') - v_i(A) - \frac{sw_U(A') - sw_U(A)}{n}$$

□

This lemma shows that the utilitarian CUF is in fact the characterization of social welfare in the eyes of myopic agents that use the the above notion of individual rationality to make their choices of deals. The two concepts of individual rationality and utilitarian social welfare thus go hand in hand from this perspective.

Note that the lemma also makes it clear that individual rationality is in fact a local optimality criteria. This comes from the fact that whether a deal increases utilitarian social welfare only depends on the sum of the agents' valuation functions in the source and destination allocations, and since the utility of those who are not involved in the deal remains the same, if the sum of utilities of the agents involved increases, so does the utilitarian CUF, and vice versa.

4.2.3. Example. Consider again the distributed negotiation problem defined in the previous example (Example 4.1.6), this time when side-payments are possible. The utilities of the agents in this example was:

$$\begin{aligned} u_1(A) &= 1, \quad u_2(A) = 16 \\ u_1(B) &= 3, \quad u_2(B) = 7 \\ u_1(C) &= 4, \quad u_2(C) = 4 \\ u_1(D) &= 20, \quad u_2(D) = 0 \end{aligned}$$

Furthermore suppose that the agents start off at allocation C where agent 1 holds r_2 and agent 2 holds r_1. The utilitarian CUF for this allocation is $sw_U(u(x)) = 8$. Comparing to allocation D where the agents swap their bundles, we see that B has a higher utilitarian collective utility, so by the previous lemma, there must be a payment function that results in an increase in utility for both agents in the $\delta = (C, B)$ deal.

Indeed such a function can be defined, take for instance $p(1) = -2$ and $p(2) = 2$. The utilities of the agents after the deal would then be $u_1(B, p) = 3 + 2 = 5$ and $u_2(B, p) = 7 - 2 = 5$, which are higher than the value 4 that was the utility of both of them before the deal. So it is possible for agent 2 to give some of his excess utility to agent 1 since his gain is more than the other's loss. In fact any payment function in the class $p(1) = -x$ and $p(2) = x$ for $1 < x < 3$ could be used in this case, and it is generally supposed that since they exist, the agents can somehow agree on one of them.

Two more points are worthy of observation here. First that with a similar procedure it is possible to go from allocation B to A with a payment function like $p(1) = -x$ and $p(2) = x$ with $2 < x < 9$, and from there to allocation D with a payment function from the class $p(1) = x$ and $p(2) = -x$ where $16 < x < 19$. It is similarly possible to go from any of the intermediate allocations directly to D, which is the only allocation optimal in the utilitarian SWO, via a single individually rational deal.

The second point is that if we do not allow side-payments, then whichever allocation we start with, there would never be a deal that is individually rational (in the sense that the utility of all agents involved in it increases, or at least does not decrease). This shows the not-so-surprising fact that allowing monetary payments will make a larger range of deals possible, and leads to a higher possible utilitarian social welfare. This is also a result of the fact that all allocations in this example are Pareto optimal, since if an allocation was Pareto dominated by another one, the deal between them would not have reduced any agent's utility and increased at least one agent's utility level.

Note that even if there were an allocation that was Pareto dominated by another one, for instance if we change the problem so that $u_1(C) = 3$ so that C is dominated by B, there is still a difference between the case which involves money and the case that doesn't. In this altered example, since B Pareto dominates C, starting from allocation C it is possible to move to allocation B, but not to the two other allocations which are also Pareto optimal but do not dominate C. Therefore while it is possible to go to *any* allocation with higher utilitarian social welfare with individually rational deals with side-payments, but it is not so when monetary payments are not allowed.

The previous lemma can be used to give a more straightforward proof of the following fundamental theorem due to Sandholm [50].

4.2.4. Theorem. *Starting at any initial allocation, any sequence of deals that are all individually rational will end in a state that is optimum in utilitarian social welfare.*

Proof Sketch: The number of allocations are finite, and individually rational deals strictly increase utilitarian social welfare, so any sequence of deals will eventually end at some point. If, by contradiction, an eventual allocation is reached which is not optimal in utilitarian SWO, it is possible to define a deal from the eventual allocation to this allocation and this is an individually rational deal based on Lemma 4.2.2. Hence the aforementioned allocation is not an eventual allocation in the sequence as supposed, and the theorem is proven. □

This theorem shows the convergence of rational deals to a state with maximal utilitarian social welfare. It implies that the nearsightedness of the agents does not go against them in the outcome since proceeding with individually rational deals will not get them stuck in a local optimum, and they are always able to achieve an optimal allocation. Of course this does not say anything about the length of a sequence of deals (which actually may be very long as we shall see later), and more sophisticated agents may come to the optimal allocation more quickly. Also, this is a strong result since it guarantees convergence of all allocations, but the caveat is that there is also no restriction on the structure of the deals, and we have already explained what a wide range deals may fall into and their possibly very high complexity. This is exactly what is exploited in the following result that casts a shadow on the usefulness of the previous theorem. It is originally due to [50], although the restriction to individually rational deals was overlooked there and later corrected in [18].

4.2.5. Theorem. *For any fixed set of agents and resources, for any particular deal δ that is not independently decomposable there exists a choice of utility functions and an initial allocation such that δ would necessarily have to be included in any sequence of individually rational deals that leads to an allocation that has maximal utilitarian social welfare.*

This theorem implies that *any* deal may be necessary in the path to reaching an optimal allocation. Considering how complex a possible deal may be, it can be particularly hard for agents to come up with a deal that is needed at some point, and the process of reaching an optimal allocation may not be as smooth as the previous theorem makes it look.

In light of such negative results, there have been efforts for finding cases in which less complicated deals would be needed. One approach may be to restrict the utility function of the agents. Endriss et al. [18] have extended Theorem 4.2.5 to cases where all utility functions are monotonic and also cases where all utility functions are dichotomous. Work on additive functions, however, has proven more fruitful [18]:

4.2.6. Theorem. *In distributed negotiation problems where all utility functions are additive, any sequence of 1-deals will ultimately result in an allocation with maximal utilitarian social welfare.*

In addition to this, it is proven in [8] that the set of modular utility functions is maximal with respect to the given property, i.e. if agents' utility functions fall in any class that strictly includes the set of modular utility functions, 1-deals would not be enough to converge to an optimal allocation (in terms of the utilitarian SWO). Note that modular utility functions are slightly more general than additive functions, they allow the empty set to have a non-zero utility value.

Also, in [8] it is shown that individually rational deals in which at most k resources are reallocated are sufficient for convergence to an allocation optimal in utilitarian CUF, whenever all agents' utility functions are *additively separable* with respect to a common partition $\mathcal{R}$ and each set in $\mathcal{R}$ has at most k elements. A utiltiy function is additively separable with respect to a partition if there are no synergies between members of different sets in the partition, and the utility of a bundle is defined as the sum of utilities of each of the sets in the partition [21].

Other Social Welfare Orderings

Convergence and necessity results, similar to Theorem 4.2.4 and 4.2.5, have been proven [18] for distributed allocation problems without side-payments where the deals are *cooperatively rational* and the eventual allocations are Pareto optimal (this has already been illustrated in Example 4.2.3). Cooperatively rational deals are similar to individually rational deals, but with the less restraining constraint that a deal can take place if at least one of the agent involved is better off, and the rest are not worse off, i.e. the agents are willing to cooperate for the benefit of others as long as they do not lose anything. Additionally, they have shown that cooperatively rational 1-deals are sufficient to reach allocations optimal in utilitarian social welfare when all utility functions are 0-1 functions.

In [18], Endriss et al. have investigated the problem of taking other concepts of social welfare into account. In aiming to find deals that can optimize egalitarian social welfare, they have thus defined the notion of equitable deals.

4.2.7. DEFINITION. A deal $\delta = (A, A')$ is an equitable deal if and only if

$$\min\{v_i(A)|i \in \mathcal{A}^\delta\} < \min\{v_i(A')|i \in \mathcal{A}^\delta\}.$$

Namely a deal is equitable whenever it results in an increase in the utility of the worst-off agent among those participating in the deal, this property on deals is obviously a local rationality criterion. They further go on to prove that if a deal results in an increase in egalitarian social welfare, then it is equitable, and if it is equitable, then it will result in an allocation that is higher in terms of the leximin ordering. Finally they prove the following theorem:

4.2.8. THEOREM. *In a distributed negotiation framework with no side-payments, any sequence of equitable deals will ultimately result in an allocation that is optimum in the egalitarian SWO.*

A corresponding necessity theorem (similar to 4.2.5) is also proven that is also valid when all utility functions are dichotomous.

In [18] it is also proven that using only cooperatively rational or Pigou-Dalton deals will result in convergence to a *Lorenz optimal* [42] allocation when the utilities are 0-1 functions. We will not elaborate more on the concept of Lorenz optimality here.

4.2.2 The Nash Case

We are interested in optimizing Nash social welfare, and thus in knowing whether there exists a class of deals that could ensure convergence to an allocation optimal in Nash social welfare, as we have seen previously for other concepts of social welfare. The following class of deals turns out to be useful to that effect. Note that as we shall see shortly, in order for it to be meaningful to optimize the Nash CUF in the distributed allocation framework, all utilities of agents must be strictly positive. We are supposing here that there are no side-payments.

4.2.9. Definition. A deal $\delta = (A, A')$ is a *Nash deal* if and only if

$$\prod_{i \in \mathcal{A}^\delta} v_i(A) < \prod_{i \in \mathcal{A}^\delta} v_i(A').$$

So a Nash deal is a deal that locally increases the Nash CUF, i.e. increases the Nash CUF of those involved in the deal. It is easy to see from the definition that Nash deals can be characterized by a local rationality criterion. The following lemma shows that Nash deals actually characterize deals that result in an increase in Nash social welfare.

