Issues in Multiagent Resource Allocation

Yann Chevaleyre
LAMSADE, Université Paris-Dauphine (France)
Email: chevaley@lamsade.dauphine.fr

Paul E. Dunne
Department of Computer Science, University of Liverpool (UK)
Email: ped@csc.liv.ac.uk

Ulle Endriss
ILLC, Universiteit van Amsterdam (The Netherlands)
Email: ulle@illc.uva.nl

Jérôme Lang
IRIT, Université Paul Sabatier, Toulouse (France)
Email: lang@irit.fr

Michel Lemaître
ONERA, Centre de Toulouse (France)
Email: michel.lemaitre@cert.fr

Nicolas Maudet
LAMSADE, Université Paris-Dauphine (France)
Email: maudet@lamsade.dauphine.fr

Julian Padget
Department of Computer Science, University of Bath (UK)
Email: jap@cs.bath.ac.uk

Steve Phelps
Department of Computer Science, University of Liverpool (UK)
Email: sphelps@csc.liv.ac.uk

Juan A. Rodríguez-Aguilar
Artificial Intelligence Research Institute (IIIA-CSIC), Barcelona (Spain)
Email: jar@iiia.csic.es

Paulo Sousa
DEI, Instituto Superior de Engenharia do Porto (Portugal)
Email: psousa@dei.isep.ipp.pt

Keywords: resource allocation, negotiation, preferences, social welfare, complexity, simulation

The allocation of resources within a system of autonomous agents, that not only have preferences over alternative allocations of resources but also actively participate in computing an allocation, is an exciting area of research at the interface of Computer Science and Economics. This paper is a survey of some of the most salient issues in Multiagent Resource Allocation. In particular, we review various languages to represent the preferences of agents over alternative allocations of resources as well as different measures of social welfare to assess the overall quality of an allocation. We also discuss pertinent issues regarding allocation procedures and present important complexity results. Our presentation of theoretical issues is complemented by a discussion of software packages for the simulation of agent-based market places. We also introduce four major application areas for Multiagent Resource Allocation, namely industrial procurement, sharing of satellite resources, manufacturing control, and grid computing.

Contents

1 Introduction

The allocation of resources is a central matter of concern in both Computer Science and Economics. To emphasise the fact that resources are being distributed amongst several agents and that these agents may influence the choice of allocation, the field is sometimes called *Multiagent Resource Allocation* (MARA). The questions investigated by computer scientists are often of a procedural nature (*how* do we find an allocation?), while economists are more likely to concentrate on qualitative issues (*what* makes a good allocation?). A comprehensive analysis of the problem at hand, however, requires an interdisciplinary approach. Here the *multiagent system* (MAS) paradigm offers an excellent framework in which to study these issues.

MARA is relevant to a wide range of applications. These include, amongst others, industrial procurement [45], manufacturing and scheduling [15, 71, 89], network routing [38], the fair and efficient exploitation of Earth Observation Satellites [59, 60], airport traffic management [52], crisis management [62], logistics [49, 77], public transport [16], and the timely allocation of resources in grid architectures [48].

This paper is a survey of some of the most salient issues in MARA. In the remainder of this introduction, we first give a tentative definition of MARA and introduce its main parameters (Section 1.1). To illustrate the interdisciplinary character of the field, we then list some of the research questions that we consider particularly interesting and challenging (Section 1.2). Finally, we give an overview of the content of the main body of the paper (Section 1.3).

1.1 What is MARA?

A tentative definition would be the following:

> *Multiagent Resource Allocation is the process of distributing a number of items amongst a number of agents.*

However, this definition needs to be further qualified: *What* kind of items (resources) are being distributed? *How* are they being distributed (in other words, what kind of allocation procedure or mechanism do we employ)? And finally, *why* are they being distributed (that is, what are the objectives of searching for an allocation and how are these objectives determined)?

1.1.1 Resources

We refer to the items that are being distributed as *resources*, while *agents* are the entities receiving them. We should stress that this terminology is not universally shared. In the context of applications of MARA in manufacturing, for instance, we usually speak of *tasks* that are being allocated to *resources*. That is, in this context, the term "resource" (*i.e.* the resources available to the manufacturer for production) refers to what we would call an "agent" here.

We can distinguish different types of resources. For instance, resources may or may not be divisible. For divisible resources (such as electricity), different agents may receive different fractions of a resource. In the case of indivisible resources, it may or may not be possible for different agents to *share* (jointly use) the same resource (e.g. access to network connections as opposed to items of clothing). For many purposes, *task allocation* problems can be regarded as instances of MARA (if we think of tasks as resources associated with a *cost* rather than a benefit).