4.2.10. Lemma. *A deal is a Nash deal if and only if it increases Nash social welfare.*

Proof: Suppose we have a Nash deal $\delta = (A, A')$, then A' will be higher in the Nash SWO than A since the deal increases the product of utilities of agents involved in the deal, and the utilities of the rest are not changed and positive. Similarly, if the Nash social welfare of A' is higher than A, then since only the utilities of agents whose bundles have changed could have changed, and all utilities are positive, the deal $\delta = (A, A')$ must increase the product of utilities of those involved in it, and so it is a Nash deal. □

Note that the utilities being strictly positive is crucial in this lemma because if one of the agents not involved in the deal has a zero utility function, then the Nash social welfare will be zero in both allocations and the Nash deal would not result in an increase in Nash social welfare. The following convergence theorem uses the concept of Nash deals.

4.2.11. Theorem. *Starting from any initial allocation, any sequence of Nash deals will eventually lead to an allocation that is maximal with respect to the Nash CUF.*

Proof: Since the number of agents and resources are finite, the number of allocations are also finite. Each Nash deal strictly increases the Nash social welfare, so any sequence of Nash deals will eventually come to an end.

Now suppose that a sequence of Nash deals ends in an allocation A that is not Nash optimal. For any other allocation A^* that is Nash optimal, the deal $\delta = (A, A^*)$ would be a Nash deal by Lemma 4.2.10 since $sw_U(A^*) > sw_U(A)$. This is in contradiction with A being the end of a sequence of Nash deals and hence the desired result follows. □

This theorem is similar to the convergence results that we have seen in the previous section. It shows that any sequence of Nash deals are indeed sufficient to guarantee an optimal outcome and the process of negotiation would not terminate in a local optimum, but it is always possible to reach an optimal allocation. This is particularly interesting since agents can choose the Nash deals only depending on the effect of that particular deal on their Nash social welfare and do not need to compute any of the possible outcomes afterwards, and they still reach the optimal outcome. In other words, no backtracking is needed.

However there is a difference between these sequences of Nash deals and their individually rational counterparts. In the individual rational deals, given a deal and payment function, each agents is able to individually decide whether he would accept the deal or not, the acceptance depended on the utility of the single agents and the rule is such that it could successfully be carried out when the agents are completely selfish. The Nash deals on the other hand can only be carried out when all agents are interested in the overall welfare of the society and even ready to lose some welfare for the sake of the social welfare. This may not always be the case, but is relevant if we are in a context where all agents somehow benefit from increasing the collective welfare. It is also not possible for an agent to decide whether a deal is a Nash deal without knowing the utilities of other agents involved in it, this requires more communication between agents in the computation stage, and results in less autonomy for the agents.

An interesting point about the negotiation process with Nash deals is that starting with any initial allocation, it is possible to reach *any* of the possible Nash optimal allocations via negotiation (as long as the initial allocation is not Nash optimal in first place, if it is so no deal would be possible since Nash deals must strictly increase social welfare). This is a consequence of Lemma 4.2.10; if a state is not optimal, the deal from it to *any* Nash optimal allocation would be a Nash deal. This also holds for sequences of individually rational deals with side-payments, *any* allocation optimal in utilitarian social welfare can be reached from a given initial allocation as long as the initial allocation is not itself optimal. However it is not true for some of the other negotiation procedures we have shown convergence results for, particularly those without money. For sequences of co-operatively rational deals without money that lead to Pareto optimal allocations for instance it is not possible to reach any given optimal allocation. Sequences of 1-deals leading to allocations optimal in the utilitarian SWO when the utilities are additive also cannot reach any desired optimal allocation, but it is essentially different since the restriction on the structure of the deal imposes this limitation. For these procedures, for each initial allocation, it is always possible to reach *some* optimal allocation via a sequence of deals, but not necessarily all of the possible optimal allocations.

Two relevant questions arise at this point. The first, which was also addressed in the previous section, is how complicated may the deals needed in the process be? The second question is how long –in terms of number of deals– may the negotiation process be? We will address the first question in the rest of this section and the second one in the next section.

Before continuing, we should emphasize a fact about deals that are not independently decomposable which will be used in the proof of the next theorem. If a deal $\delta = (A, A')$ is independently decomposable, then the deal can be split into two independent deals $\delta_1 = (A, B)$ and $\delta_2 = (B, A')$ whose set of involved agents are disjoint. So the intersection of $\{i \in \mathcal{A} | A(i) \neq B(i)\}$ and $\{i \in \mathcal{A} | A'(i) \neq B(i)\}$ is empty. Therefore by the de Morgan rule the union of $\{i \in \mathcal{A} | A(i) = B(i)\}$ and $\{i \in \mathcal{A} | A'(i) = B(i)\}$ is the set of all agents. This means that if a deal $\delta = (A, A')$ is independently decomposable, then there is an allocation B different from both A and A', such that the bundle of any agent in B is either the same as her bundle in A or A'. Hence if a deal is not individually decomposable, such an intermediate allocation does not exist, i.e. for any allocation B distinct from both A and A', there exists an agent i whose bundle is neither the same as what he had in A nor the same as what his bundle in A'.

Now we continue with a result on the structural complexity of the Nash deals used in the negotiation process.

4.2.12. Theorem. *For a fixed set of agents and resources, for any particular deal δ that is not independently decomposable there exists a choice of valuation functions and an initial allocation such that δ would have to be included in any sequence of Nash deals that leads to a Nash optimal allocation.*

Proof: Suppose $\delta = (A, A')$, we shall construct utility functions for the agents such that δ is needed in any sequence of Nash deals that lead to a Nash optimal allocation. Since A and A' are distinct, there exists an agent j such that $A(j) \neq A'(j)$. Now consider the following utility functions:

$$v_i(R) = \begin{cases} 1 & \text{if } A'(i) = R \text{ or } (i \neq j \text{ and } A(i) = R) \\ \epsilon & \text{otherwise} \end{cases}$$

where ϵ is a very small positive value close to zero ($\epsilon \ll 1$). So $sw_N(A') = 1$, and by the hypothesis, in any allocation B that is different from both A and A', there is at least one agent whose bundle in B is different from what his bundle in both A and A' is. Also all agents assign the value ϵ to bundles they do not get in A or A'. It follows that at least one agent assigns the value ϵ to his bundle of resources in B, and hence $sw_N(B) \leq \epsilon$.

The Nash collective utility function in the allocation A is ϵ, since all agents except for j assign value 1 to their bundle in A, and j assigns value ϵ to her bundle. Now if with such a utility function we start at allocation A, with Nash collective utility of ϵ, the only state that has is higher in the Nash SWO is A', since we have already shown that all other allocations have a Nash CUF value of at most ϵ. It follows (Lemma 4.2.10) that $\delta = (A, A')$ is the only possible Nash deal, and thus it is necessary to use it in reaching the only Nash optimal allocation (A'). □

This necessity result shows that when using Nash deals to reach a Nash optimal allocation, it may be necessary to use *any* deal that is not independently decomposable, no matter how complex it may be, and such deals may definitely involve all agents and resources. This points out a drawback of the previous result, although an optimal outcome can always be reached, the sequence of deals may have to incorporate very complex deals. Furthermore it shows that the convergence result will not hold when the deals are restricted in *any* structural way.

Also observe that the requirement for deals not to be independently decomposable is crucial in this result. If a deal is independently decomposable, it can be decomposed into two subsequent deals that involve distinct sets of agents. It is also possible that both of these deals are Nash deals, in this case the original deal would not be necessary any more since the two sub-deals can be carried out one after the other, instead of the original deal.

It may be hypothesized that it might be possible to restrict the deals in terms of structural complexity if we impose constraints on the agents' utility functions (e.g. similar to Theorem 4.2.6). The following result shows that restricting utility functions to modular ones is not useful to this effect.

4.2.13. THEOREM. *For any set of agents and resources, if the number of resources is not less than the number of the agents, it is possible to define the utility functions of the agents and an initial allocation such that when only Nash deals are used, a deal that involves all agents may be necessary in order to reach an optimal Nash allocation, even if all of the utility functions are required to be modular.*

Proof: Suppose we have a distributed negotiation problem where the number of resources are at least as as many as the agents ($m \geq n$). Let A be any allocation in which each agent i owns resource r_i, $r_i \in A(i)$. We will use three parameters M, d, and ϵ in defining the utilities. Suppose that $0 < \epsilon < d < M$. We shall show that it is always possible to define these parameters such that a deal involving all agents would be necessary to reach the Nash optimal allocation.

The utility functions in k-additive form are defined such that $\alpha_i^T = d - \epsilon$ for $T = \{r_i\}$ for all $1 \leq i \leq n$, $\alpha_i^T = M - \epsilon$ for $T = \{r_{i-1}\}$ for all $1 < i \leq n$, $\alpha_1^T = M - \epsilon$ for $T = \{r_n\}$, and $\alpha_i^{\emptyset} = \epsilon$ for all i. This means that each agent gives utility d to allocation A, M to allocation A' that can be reached from A if each agent gives the resource corresponding to her index to the next agent in a circular manner (i.e. agent n gives r_n to 1), and ϵ to cases where they have neither of their two desired items. We want to show that it is possible to define M, d, and ϵ such that A' is the only allocation with higher Nash CUF than A.

So we have $sw_N(A) = d^n$ and $sw_N(A') = M^n$. Now for any deal starting from A in which the agents do not trade in the cycle specified above that takes A to A', the utility of the agents would not increase, since either resources that are redundant in the outcome (r_i with $i > n$) would be traded, or at least one of the agents would be deprived of the resource he values d, and his utility would drop to ϵ, without the utility of another agent increasing.

Thus the only other deals we have to consider are deals that involve trading on the specified cycle, but do not complete it. For instance a deal in which agent 2 gives r_2 to agent 3, and agent 3 gives r_3 to 4 is such a deal with length 2. Any deal of this sort of length i has the same effect. It will reduce the utility of the first agent in the chain from d to ϵ, increase the utility of the last agent to $M + d - \epsilon$, and increase the utility of the agents in between to M. In order for A' to be the only allocation that improves upon the utility of A, we need to define the parameters such that the Nash CUF is smaller than that of A in allocations resulting from all of these deals. The length of such deals can be between 1 and $n - 1$, so we must have:

$$\begin{aligned} \epsilon(M + d - \epsilon) &< d^2 \\ \epsilon M(M + d - \epsilon) &< d^3 \\ &\vdots \\ \epsilon M^{n-2}(M + d - \epsilon) &< d^n \end{aligned}$$

It is easy to see that for any M, d and n, there exists an ϵ that satisfies

$$\epsilon < \frac{d^n}{M^{n-2}(M + d)},$$

and that it satisfies the above properties. So A' would be the only allocation with Nash CUF higher than A, and thus starting from A, the deal (A, A') that involves all agents would be necessary. □

So restricting the utility function to the rather restrictive class of modular functions (note that the utility functions are also required to be positive here, as stated of all utility functions in this chapter) is not particularly useful in this case. However, by defining the utility functions such that the utility of each bundle is the product of the utility of all items in it (supposing that the utility of the empty set is equal to 1), 1-deals would always be sufficient to reach an optimal Nash allocation. This corresponds to the result of theorem 4.2.6 for additive utilities. This class of utility functions, which may be called *multiplicative*, is not exactly conventional, and it is not clear whether they would prove to be useful for any particular application.

As we have seen in Example 4.1.6 it is possible to increase Nash social welfare while decreasing utilitarian social welfare. This is rather obvious and precisely why it is not possible to identify individual rationality with Nash deals. It is however possible to define payments or *taxes* on the agents such that those who lose a great amount of welfare for the sake of society are compensated to some extent, but since the overall utility may decrease, it will not be possible to define taxes such that all agents' utilities increase (or do not change) if we consider utility and money to be on the same scale. However if agents' utility functions are defined such that money is valued exponentially more than the value assigned to bundles, e.g. $u_i(A, p) = \log(v_i(A)) - p(i)$, Nash deals would always be individually rational in the sense that for any Nash deal there will be a payment function that increases the utilities of all agents involved in the deal. Such utility functions do not seem to very reasonable though, and we may not be able to find any application for which they may be useful.

We have actually put some effort in finding different interpretations for which a procedure that leads to optimal Nash social welfare could be considered as a rational choice for the agents individually. The result has been that we have not been able to find a formulation better than the one stated above. It seems like for a negotiation procedure to lead to Nash optimal allocation some level of cooperation and commitment to collective welfare is required of the agents. There is however the possibility of the existence of some interpretation or procedure that we have missed (e.g. some negotiation procedure in which the Nash CUF does not increase in each step, but does converge eventually), but this is the insight we have obtained up to this point.

4.3 Communication Complexity

4.3.1 Introduction

The complexity of the negotiation procedure we have studied in this chapter can be investigated from various aspects. A general classification of the criteria characterizing some of these aspects has been presented in [16]:

- The number of deals needed to reach an optimal allocation.
- The number of messages that need to be exchanged between agents in order to reach each deal.
- The complexity of the computations the agents have to carry out in order to decide what to communicate in each step.
- The expressiveness of the language needed for negotiation.

The first three aspects cover the complexity of the negotiation process in a top-down approach, while the last affects the other aspects in parallel. In the first level the deals are seen as the unit of communication, the criteria for evaluating complexity in this level is the number of deals carried out in the negotiation process. In the second level, the communication needed between the agents to reach a single deal is the determining factor, this level is lower and more detailed than the previous one, since it takes the process of agreeing on a single deal into account. The third step takes the amount of calculations needed on behalf of the agents in any communication step within a deal into account, which is the most detailed level.

Communication complexity usually refers to the minimum number of bits of information various parties have to communicate in order to compute a function [64]. Following [16], we can consider communication complexity to refer to complexity in the first, second and fourth aspects of the ones listed above. This does not exactly conform to the conventional sense of this definition, but is however relevant. The communication complexity of reaching an optimal allocation is a function of the number of deals needed in a negotiation procedure and the communication complexity of the procedure itself. The communication complexity of a single deal in turn is a combination of the number of communication steps between the agents for a single deal and the amount of information conveyed in each one of these steps.

In the rest of this section we study the communication complexity of the distributed negotiation process involving Nash deals and leading to a Nash optimal allocation in the sense of the first aspect above, again following the approach of [16]. This is generally because we have not actually addressed different protocols for negotiation in our work, and studying the other aspects would need some details of the negotiation protocol.

The third aspect of complexity listed above is actually a *computational complexity* problem in nature. In [14] some work has been done in this effect. They have studied the complexity of the problem of deciding whether it is possible to go to an allocation with higher utilitarian social welfare by means of one-resource-at-a-time trading. They have proven this problem to be NP-complete. As stated above we do not actually study a problem of this sort in this thesis, but do address the computational complexity of computing a Nash optimal allocation in the general case, not in a negotiation process, in Chapter 5.

It is also important to take into account that in our protocol, deals involving multiple agents may very well be necessary in our negotiation framework. This complexity comes as the price of the simplicity of the agents in the model. Although this does not affect the first aspect that we have will consider in the following discussions, it does make the communication needed to reach a single deal substantially more complex. In fact it is not hard to see that if the cost of arranging a multilateral deal is proportional to the number of pairs of agents involved in it, it will increase quadratically compared to the cost of a bilateral deal. Also if the cost is proportional to the number of subgroups of agents involved, it will increase exponentially compared to the cost of a bilateral deal [19].

4.3.2 Results

In this section we will study the first aspect of complexity put forward above: how many deals does it take to reach an optimal allocation?

In our first approach to this question, we would like to know how many Nash deals would be absolutely required in a sequence of deals that leads to an optimal allocation. In other words what is the length of the shortest path between an initial allocation and a Nash optimal allocation? The following theorem answers this question.

4.3.1. THEOREM. *Starting with any initial allocation, it is always possible to reach an allocation maximal in the Nash SWO with at most one Nash deal.*

Proof: Starting from an initial allocation A, if it is Nash optimal then no deal would be needed. If it is not, then there exists some allocation A^* distinct from A that is Nash optimal and by Lemma 4.2.10 the deal (A, A') is Nash optimal. So a Nash optimal allocation can always be reached by at most one deal from any initial allocation. □

So there is always a path with length 1 to a Nash optimal allocation, no matter what allocation we start out with. This result may seem surprising, but it is actually rather trivial considering the lemma that we have previously discussed, although it is important in understanding the dynamics of the negotiation procedure. Actually the more interesting question here does not involve the shortest negotiation sequence, but the longest one. The emphasis of our main convergence result, Theorem 4.2.11, is that an optimal Nash allocation will eventually emerge, no matter which Nash deals the agents choose and agree on. It may be always possible to reach an (or actually any) optimal allocation via at most a single deal, but the agents may not be capable of computing such a deal (that can be *any* deal, as complicated as possible). The next result gives an upper bound on how long a negotiation process can get in the worst case, but we first need to prove the following lemma.

4.3.2. LEMMA. *For any given set of agents and resources, it is possible to define valuation functions such that any two distinct allocations have a different Nash CUF value.*

Proof: Lets assign to each agent i a prime number P_i so that $P_i \neq P_j$ whenever $i \neq j$. Now suppose each agent has an ordering on all possible bundles, and $v_i(R) = (P_i)^j$ if R is the j^{th} bundle in agent i's ordering. For any two distinct allocations A and A', there must be an agent k whose bundle is different in these two allocations. So the power of P_k is different in $sw_N(A)$ and $sw_N(A')$. Since any natural number has a unique prime factorization, $sw_N(A) \neq sw_N(A')$. Hence each two distinct allocations have a different Nash CUF value. □

This lemma lies at the heart of the following proof. Whenever all allocations have distinct values, it is possible to have a sequence of Nash deals of the maximal size:

4.3.3. Theorem. *Any sequence of Nash deals can consist of at most* $|\mathcal{A}|^{|\mathcal{R}|} - 1$ *deals.*

Proof: By Lemma 4.3.2 there are utility functions for which each allocation has a different Nash CUF value, suppose that we are dealing with a case with such a valuation function. There are a total of $|\mathcal{A}|^{|\mathcal{R}|}$ allocations so the worst possible case is to start from the allocation with the least NSW and go through all allocations in order of increasing Nash SWO in consecutive deals. All of the deals would be Nash deals since they increase the Nash CUF. Such a sequence would consist of $|\mathcal{A}|^{|\mathcal{R}|} - 1$ deals. It is obvious that a longer sequence is not possible since each deal has to strictly increase Nash CUF and there are only $|\mathcal{A}|^{|\mathcal{R}|}$ allocations. □

The positive contribution of this theorem is it is not possible to have an infinite sequence of deals. In fact it is not even possible to run into the same allocation twice when negotiating with Nash deals: this directly follows from the fact that Nash deals must strictly increase Nash social welfare. The downside is that it is in fact possible to go through every single allocation before converging to the optimal allocation, i.e. the structural upper bound set by the allocations not being repeatable is a strict one.

In [16], Endriss and Maudet have studied similar problems for a number of negotiation procedures. They have considered four cases with general and one-resource-at-a-time deals with or without money; we have presented some of the corresponding convergence results as Theorems 4.2.4, 4.2.6, and the result mentioned on reaching Pareto optimal allocations when there are no side-payments. They proved results that put upper bounds on the shortest and longest path in each case. All of these bounds are strict as those in Theorems 4.3.1 and 4.3.3 are, except for the upper bound on general rational negotiation without side-payments. Dunne [13] has also studied similar problems and proven upper and lower bounds in various negotiation procedures.

Chapter 5

Computational Complexity

In this chapter we consider the *computation complexity* of the problem of computing an optimal Nash allocation[1]. The work in this chapter does not necessarily address the complexity in a particular framework, but applies to the complexity of computing an allocation that is maximal in the Nash SWO in general. The basic concepts of the model we consider in this here is the same as that defined in the previous chapter, but it is actually more natural if we suppose that the allocation procedure is is centralized. The language that the agents use to represent their utilities is relevant to the complexity of the problem.