1.1.2 Allocations

A particular distribution of resources amongst agents is called an *allocation*. For instance, in the case of non-sharable indivisible resources, an allocation is a partition of the set of resources amongst the agents. The set of resources assigned to a particular agent is also called the *bundle* allocated to that agent.

1.1.3 Agent Preferences

Agents may or may not have *preferences* over the bundles they receive. In addition, they may also have preferences over the bundles received by *other* agents (in the case of network connections, for example, the value of a resource diminishes if shared by too many users). The latter type of preferences are called *externalities*.

Agents may or may not report their preferences truthfully. To provide incentives for agents to be truthful is one of the main objectives of *mechanism design*.

1.1.4 Allocation Procedures

The *allocation procedure* used to find a suitable allocation of resources may be either *centralised* or *distributed*. In the centralised case, a single entity decides on the final allocation of resources amongst agents, possibly after having elicited the agents' preferences over alternative allocations. Typical examples are combinatorial auctions. Here the central entity is the auctioneer and the reporting of preferences takes the form of bidding. In truly distributed approaches, on the other hand, allocations emerge as the result of a sequence of local negotiation steps.

1.1.5 Objectives

The objective of a resource allocation procedure is either to find an allocation that is *feasible* (e.g. to find any allocation of tasks to production units such that all tasks will get completed in time); or to find an allocation that is *optimal*. In the latter case, the allocation in question could be optimal either for the central entity choosing the allocation (e.g. a solution to a combinatorial auction that maximises the auctioneer's revenue); or with respect to a suitable aggregation of the preferences of the individual agents in the system (e.g. an allocation of resources that maximises the average utility enjoyed by the agents).

Combinations are also possible: The objective may be to find an optimal allocation amongst a small set of feasible allocations; and what is considered optimal could depend both on the preferences of a central entity and on an aggregation of the other agents' individual preferences (e.g. auction mechanisms aiming at balancing revenue maximisation and bidder satisfaction). Of course, where computing an optimal allocation is not possible (due to lack of time, for instance), any progress towards the optimum may be considered a success.

1.1.6 Social Welfare

Multiagent systems are sometimes referred to as "societies of agents" and the aggregation of individual preferences in a MARA system can often be modelled using the notion of *social welfare* as studied in Welfare Economics and Social Choice Theory. Examples include utilitarian social welfare, where the aim is to maximise the sum of individual utilities, and egalitarian social welfare, where the aim is to maximise the individual welfare of the agent that is currently worst off.

1.1.7 The Role of Agents

Our discussion shows that the term "multiagent" in Multiagent Resource Allocation can have different interpretations:

- If a *distributed* resource allocation procedure is used, then the term "multiagent" indicates that the computational burden of finding an allocation is shared amongst several agents.

- If an *aggregation of individual preferences* is used to assess the quality of the final allocation, then the term "multiagent" refers to the fact that the choice of allocation depends on the preferences of several agents (rather than on the preferences of a single entity).

Of course, the term "multiagent" could also be derived merely from the fact that resources are being allocated to several different agents. However, if individual agents have no preferences (or such preferences are not taken into account) and the allocation procedure is centralised, then using the term "multiagent" may be less appropriate.

1.1.8 A Computational Perspective

MARA, as introduced at the beginning of Section 1.1, may not seem to differ significantly from what has traditionally been studied in Microeconomics. However, a distinctive feature of MARA is the focus on *computational* issues. For instance, with respect to the preferences of individual agents, we are interested in representations that can be efficiently managed and communicated. Similarly, in the case of allocation procedures, MARA encompasses both the theoretical analysis of their computational complexity and the design of efficient algorithms for scenarios for which this is possible. As a final example, concerning the strategic aspects of negotiation, we may find that classical results in Game Theory fail to hold due to the computational limitations of the participating agents.

1.2 Research Topics

MARA is a highly interdisciplinary field; relevant disciplines include Computer Science, Artificial Intelligence, Decision Theory, Microeconomics, and Social Choice Theory. Research in MARA can take a variety of forms:

- *Preferences:* What are suitable representation languages for agent preferences? Issues to consider include their expressive power, their succinctness, and their suitability in view of preference elicitation.
- *Social welfare:* What are suitable measures of social welfare to assess the quality of an allocation for a given application? Under what circumstances can we expect an optimal allocation to be found?
- *Complexity:* What is the overall complexity of finding a feasible/optimal allocation? What is the complexity of the decision problems that agents need to solve locally? What is the communication complexity (amount of information to be exchanged) of negotiation?
- *Negotiation:* In particular for the distributed approach, what are suitable negotiation protocols? What are good strategies for agents using such protocols?
- *Algorithm design:* How can we devise efficient algorithms for MARA (e.g. algorithms for combinatorial auction winner determination in the centralised case; algorithms to support complex negotiation strategies in the distributed case)?
- *Mechanism design:* How can we devise negotiation mechanisms that force agents to report their preferences truthfully (both to reduce strategic complexity and to allow for a correct assessment of social welfare)?
- *Implementation:* What are best practices for the development of prototypes for specific MARA applications and general-purpose platforms to support quick prototyping?
- *Simulation and experimentation:* How do different optimisation algorithms or negotiation strategies perform in practice? How serious is the impact of theoretical impossibility results in practice? How prohibitive are theoretical intractability results (computational complexity) in practice?
- *Interplay of theory and applications:* What constraints do real-world applications impose on theoretical models for MARA? How can theoretical results inform the development of new tools?