The similar problem of computing the maximal utilitarian social welfare in a combinatorial settings has been widely studied in the centralized setting and known in the combinatorial auctions literature as the *winner determination problem (WDP)* [10]. The problem that we study here can be considered as the *Nash winner determination problem.*

The problems addressed here are also different from the communication complexity issues raised in Section 4.3, even though the third aspect of complexity discussed there is a problem of computational complexity also. The level of computation involved in that problem is lower than the general problem we discuss here, although depending on the negotiation protocol, it may be possible to draw some relationship between them.

5.1 Introduction

Computational complexity deals with the inherent complexity of a problem in terms of the number of computational steps or amount of memory that is needed to compute a solution to the problem. In order to assess complexity in a consistent manner, problems have to be stated in a special format that is solvable by a standard model known as a *Turing machine*, and the amount of resources is

[1]Some results of this chapter are based on joint work with Vahid M. Hashemi.

computed as a function of the length of the input. Problems would usually be expressed as *decision problems*, problems that have a yes or no answer, in order to be compatible with this framework. A thorough treatment of the theory of computational complexity can be found in [46].

The complexity of a method or an algorithm is the worst case performance of it on any instance of the problem. The complexity of the problem is the complexity of the best algorithm that solves it. A problem is in the complexity class P, or solvable in *polynomial-time* if there exists an algorithm that can solve *any* instance of the problem in a number of computation steps that is polynomial in the length of its input. The complexity class P is generally considered as the class of *tractable* problems, since it is usually possible to effectively run algorithms of polynomial time complexity on machines that are technically available up to this day. In other words, The resources needed by problems that do not have a known polynomial-time algorithm would increase in so sharply with increase in input size, that solving slightly large instances of them proves practically impossible.

Another class of problems that is of particular interest is the class of *NP* problems. These are problems for which a *non-deterministic polynomial-time* algorithm exists. This is equivalent to saying that it is possible to verify whether a given solution is in fact a solution to the problem or not in polynomial time. It is not known whether problems in NP are solvable by polynomial-time algorithms or not (whether P=NP), but it is widely conjectured that they are not.

An important subclass of NP problems are the class of *NP-complete* problems. These are problems that are provably as hard as all problems in NP. By *hardness* it is meant that solving an NP-complete problem with any algorithm would lead to a solution for any other problem in NP with at most a polynomial-time overhead. The notion of finding a solution is formalized by a *polynomial-time reduction*. A problem A is reducible to B in polynomial time if for any instance of A there exists an instance of B that is computable in polynomial time such that the solution for the two problems are equivalent. This means that if a solution for B exists, then a solution for A can be computed with at most polynomial-time more effort, and thus B is at least as hard as A if both of them are not in P. A problem is *NP-hard* if it is at least as hard as all NP problems, but is not necessarily in NP itself.

Polynomial-time reductions to or from a problem whose complexity is known are generally used to assess the complexity of a problem. Particularly by providing a reduction from a known NP-complete problem to a problem at hand, it can be proven that it is also NP-hard. The theory of NP-completeness has been studied extensively in [24] and many of the well-known NP-complete problems, namely the ones we mention in what follows, are also defined there.

5.2 The Preference Representation Issue

We will also use reductions to show that the Nash winner determination problem is NP-complete. We study this problem in two different cases, using two different representation languages for the utility functions.

The representation language used for the utility functions is an important matter when studying the complexity of this problem. This arises form the fact that complexity is computed based on of the length of the input. The utility functions of the agents are a part of the input in the MARA problem, along with the agents and resources (or at least the number of them). The number of bundles is exponential compared to the number of agents and resources and since the utility functions are defined on the bundles, the representation may potentially be exponential also. In fact for any representation that is fully expressive, there are instances that can be represented no better than exponentially in that particular representation. Their differences lie in *which* instances can be succinctly expressed in each one, and they may be useful in different applications based on this quality.

Therefore when computing the complexity of various problems in the MARA framework, the complexity would be based on the size of input which also involves the utility functions. Since all instances are taken into account when evaluating complexity, there will always be instances in which the representation of utilities are exponentially more than the number of resources. Thus showing that these problems are hard is quite difficult.

Furthermore, because various representations may vary significantly in size, a complexity result using one representation does not automatically apply to other representations. So similar problems must be dealt with separately for different representation languages. For example we have seen in Section 2.2.3 that the k-additive representation may be exponentially more succinct than the bundle enumeration form and vice versa. In what follows we shall consider the Nash winner determination problem with each of these two representation schemes.

5.3 Results

In order to study the complexity of the winner determination and Nash winner determination problems, we first express these optimization problems as corresponding decision problems.

Utilitarian Welfare Optimization (UWO)	
Instance:	$\langle \mathcal{A}, \mathcal{R}, \mathcal{V} \rangle$, $K \in \mathbb{Q}$
Question:	$\exists$ allocation A : $sw_U(A) \geq K$?

Nash Welfare Optimization (NWO)	
Instance:	$\langle \mathcal{A}, \mathcal{R}, \mathcal{V} \rangle$, $K \in \mathbb{Q}$
Question:	$\exists$ allocation A : $sw_N(A) \geq K$?

We state these problems without specifying how the preferences are represented, that will be specified in the individual results. The UWO problem is generally known to be NP-complete [48, 14] when the utilities are represented in bundle form for the first time. The reduction used in [14] is from a variation of the 3-SAT problem. In [7] a more straightforward proof is presented that uses a variant of the weighted set packing problem that is tailored to exactly correspond to the problem. In [7] it is also proven (by a reduction from the maximum independent set problem) that NWO is NP-complete for the k-additive representation, even when k is equal to 2 (reduction from MAX-2-SAT).

As far as we have seen, there have not been any complexity results for NWO so far. In fact, we have only found one complexity result related to the Nash WDP in [59], proving that the Nash WDP is NP-complete when the utilities are represented as weighted propositional formulas that are 2-cubes (conjunctions of exactly two literals), and the goal base is aggregated by maximization instead of summation (the utility of each agent is the maximum among the weights of satisfied formulas).

The following theorem helps us prove the NP-completeness of NWO with the utilities expressed in bundle form.

5.3.1. THEOREM. *There is a polynomial time reduction from the UWO to NWO provided that the agents' utilities are represented in bundle enumeration form in both problems.*

Proof: Suppose that $\langle \mathcal{A}, \mathcal{R}, \mathcal{V} \rangle$ and $K \in \mathbb{R}$ is an instance of the UWO problem. We can reduce this to the NWO problem $\langle \mathcal{A}, \mathcal{R}, \mathcal{V}' \rangle$ and 2^K such that $v'_i(A) = 2^{v_i(A)}$ for any allocation $A : \mathcal{A} \to 2^{\mathcal{R}}$. Obviously this reduction needs no more than polynomial time in the size of the input.

To show that the two problems are equivalent, suppose that the there is an allocation A such that $sw_U(A) = \sum_{i \in \mathcal{A}} v_i(A) \geq K$, then we have for the NWO problem:

$$sw_N(A) = \prod_{i \in \mathcal{A}} v'_i(A) = \prod_{i \in \mathcal{A}} 2^{v_i(A)} = 2^{\sum_{i \in \mathcal{A}} v_i(A)} \geq 2^K.$$

Similarly if $sw_N(A) \geq 2^K$, then $sw_U(A) \geq K$. □

So if the preferences of agents are represented in bundle enumeration form, any complexity result that is valid for the problem of optimizing utilitarian social welfare is also valid for optimizing Nash social welfare up to a polynomial factor. (Also note that a similar reduction is possible from the NWO problem to the UWO problem by changing the utilities of the Nash problem and the threshold K to their logarithms, so the two problems are equivalent in terms of complexity, up to a polynomial factor). So since, as we have already mentioned, UWO with bundle form utilities is NP-complete:

5.3.2. Corollary. *The NWO problem is NP-complete when the agents' utilities are represented in bundle form.*

Proof: The problem is in NP. Given an instance of the problem, it is possible to compute its Nash social welfare in polynomial time (in the size of the input) since it only involves finding the utility of the corresponding bundle in each agent's list of utilities, and multiplying them. This takes no more than a linear number of steps. Once this computation is carried out, the result should simply be compared to the given threshold.

The NWO problem with the utilities in bundle form is also NP-hard, following the previous theorem and the fact that UWO is known to be NP-complete. It follows that the problem is NP-complete. □

Next we discuss the complexity of the Nash WDP when preferences are represented in k-additive form. The technique used in the previous theorem will not work for this case, because in the k-additive form, the coefficients of the agents' preferences are first summed up, and then the product is computed as the Nash CUF.

The following problem, the *subset product (SP)* problem, which is less well-known but rather similar to the *subset sum* problem will prove useful in our endeavor.

Subset Product (SP)	
Instance:	$S = \{s_1, \cdots, s_r\}$, $size : S \to \mathbb{N}$, and $K \in \mathbb{N}$
Question:	$\exists S' \subseteq S : \prod_{s_i \in S'} size(s_i) = K$?

SP is NP-complete [24]. It is also genuinely a decision problem, unlike the optimization problems that we have rendered into decision problems above. In order to be able to make use of this problem, we define anther version of the Nash WDP problem that involves finding an allocation with an *exact* amount of Nash social welfare:

Exact Nash Welfare (ENW)	
Instance:	$\langle \mathcal{A}, \mathcal{R}, \mathcal{V} \rangle$, $K \in \mathbb{Q}$
Question:	$\exists$ allocation A : $sw_N(A) = K$?

Now we are ready for the next result.

5.3.3. Theorem. *There is a polynomial time reduction from Subset Product to ENW where the agents' utilities are represented in k-additive form.*

Proof: Suppose we have an instance $S = \{s_1, \cdots, s_r\}$, $size : S \to \mathbb{N}$, and K of the SP problem. We reduce it to an instance $\langle \mathcal{A}, \mathcal{R}, \mathcal{V} \rangle$, K of the ENW problem, such that $\mathcal{A} = \{1, \cdots, r\}$, $\mathcal{R} = \{s_1, \cdots, s_r\}$, and v_i is defined in the k-additive form in the following way

$$\alpha_i^T = \begin{cases} size(s_i) - 1 & \text{if } T = \{s_i\} \\ 1 & \text{if } T = \emptyset. \end{cases}$$

So the agent i gives utility $size(s_i)$ to any bundle containing the resource s_i, and a utility of 1 to any other bundle. Note that the bundles for which the coefficient is zero need not be specified in the k-additive form. This reduction is obviously possible in time polynomial (and in fact linear) in the input size, as the utility function of each agent only consists of two values in the k-additive form and the number of agents and resources are the same as the elements of the input set.