The aim of this survey is to provide a base line for some of these issues. In particular, we present a range of languages for representing preferences, we give an overview of the social welfare measures most relevant to MARA, and we review known complexity results in the area. As it is often difficult to make precise predictions on the performance of a resource allocation procedure by theoretical means alone, we also discuss the requirements to be met by software packages for MARA simulations. To underline the importance of further research in the area, we introduce several prestigious applications and discuss the challenges imposed on MARA models by these applications.

1.3 Paper Overview

The remainder of this survey paper is organised as follows. In Section 2, we introduce four major *application areas* for MARA technology. These are industrial procurement, the joint exploitation of Earth Observation Satellites, manufacturing control, and grid computing. Throughout Section 2, we highlight the specific challenges raised by these applications.

Next we review three important parameters that are relevant to the *definition* of a MARA problem. Firstly, in Section 3, we discuss generic properties of resources, such as being *indivisible* or *sharable*, and how such properties would affect the design of a concrete MARA system. We then move on, in Section 4, to the issue of *preference representation* for individual agents. Each agent needs to be endowed with a suitable representation of preferences over alternative allocations and it is important to be able to express these preferences in a compact way. We discuss both *quantitative* and *ordinal* preference languages. A third parameter in the definition of a MARA problem is the *social welfare* measure (or a similar tool) we

employ to assess the overall quality of a given allocations. A range of different concepts—including *collective utility functions*, *Pareto optimality*, and *envy-freeness*—are reviewed in Section 5.

In Section 6, we attempt to give a short overview of the parameters that are relevant when one chooses (or designs) an *allocation procedure*. We discuss the respective merits and drawbacks of *centralised* and *distributed* approaches to MARA, and we briefly introduce some (centralised) *auction protocols* as well as (distributed) *negotiation protocols*. We also report on results that establish under what circumstances allocations can be expected to *converge* to a socially optimal state in a distributed negotiation setting. Section 7 is a survey of relevant *complexity results*. We mostly concentrate on the *computational complexity* of problems such as finding a socially optimal allocation, but we also briefly discuss issues in *communication complexity* for MARA, which is concerned with the length of negotiation processes.

Our presentation of theoretical issues is complemented by a discussion of software packages for the *simulation* of agent-based market places in Section 8. We start by giving an overview of the typical requirements to be met by such packages and then list the most relevant software products available to MARA researchers interested in simulation. Finally, Section 9 concludes.

2 Application Areas

As mentioned already in the introduction, MARA is relevant to a wide range of application domains. In this section, we introduce four of these problem domains, all of which have recently been addressed by (some of) the authors of this survey.

2.1 Industrial Procurement

The sourcing process of multiple goods or services usually involves complex negotiations that include discussion of product features as well as quality, service, and availability issues. Consequently, several commercial systems to support online negotiation (*e*-sourcing tools) have been developed. In fact, *e*-sourcing is becoming an established part of the business landscape [90]. However, there are still enormous challenges confronting users who want to get the maximum value out of *e*-sourcing.

2.1.1 Problem Description

Traditionally, the core of the sourcing process comprises the following tasks:

- request for quotation/proposal (RFQ/RFP);
- provider selection for RFQ/RFP delivery;
- offer generation;
- negotiation through offer/counter-offer interaction or reverse auction; and
- selection of best offers.

Typically a buyer creates an RFQ by sequentially adding items. Each item specifies a product, be it a good or service. A paradigmatic example of multi-item RFQ occurs in industrial settings. The production plan outlined by a company's ERP (Enterprise Resource Planning) or SCM (Supply Chain Management) application comes in the shape of a list of items to be produced along with the parts required for each product, the so-called bill of material. This is the basis for the buyer to initiate multiple sourcing events, each devoted to the procurement of the parts for each of the items to be produced.

Although several commercial systems to support online negotiations have been released, to the best of our knowledge, not a single system can claim to address the full complexity of online negotiation. The first generation of sourcing tools merely incorporate single-item, price-quantity reverse auction mechanisms. Others only offer basic negotiation capabilities that are usually reduced to a demand-offer matching tool. In general terms, there is a lack of decision support functionalities (decision making in sourcing can involve a few hundred offers, each of which is described by several dozen attributes). Furthermore, there is a lack of technology support for computationally complex negotiation paradigms, which inhibit the application of promising mechanisms such as combinatorial reverse auctions [24, 54].