Now if there is an $S^* \subseteq S$ such that the product of the sizes of the elements of S^* are exactly equal to K, then allocation A with the following properties has Nash CUF of exactly K. For each $s_j \in S^*$, let $s_j \in A(j)$, and for any $s_k \notin S^*$ choose some $l \neq k$ and set $s_k \in A(l)$.

Conversely, suppose that there is an allocation A with $sw_N(A) = K$, then there is corresponding subset $S^* \subseteq S$ such that $\prod_{s_i \in S'} size(s_i) = K$ defined in the following way: $s_i \in S^* \iff s_i \in A(i)$. □

5.3.4. Corollary. *The ENW problem is NP-complete when the agents' utilities are represented in k-additive form. This holds even when $k = 1$.*

Proof: NWO with utilities in k-additive form is in NP, since given an instance of the problem it is possible to compute the Nash social welfare of the given allocation in a number of steps that is a linear function in the size of the input.

Subset product is known to be NP-complete [24], so based on Theorem 5.3.3, the ENW problem is also NP-complete since a polynomial time reduction exists from SP to ENW with k-additive utility functions, and it is in NP.

Since in the proof of Theorem 5.3.3 we only need to define non-zero coefficients for bundles of at most one item, the result also holds for 1-additive functions. □

The last result is particularly interesting since it shows that ENW is not only NP-complete for general k-additive functions, but even in the much more restricted case of 1-additive utilities. Nevertheless, the complexity result for ENW does not immediately imply a result on the complexity of the NWO problem for k-additive representations. This arises from the fact that ENW is not an optimization problem. It does however give us some insight into the nature of the problem, showing the fact that figuring out whether there there exists an allocation with a particular Nash social welfare value is a hard problem in itself.

Chapter 6

Combinatorial Auctions

Multiagent resource allocation can also be studied in a centralized context, as we have mentioned in the previous chapters. In this case, instead of leaving agents to negotiate between themselves, an authority can decide on the optimal allocation, given the utility functions of the agents. The optimization problem in this case is generally the same, but the approach is different and generally known as *combinatorial auctions (CA)* [10]. The problem of finding an optimal solution for this problem, in the sense of utilitarian social welfare, is referred to in the CA literature as the *winner determination problem (WDP).* As in Section 5, we will call the similar problem of finding an allocation with maximum Nash social welfare in a combinatorial auction the *Nash winner determination problem (Nash WDP)*, and will study it in the rest of this chapter.

WDP is generally an NP-complete problem, as we have seen in Section 5. We have also shown there that the Nash WDP is at least as hard for some cases, and it can be conjectured that it is probably as hard for other representations as well. Thus there is no efficient (i.e. polynomial-time) algorithm for effectively solving all instances of these problems. To overcome this obstacle *heuristic* algorithms can be devised. These are algorithms that search the extremely large space of possible solutions (here allocations) by rules of thumb that guarantee to find an optimal solution and generally do so in less time than a thorough search (at least for instances that are not too large). The performance of a heuristic algorithm usually cannot be theoretically assessed, but is judged by means of extensive experimentation. Another way of overcoming the curse of NP-copmleteness is using *approximation algorithms.* These algorithms can compute solutions that are provably close to the optimal solution, but not the exact solution.

Various heuristic methods and approximation algorithms have been developed for solving the WDP [51, 23]. In this chapter a heuristic method is presented for solving the Nash WDP when agents use a particular logic-based bidding language to represent their preferences. Our approach generally follows [60, 51, 23].

The basic idea is a *branch-and-bound (B&B)* algorithm where at each step the maximum that each agent can achieve without what has already been assigned to other agents is computed, and the product is used as an admissible heuristic to bound the state space tree.

6.1 Positive Cubes

The language we use in this section for preference representation is a subset of the language of *weighted propositional formulas (WPF)* introduced in Section 2.2.3 [35]. We present these languages and the particular the one we are going to use in this chapter more formally here.

Suppose we have a general MARA problem $\langle \mathcal{A}, \mathcal{R}, \mathcal{V} \rangle$ that is defined as in Chapter 4. The utiltiy functions can be defined as weighted propositional formulas in the following way.

6.1.1. Definition. Suppose that there is a propositional formula p_a for every resource $a \in \mathcal{R}$. A utility function u expressed in the language of *weighted propositional formulas (WPF)* as a set of pairs of propositional formulas and weights $G = \{(\phi_i, w_i)\}_i$ called a *goal base* (the formulas are satisfiable and distinct). In any state of the system, the propositional variable corresponding to an item is true if it is held by the individual computing the utility function. Thus each state can be considered as a model M_A based on the bundle A owned by the agent.

The utility of the agent for A is thus computed as:

$$u(A) = \sum \{w_i \mid (\phi_i, w_i) \in G \ \& \ M_A \models \phi_i\}.$$

This means that the sum of the weights corresponding to the formulas satisfied by the items assigned to the agent is considered as the utility in each allocation. This is the most common way to use logic based languages, but there are variations on how the weights should be aggregated, for instance in [59] the *maximum* function is used as an aggregator, i.e. the utility is the maximum of the weights of the satisfied formulas.

Apart from variations on the aggregation method, the language of WPFs can be restricted in two different ways, structural constraints on the formulas, and restrictions on the weights. Any combination of these restrictions lead to a different language. Here we restrict the formulas to *positive cubes*, i.e. conjunctions of the propositional variables (negative literals are not allowed, hence the term *positive*). We will call this language *PPCubes*. This language has also been used in [60], and is similar to that introduced in [30] in terms of expressiveness.

The PPCubes language is similar to the k-additive representation language restricted to positive coefficients. Each cube corresponds to a bundle of resources and the weights correspond to the coefficients. The difference is that the k-additive language limits the number of items in a bundle in the representation to k (this would be equivalent to restricting the length of the cubes to k in the

PPCubes representation). We know that the k-additive representation (without a restriction on k) is fully expressive, so is the language of positive cubes directly corresponding to it. However PPCubes is not fully expressive since the lack of negative weights makes it unable to express non-monotonic utility functions. To be exact, it can express the class of all nonnegative supermodular monotone utility functions satisfying the following constraint $\sum_{Y \subseteq X}(-1)^{|X \setminus Y|} \cdot u(Y) \geq 0$ for all subsets X of the propositional variables [58].

The following example illustrates a concrete case of how utilities are computed using this language in a MARA setting.

6.1.2. EXAMPLE. Let $\mathcal{A} = \{1, 2, 3\}$ and $\mathcal{R} = \{a, b, c, d, e\}$. The agents' preferences are represented in the PPCubes language by the following goal bases:

i	G_i
1	$(p_a,\ 4)\ ,(p_b,\ 3),\ (p_a \wedge p_b,\ 2),\ (p_a \wedge p_e,\ 2),\ (p_a \wedge p_b \wedge p_c,\ 5)$
2	$(p_b,\ 2)\ ,(p_c,\ 2),\ (p_d \wedge p_b,\ 4),\ (p_c \wedge p_e,\ 1),\ (p_b \wedge p_c \wedge p_e,\ 3)$
3	$(p_b,\ 1)\ ,(p_c,\ 3),\ (p_d,\ 4),\ (p_b \wedge p_d,\ 3),\ (p_a \wedge p_c \wedge p_e,\ 5),\ (p_a \wedge p_b \wedge p_d,\ 2)$

Now if allocation A is defined such that $A(1) = \{a,\ e\}$, $A(2) = \{c\}$, and $A(1) = \{b,\ d\}$, then the utilities are computed by adding up the weights of the formulas that are satisfied by the agents bundle in A. Thus $u_1(A) = 4 + 2 = 6$, $u_2(A) = 2$, and $u_3(A) = 1 + 4 + 3 = 8$. The Nash CUF of this allocation is $sw_N(A) = 6 \times 2 \times 8 = 96$.

6.2 The Heuristic Algorithm

The algorithm that we present here solves the Nash WDP. Given an instance $\langle \mathcal{A}, \mathcal{R}, \mathcal{V} \rangle$ of the Nash WDP, the heuristic algorithm computes an allocation maximal in Nash social welfare.

We construct a search tree, each of its nodes a *partial allocation* and the leaves complete allocations. A partial allocation is similar to an allocation, except that some of the items may not be assigned to any agent (if A is a partial allocation, then there may be a resource r such that $\forall i \in \mathcal{A}, r \notin A(i)$). The root of the tree is the partial allocation in which no resources have been assigned yet. Each child of a node corresponds to the allocation that is the result of assigning one more item to one of the agents. Thus each node has $|\mathcal{A}| = n$ children, and the depth of the tree is $|\mathcal{R}| = m$, resulting in n^m leaves, and $\frac{n^{m+1}-1}{n-1}$ internal nodes. This is a huge number even for rather small numbers of agents and resources, e.g. for 5 agents and 8 resources, there are $390,625$ allocations and $488,281$ nodes in total.

In order to be able traverse the search tree for finding an optimal allocation, we use a branch-and-bound algorithm with a heuristic that shall be explained shortly. A B&B algorithm uses a heuristic estimation for the optimal allocation to *prune* branches of the tree that will provably not lead to an optimal allocation. A heuristic estimation is a measure that estimates the maximum amount of utility that can possibly be achieved by continuing a branch of the search tree (completing a partial allocation) in any way. A heuristic estimation is *admissible* if it always predicts a value at least as high as the actual value attainable for any state.

So if an admissible heuristic estimate is used, a B&B algorithm that prunes any branch that has an estimated value lower than the value already achieved by another state in the search tree would not prune out any possible optimal allocation. Intuitively this procedure puts a partial allocation aside if it can be shown that it is not promising early on. Therefore depending on how close the heuristic estimation is to the actual achieved value (how *tight* it is), a number of nodes can be eliminated from the search tree. Obviously, the tighter the heuristic estimation, the better.

We must extend the notions of satisfiability to partial allocations in order to use it in the heuristic B&B algorithm. A formula is satisfiable for agent i in partial allocation A if the item corresponding to any propositional variable in the formula is either assigned to i, $i \in A(i)$, or not allocated yet. So a formula is not satisfiable for i if the resource corresponding to at least one of its variables is assigned to another agent. And as already stated, a formula is satisfied by A for i whenever all items corresponding to its variables are owned by i.

Suppose we are at a state corresponding to the partial allocation A. Then the current value of the state for any agent i, $g_i(A)$, is the sum of the weights of all of his *satisfied* formulas. The achievable value of the state for any agent i, $h_i(A)$, would then be the sum of the weights of all of his *satisfiable* formulas. Furthermore, the current value for each state, $g(A)$, is the product of the current values of the individual agents in that state, and the estimated heuristic, $h(A)$, is the product of their achievable values.

In some notations h may denote the amount that is attainable for a node in addition to the value it already has, but here it is the total attainable value.

More formally, at any partial allocation A, the current Nash value and the estimated heuristic for the Nash value, represented as g and h respectively, are computed as follows (the utility for a partial allocation is computed similar to that of a complete one, by summing the weights for the satisfied formulas for each agent):

$$g(A) = \prod_{i \in \mathcal{A}} u_i(A),$$

$$h(A) = \prod_{i \in \mathcal{A}} h_i(A),$$

such that

$$h_i(A) = \sum \{w_j \mid (\phi_j, w_j) \in G \ \& \ M_A \models \phi_j \text{ or } \forall p \in \phi_j,\ M_A \not\models p \rightarrow \bot\}$$

It is easy to see that this heuristic is admissible:

6.2.1. Theorem. *The h function defined above is an admissible heuristic for the Nash WDP problem when the preferences are represented in the PPCubes language.*

Proof: In order to prove the result we must show that the heuristic function for any partial allocation A estimates a value that is at least as large as the Nash CUF value for any of the allocations that may arise from A by allocating the unallocated resources. To see this, observe that for each agent i, $h_i(A)$ is equal to the value that agent i will assign to an allocation that gives all of the unallocated items to himself. Since the utility functions are monotonic, $h_i(A)$ is a (tight) upper bound on what i can get in any complete allocation arising from A. Since this is true for all of the agents, h is an admissible heuristic. □

So our proposed B&B algorithm works as follows. It starts from the root (empty partial allocation), and an empty frontier of nodes. The threshold for pruning is set to the the g of the root node. At each step, it chooses an item that is not allocated at the current node to allocate (we will discuss the choice of this item in the Section 6.4) and computes the g and h functions for the children of the node that arise from allocating the chosen item to different agents. If a node has a heuristic value that is estimated to be less than the pruning threshold, it is pruned. If it has a g value that is larger than the current pruning threshold, the threshold is updated. Finally the expanded node is taken off of the frontier and its (unpruned) children are added in its place. The next node to be expanded would then be the one on the frontier with highest estimated value. Whenever a complete allocation is reached, it is chosen as the current node, we are done and it must be an optimal allocation since the heuristic is admissible.

6.2.2. Example. Suppose $\mathcal{A} = \{1, 2\}$ and $\mathcal{R} = \{a, b, c\}$, and the utility functions are defined in the PPCubes form in the following table:

i	G_i
1	$(\top,\ 1)\ , (p_a,\ 4)\ , (p_a \wedge p_b,\ 1)\ , (p_b \wedge p_c,\ 2)\ , (p_a \wedge p_b \wedge p_c,\ 2)$
2	$(p_b,\ 2)\ , (p_a \wedge p_b,\ 4)\ , (p_a \wedge p_c,\ 3)\ , (p_b \wedge p_c,\ 3)$

For the partial allocation A_0 where no items are assigned yet we would have $g_1(A_0) = 1,\ g_2(A_0) = 0$ since the first agent assigns utility 1 to the empty bundle, thus $g(A_0) = 0$. The heuristic values are $h_1(A_0) = 1+4+1+2+2 = 10,\ h_2(A_0) = 2+4+3+3 = 12$, and so the estimated value in this step is $h(A_0) = 10 \times 12 = 120$.

Now let's suppose A_1 is the partial allocation in which a is given to 1, and b to 2. We have $g_1(A_1) = 1+4 = 5,\ g_2(A_1) = 2$, and $h_1(A_1) = 1+4 = 5,\ h_2(A_1) = 2+3 = 5$, and so $h(A_1) = 5 \times 5 = 25$. This also happens to be the Nash CUF value for the optimal Nash allocation in this problem, which is achieved by allocating c to 2.

6.3 Generating Random Problems

A fundamental issue in experimenting and assessing the effectiveness of algorithms generated for combinatorial auctions has been finding test cases suitable for experimentation. The aim is to generate test cases consisting of resources, agents, and their preferences such that they are random in the sense that a particular algorithm may not have an advantage when running them, and also are a reasonable representation of the kind of problems that the algorithms would face in actual applications. This problem may not seem as hard as it actually is in first sight, but it is extremely difficult. This is because of the fundamental complexity and combinatorial nature of an instance of the problem, and the hugeness of a search space that must be rendered in finding an optimal solution. There is no obvious way to generate completely random instances, even if such instances are generated, they may be much harder than that expected from the algorithm in *real* situations. It is thus very easy to end up with cases that are too hard or too easy.

The *CATS (combinatorial auctions test suite)* has been proposed for overcoming this obstacle [38]. It is a software exclusively designed to generate test cases for combinatorial auctions, for a variety of different applications. CATS is widely regarded as a benchmark for experimenting algorithms in CA, and thus provides a reasonable measure for comparing them. The idea behind it is to consider reasonable random instances of real-life cases where combinatorial auctions are used, e.g. allocating routes to vehicles in a logistics problem, and computing the parameters of the CA problem based on the preferences that agents may typically have in such cases.

Nevertheless, unfortunately we can not make use of the CATS software in our experiments, since it does not support logic-based languages, but languages that are more common in CA such as the OR and XOR languages. So we have devised a simple method for generating instances for experimentation.

The basic principle we take into account is that since we are using a representation that is rather similar to the k-additive form, and as we have discussed previously in Section 2.2.3, usually agents would generally assign lower values to longer cubes particularly if some of the items in them are also contained in other (particularly smaller) cubes. This is reasonable since the value of each cube is actually the utility that an agent assigns to *all* of the items in it *in addition to* the that of its individual resources.

Another principle that also makes sense in many actual applications is that the number of items in a cube would not typically be very high. This follows from the same fact that makes k-additive representations useful: agents would in many cases be able to express their preferences with a k-additive function with a small k.

So we apply two general rules when generating instances. First that the probability of generating a cube of a particular length has an inverse relationship with its length, the longer the length of the cube, the lower the probability of generating it. Second, the value of a cube of higher length is generally lower than that of a cube with a lower length, or better yet, a cube that has a subset that is also a cube would probably get a lower value.

We use a simple method that takes these principles into account to some extent for generating the test cases. The method takes the number of agents, n, and the number of items, m, as input and generates each agent's goal base as follows. The agent is assigned a random number $k \in [1, 2m]$ as the number of pairs in its goal base.

Half of each agent's goal base formulas are generated so that each of propositional formulas has a 20% chance of containing each of the variables. This will make the average length of these formulas equal to one fifth of the total number of resources, and reduces the probability of having very long formulas. On the other hand it also results in fewer formulas whose lengths are much smaller than a fifth of the resources. Since, as stated before, having more of smaller formulas is more desirable, the other half of the formulas of these goal bases each contain a single variable, these are also determined randomly.

For the weights, each agent assigns a random integer between 1 and 10 to each of the goods as its estimated value, these numbers are used in the computation of the bundles' weights. After generating the goal base, each of the formulas that does not contain any of the agent's other formulas (does not contain all of the variables in the other formula) is assigned a value that is equal to the sum of the values of the goods corresponding to its variables with 30% noise. If a formula does have subsets, the sum of the weights of its subsets is subtracted from this computed weight (if this sum is more than the computed weight, the final weight is set to 1). This results in a utility function that is not additive but somewhat realistic in the sense that the weights have a relation with some sort of values on the items, the values of subsets are not counted multiple times, thus the values of longer formulas is generally lower if the items in them have already occurred in other formulas.

6.4 Experiments

We have implemented the presented heuristic algorithm[1] and ran some experiments using cases generated by the method in the previous section.

The implementation is a Java program and the experiments have been run on a Fedora Linux operating system on an Intel Pentium 4 with 3.00GHz CPU and 1 GB of RAM.

It should be noted here that we have used the *best-estimate first* branching policy of [60] in the following experiments. This means that in each step the next

[1]The framework of the source code used to implement the algorithm was written by Joel Uckelman, and is the same as that used to run the experiments in [60].

m	time h	time BF	nodes h	nodes BF
5	2	2	301	3,906
6	3	5	659	19,531
7	6	23	2,080	97,656
8	9	116	3,184	488,281
9	29	718	8,108	2,441,406
10	58	3,736	15,200	12,207,031
11	141	22,021	32,887	61,035,156
12	543	-	135,475	305,175,781
13	654	-	163,310	1,525,878,906
14	1,330	-	255,325	7,629,394,531
15	3,406	-	592,677	38,146,972,656
16	6,704	-	1,012,522	190,734,863,281
17	10,707	-	1,496,180	953,674,316,406

Table 6.1: Comparison of running times and number of nodes generated by the heuristic h and brute force search. Consists of cases where the number of agents are 5 and the number of resources are between 5 and 17. Each entry is the average of the results for runs on fifty randomly generated cases, both algorithms have been run on the same cases. The times are CPU time and each unit represents 0.01 seconds. Brute force search was unable to complete execution for the larger cases with the available resources, for which the entry is left empty.

good to be allocated is one of the goods that has the best estimate according to the heuristic presented in that article. In short each agent computes an estimate for the value that each unallocated resource has for it, and the estimate of the value of each good is equal to the maximum of all estimates. The agents computes the estimates by adding up for each formula in their goal base that is still satisfiable and contains the resource in consideration, the value of the goal base divided by the number of resources that still need to be acquired to satisfy the formula. Using this estimate has better results with our heuristic than selecting the first unallocated resource lexicographically.

In order to assess the efficiency of the algorithm we have compared the running time and number of nodes created in the heuristic algorithm to a brute force search that traverses all nodes of the search tree. The experiments have each been run on randomly generated instances that all have 5 agents and the number of resources varies between 5 and 17. Each experiment has been run 50 times. The results are summarized in Table 6.4.

It can be seen here that the heuristic algorithm outperforms brute force search by a large margin. It is in fact so as we shall see shortly.

Unlike the brute force algorithm, the run times and number of nodes generated by the heuristic for the Nash WDP has a very high variance, for instance for 17 items and 5 agents the number of nodes is between 3,586 and 9,661,551, the second being almost 3000 times the first. The size of the minimum and maximum cases in comparison to the average can be seen in Figure 6.1. This shows that (at least with the random cases that we generate here) problems of the same size may be of drastically different levels of hardness for our heuristic algorithm. However there is no particular pattern between the size of this ratio and the size of the problem, it depends on the particular cases that happened to be generated.

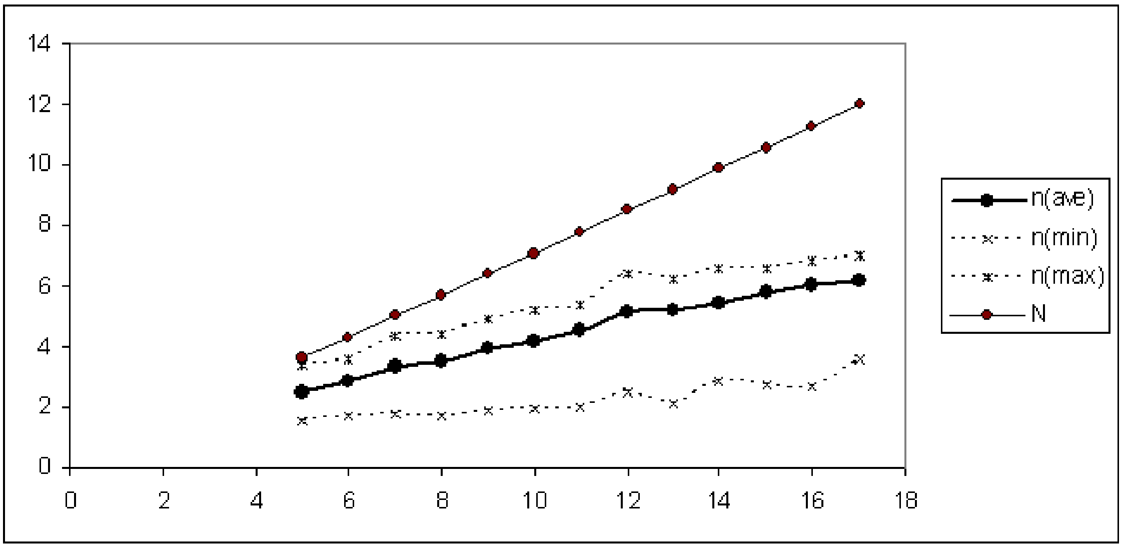

Figure 6.1: The comparison of the number of nodes generated by the heuristic method and brute force search. The horizontal axis is the number of resources, the vertical axis is the logarithm in base 10 of the number of nodes of the tree generated. N is the number of total nodes of the search tree. n(ave), n(min), and n(max) are the average, minimum and maximum of the number of modes generated in the experiments respectively.

By closely observing Table 6.4, it is apparent that although the heuristic works much better than brute force search, it seems to increase rather quickly itself. In fact in the results for these experiments it can be observed that the run time of the heuristic algorithm (which has a linear relation with the number of nodes) roughly doubles with the addition of each extra item. This can be seen more precisely by computing the 12th root of the ratio of the run times for $m = 17$ and $m = 5$ which is exactly 2.03. Thus the performance of this algorithm also increases exponentially, but with a factor of 2 in contrast to the factor of 5 with which the search tree and thus brute force search increases.

This implies that although the percentage of nodes visited by the heuristic algorithm decreases on average in each step, it still increases exponentially in time.

Because of the high variance of the performance, some instances of a particular size may be solved very efficiently by the heuristic, while others will not be effectively solvable. Thus the average performance for large instances may be extremely hard or impossible to compute (the computable size of course depends on the available computational resources). Furthermore the average performance does not say much about what may be expected because of the high fluctuation of the results.

In conclusion, it would generally be desirable to find a heuristic for this version of the Nash WDP that has better performance than this one. It seems plausible that an approach similar to the estimate used in [60] that we have briefly sketched above might be possible. Dividing the value of formulas by the number of variables that are yet to be satisfied for them results in a tighter heuristic in that case.

We have not been able to use a similar technique for the Nash WDP using PPCubes because computing the Nash CUF involves adding up the values for all formulas for each agent first, and then computing the product of all of these values. This does not allow us to for instance assign a separate value to each resource by computing the root or a similar method, while it is possible for the classical WDP since all steps involve summation. However, it may be possible to find a method inspired by this technique upon further research.

Chapter 7

Conclusion

This thesis concentrated on the study of Nash social welfare in MARA. We have investigated various problems in this regard. The theoretical results have mostly focused on distributed negotiation.

We covered negotiation processes on the level where the unit of negotiation is a single deal, and not details of protocols leading to a deal. We have devised the concept of a Nash deal which is a local rationality criteria and have shown that all negotiation sequences using these deals will converge to a Nash optimal allocation. We have also shown that a deal with any level of complexity may be required in such a sequence as long as it is not individually decomposable. We have further shown that a complex deal involving all agents may be needed in a sequence of Nash deals as long as the number of resources exceeds the number of agents even if the utility functions of all agents are modular.

We have also studied the computational complexity of computing the Nash CUF and an optimal Nash allocation. We have proven that the problem of optimizing Nash social welfare is NP-complete when utilities are represented in the bundle form by a reduction from the classical (utilitarian) winner determination problem. We have studied the problem of deciding whether there exists an allocation with a particular Nash social welfare and have proven this problem to be NP-complete when the utility functions are represented in k-additive form, even if they are all 1-additive.

Finally we have studied the Nash WDP and presented a heuristic branch and bound algorithm for computing the optimal Nash allocation in a combinatorial auction. The heuristic is proven to be admissible. An implementation of the algorithm has been experimented which shows that the algorithm succeeds in reducing the search space substantially. However it also shows that it is not a very tight heuristic, and would need improvement to be able to solve larger instances of the problem.

7.1 Future Work

Although we have mostly focused on optimizing Nash social welfare in a distributed MARA problem where the trading is carried out via negotiation, we have not studied this in a level more detailed than Nash deals. It would particularly be interesting to study procedures that agents may use in order to compute a Nash deal. Algorithms can be designed for such procedures, their complexity can be studied, et cetera. There could also be a focus on designing heuristic (or even theoretically provable) algorithms for choosing Nash deals that result in faster convergence of the sequence of deals, or less reallocation of items between agents in the process, or satisfy various other criteria. These problems can be pursued both in theory and experimentation.

Another line of research not far from those mentioned above could be on the communication of the agents that want to agree on a Nash deal. Various existing or novel protocols may be executed in this context. This would particularly be useful in the implementation of independent software agents that would want to carry out such procedures.

The complexity of the Nash WDP can also be studied for other representation languages. Particularly the result involving the exact Nash problem does not directly imply NP-completeness for the optimization version of the problem in the k-additive case, and a result on the complexity of that problem would be useful (it is most probably also NP-complete, finding an NP-complete problem that would naturally lend to a reduction would however be intriguing). The same goes for finding results for other notable representations.

The chapter on the heuristic algorithm for solving the Nash WDP is actually in its first stages and would need much more work to be fully developed. The heuristic presented here is not very tight, as we have pointed out already, and it may be possible to improve it in further steps.

There should also be some investigations on the possible application of all of the problems addressed in this work. Real-life frameworks in which implementing social welfare is useful and beneficial should be identified so that the theoretical efforts are not in vain. In this line, the converse approach known as *welfare engineering* [15] is also worth noting, there should be a general effort for finding, or if needed tailoring, the right notion of social welfare for any given problem at hand. Then, studies such as this dissertation can provide us with the right theoretical tools whenever they may be needed.

Bibliography

[1] D. E. Bell, H. Raiffa, and A. Tversky, editors. *Decision making: Descriptive, normative, and prescriptive interactions.* Cambridge University Press, New York, 1988.

[2] S. Bouveret and J. Lang. Efficiency and envy-freeness in fair division of indivisible goods: Logical representation and complexity. *Journal of Artificial Intelligence Research*, 32:525–564, 2008.

[3] E. Cantillon and M. Pesendorfer. Auctioning bus routes: The london experience. In P. Cramton et al., editor, *Combinatorial Auctions.* MIT Press, 2006.

[4] Y. Chevaleyre, P. E. Dunne, U. Endriss, J. Lang, M. Lemaître, N. Maudet, J. Padget, S. Phelps, J. A. Rodríguez-Aguilar, and P. Sousa. Issues in multiagent resource allocation. *Informatica*, 30:3–31, 2006. Survey paper.

[5] Y. Chevaleyre, U. Endriss, S. Estivie, and N. Maudet. Multiagent resource allocation with k-additive utility functions. In D. Bouyssou, M. Janowitz, F. Roberts, and A. Tsoukiàs, editors, *Proceedings of the DIMACS-LAMSADE Workshop on Computer Science and Decision Theory*, volume 3 of *Annales du LAMSADE*, pages 83–100, October 2004.

[6] Y. Chevaleyre, U. Endriss, S. Estivie, and N. Maudet. Reaching envy-free states in distributed negotiation settings. In M. Veloso, editor, *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-2007)*, pages 1239–1244, Hyderabad, India, January 2007. Poster Paper.

[7] Y. Chevaleyre, U. Endriss, S. Estivie, and N. Maudet. Multiagent resource allocation in k-additive domains: Preference representation and complexity. *Annals of Operations Research*, 163(1):49–62, 2008.

[8] Y. Chevaleyre, U. Endriss, and N. Maudet. On maximal classes of utility functions for efficient one-to-one negotiation. In *Proc.19th International Joint Conference on Artificial Intelligence (IJCAI-2005)*, pages 941–946. Morgan Kaufmann, 2005.

[9] V. Conitzer, T. W. Sandholm, and P. Santi. Combinatorial auctions with k-wise dependent valuations. In *Proc. 20th National Conference on Artificial Intelligence (AAAI05)*, pages 248–254. AAAI Press, 2005.

[10] P. Cramton, Y. Shoham, and R. Steinberg, editors. *Combinatorial Auctions*. MIT Press, 2006.

[11] C. D'Aspremont and L. Gevers. Equity and the informational basis of collective choice. *Review of Economic Studies*, 44(2):199–209, 1977.

[12] G. Debreu. Topological methods in cardinal utility theory. In Karlin Arrow and Suppes, editors, *Mathematical Methods in the Social Sciences*. Stanford University Press, 1960.

[13] P. E. Dunne. Extremal behavior in multiagent contract negotiation. *J. of Artificial Intelligence Research*, 23:41–78, 2005.

[14] P. E. Dunne, M. Wooldridge, and M. Laurence. The complexity of contract negotiation. *Artificial Intelligence*, 164:23–46, 2005.

[15] U. Endriss and N. Maudet. Welfare engineering in multiagent systems. In A. Omicini, P. Petta, and J. Pitt, editors, *Engineering Societies in the Agents World IV*, volume 3071 of *LNAI*, pages 93–106. Springer-Verlag, 2004. Post-proceedings of ESAW-2003.

[16] 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.

[17] U. Endriss, N. Maudet, F. Sadri, and F. Toni. On optimal outcomes of negotiations over resources. In J. S. Rosenschein, T. Sandholm, M. Wooldridge, and M. Yokoo, editors, *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2003)*, pages 177–184. ACM Press, July 2003.

[18] U. Endriss, N. Maudet, F. Sadri, and F. Toni. Negotiating socially optimal allocations of resources. *Journal of Artificial Intelligence Research*, 25:315–348, 2006.

[19] A. M. Feldman. Bilateral trading processes, pairwise optimality, and pareto optimality. *The Review of Economic Studies*, 40(4):463–473, 1973.

[20] R. Feldmann, M. Gairing, T. Lücking, B. Monien, and M. Rode. Selfish routing in non-cooperative networks: A survey. In *Proc. 28th International Symposium on Mathematical Foundations of Computer Scienc (MFCS-2003)*, pages 21–45. Springer-Verlag, 2003.

[21] P. Fishburn. Condorcet social choice functions. *SIAM Journal of Applied Mathematics*, 33:469–489, 1977.

[22] P. C. Fishburn. *Utility Theory for Decision Making*. John Wiley and Sons, 1970.

[23] 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*, 1999.

[24] Michael Garey and David Johnson. *Computers and Intractability - A Guide to the Theory of NP-completeness*. Freeman, 1979.

[25] W. M. Gorman. The structure of utility functions. *Review of Economic Studies*, 35:367–390, 1968.

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

[27] P. Gradwell and J. Padget. Distributed combinatorial resource scheduling. In *Proc. AAMAS Workshop on Smart Grid Technologies (SGT-2005)*, 2005.

[28] M. Hajiaghayi, R. Kleinberg, and T. Sandholm. Automated online mechanism design and prophet inequalities. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2007.

[29] J. C. Harsanyi. Equity and the informational basis of collective choice. *J. of Political Economy*, 63:309–321, 1955.

[30] H. H. Hoos and C. Boutilier. Solving combinatorial auctions using stochastic local search. In *Proceedings of the 17th National Conference on Artificial Intelligence*, 2000.

[31] Brahim Chaib-draaa Houssein Benameura and Peter Kropf. Multi-item auctions for automatic negotiation. *Information and Software Technology*, 44(5):291–301, 2002.

[32] G. M. Jonker, J.-J. Meyer, and F. Dignum. Market mechanisms in airport traffic control. In *Proc. 2nd European Workshop on Multiagent Systems (EUMAS-2004)*, 2004.

[33] R.L. Keeney and H. Raiffa. *Decisions with multiple objectives: Preferences and value tradeoffs.* Cambridge University Press, New York, 1993.

[34] S. C. Kolm. *Justice et equité.* CNRS, Paris, 1972.

[35] J. Lang. Logical preference representation and combinatorial vote. *Annals of Mathematics and Artificial Intelligence*, 42(1-3):37–71, 2004.

[36] M. Lemaître, G. Verfaillie, and N. Bataille. Exploiting a common property resource under a fairness constraint: A case study. In *Proc. 16th International Joint Conference on Artificial Intel ligence (IJCAI-1999)*, pages 206–211. Morgan Kaufmann, 1999.

[37] M. Lemaître, G. Verfaillie, H. Fargier, J. Lang, N. Bataille, and J.-M. Lachiver. Equitable allocation of earth observing satellites resources. In *Proc. 5th ONERA-DLR Aerospace Symposium (ODAS-2003)*, 2003.

[38] K. Leyton-Brown, M. Pearson, and Y. Shoham. Towards a universal test suite for combinatorial auction algorithms. In *Proceedings of the 2nd ACM Conference on Electronic Commerce*, 2000.

[39] R. J. Lipton, E. Markakis, E. Mossel, and A. Saberi. On approximately fair allocations of indivisible goods. In *EC '04: Proceedings of the 5th ACM conference on Electronic commerce*, pages 125–131. ACM, May 2004.

[40] B. López, S. Suárez, and J. L. de la Rosa. Task allocation in rescue operations using combinatorial auctions. In *Proc. 6th Catalan Congress on Artificial Intel ligence (CCIA-2003).* IOS Press, 2003.

[41] S. S. Manvi, M. N. Birj, and B. Prasad. An agent-based resource allocation model for computational grids. *Multiagent and Grid Systems*, 1:17–27, 2005.

[42] H. Moulin. *Axioms of Cooperative Decision Making.* Econometric Society Monographs. Cambridge University Press, Cambridge, 1988.

[43] H. Moulin. *Fair Division and Collective Welfare.* MIT Press, Cambridge, Massachusetts, 2003.

[44] John Nash. The bargaining problem. *Econometrica*, 35:155–162, 1950.

[45] N. Nisan. Bidding languages for combinatorial auctions. In P. Cramton et al., editor, *Combinatorial Auctions.* MIT Press, 2006.

[46] C. H. Papadmitiou. *Computational complexity.* Addison Wesley, 1994.

[47] H. Van Dyke Parunak. Applications of distributed artificial intelligence in industry. In G. M. P. OHare and N. R. Jennings, editors, *Foundations of Distributed Artificial Intelligence.* John Wiley and Sons, 1996.

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

[49] T. W. Sandholm. An implementation of the contract net protocol based on marginal cost calculations. In *Proc. 11th National Conference on Artificial Intelligence (AAAI-1993)*, pages 256–262. AAAI Press, 1993.

[50] T. W. Sandholm. Contract types for satisficing task allocation: I theoretical results. In *Proc. AAAI Spring Symposium: Satisficing Models*, 1998.

[51] T. W. Sandholm. Algorithms for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135(1-2):1–54, 2002.

[52] A. K. Sen. *Collective choice and social welfare.* Holden Day, San Francisco, 1970.

[53] A. K. Sen. On weights and measures: Informational constraints in social welfare analysis. *Econometrica*, 45(7):1539–1572, 1977.

[54] A. K. Sen. *Commodities and Capabilities.* North-Holland, Amsterdam, 1985.

[55] R. G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113, 1980.

[56] P. Sousa, C. Ramos, and J. Neves. The fabricare scheduling prototype suite: Agent interaction and knowledge base. *Journal of Intelligent Manufacturing*, 14(5):441–455, 2003.

[57] P. J. t Hoen and J. A. La Poutré. A decommitment strategy in a competitive multiagent transportation setting. In *Agent-Mediated Electronic Commerce V*, pages 56–72. Springer-Verlag, 2004.

[58] J. Uckelman, Y. Chevaleyre, U. Endriss, and J. Lang. Representing utility functions via weighted goals. *unpublished.*

[59] J. Uckelman and U. Endriss. Preference modeling by weighted goals with max aggregation. In G. Brewka and J. Lang, editors, *Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR-2008)*, pages 579–587. AAAI Press, September 2008.

[60] J. Uckelman and U. Endriss. Winner determination in combinatorial auctions with logic-based bidding languages. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2008)*, pages 1617–1620. IFAAMAS, May 2008. Short Paper.

[61] H. van Brussel, J. Wyns, P. Valckenaers, L. Bongaerts, and P. Peeters. Reference architecture for holonic manufacturing systems: Prosa. *Computers in Industry*, 37(3):255–274, 1998.

[62] M. P. Wellman and P. R. Wurman. Market-aware agents for a multiagent world. *Robotics and Autonomous Systems*, 24:115–125, 1998.

[63] P. Winoto, G. McCalla, and J. Vassileva. An extended alternating-offers bargaining protocol for automated negotiation in multi-agent systems. In *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, Lecture Notes in Computer Science, pages 179–194, Berlin/Heidelberg, 2002. Springer.

[64] A. C.-C. Yao. Some complexity questions related to distributive computing (preliminary report). In *Proc. of the Annual ACM Symposium on Theory of Computing*, 1979.