2.1.2 Challenges

Although the degree of automation, namely of delegation to trading agents, in industrial procurement settings is still low, we do believe that MARA techniques can contribute to improve this

situation. In what follows we identify several challenges that any commercial tool aiming at the successful implementation of resource allocation amongst several (human or software) agents in an industrial procurement setting must address.

- *Preferences of buyers and providers.* How do we best capture and represent trading agents' preferences so that they can effectively value their trading partners' offers, counter-offers, and RFQs? While recent advances in preference elicitation are encouraging (see, for instance, the work of Bichler et al. [6]), this still remains as the Achilles' heel of industrial procurement applications.

- *Business rules to constrain admissible allocations.* While in direct auctions, the items to be sold are physically concrete (they do not allow configuration), in a negotiation involving highly customisable goods, buyers need to express relations and constraints between attributes of different items. On the other hand, multiple sourcing is common practice, either for safety reasons or because offer aggregation is needed to cope with high-volume demands. This introduces the need to express constraints on providers and on the contracts they may be awarded. Providers may also impose constraints on their offers. Therefore, highly expressive languages for both buying and providing agents are required.[1] Incorporating business rules into allocation procedures can lead to more *balanced* and *safer* allocations.

- *Automated negotiation strategies.* There are several dimensions to take into account when designing negotiation strategies. Agents may negotiate over multiple attributes of the same item, over a bundle of multiple items, or they may hold separate but interdependent negotiations. Negotiation techniques such as trade-off [37] or partial-order scheduling [102] are candidate techniques put forward from the research arena. The current procurement practices tell us that the possibility of automatic offer submission is seen with interest for repetitive sourcing events in private *e*-sourcing platforms where providers and business rules are well-known or result from a provider qualification procedure or a frame contract. Nonetheless, the full application of such automated trading still faces barriers, such as providers not wanting to reveal their capabilities/preferences to third parties.

- *Choice of mechanism.* Commercial sourcing tools offer an ever increasing number of customisable negotiation mechanisms. Nonetheless, market design is a highly complex, intricate task. New trends in automated mechanism design [22] as well as evolutionary mechanism design [73] may prove valuable in assisting in the design of market scenarios that ensure certain global properties.

- *Winner determination algorithms.* Further research into algorithms capable of identifying the optimal set of offers in multi-attribute, multi-item negotiation scenarios with side constraints representing business rules is required [45, 83].

- *Bundling.* Should a buyer (seller) conduct a single negotiation or auction for an entire bundle of goods he or she is interested in purchasing (selling) or should they group items into bundles and conduct several negotiations? Unfortunately, for complexity reasons, combinatorial bidding capabilities are rarely found on commercial systems. To overcome this problem, we can think of a third approach: Based on past market real data and knowledge, the whole bundle of items can be divided into separate negotiations for which the appropriate providing agents are invited and for which certain properties are satisfied (e.g. invite providing agents that can offer at least 90% of the items in the bundle). These

[1] Consider a buyer who wants to buy 200 chairs (any colour/model is fine) for the opening of a new restaurant and who uses an *e*-procurement solution that launches a reverse auction. If we employ a state-of-the-art combinatorial auction solver, a possible solution might be to buy 199 chairs from provider A and 1 chair from provider B, simply because this is 0.1% cheaper than the next best allocation and it has not been possible to specify that, in case of buying from more than one provider, a minimum of 20 chairs purchase is required. In a different scenario, the optimal solution might tell us to buy 150 blue chairs from provider A and 50 pink chairs from provider B. Why? Because, although we had no preferences over the regarding colour, we could not specify that all chairs should be of the same colour. Although simple, this example shows that without modelling natural constraints, solutions obtained may be mathematically optimal, but unrealistic.

properties model the expertise of *e*-sourcing specialists in the form of rules of thumb [76].

Some of these challenges are already being tackled by recently developed negotiation support tools. *iBundler* [44, 44], for instance, is an agent-aware decision support service acting as a combinatorial negotiation solver for both multi-item, multi-unit negotiations and auctions that can integrate business rules to constrain admissible solutions. *iAuctionMaker* [76] is a novel decision support tool for mixed bundling that can help an auctioneer determine how to group items into *promising* bundles that are likely to produce a high revenue. Promising bundles are those that satisfy certain properties believed to be present in competitive sourcing scenarios. These properties are defined by *e*-sourcing professionals and capture their experience and knowledge in the domain.

2.2 Earth Observation Satellites

Next we consider another real-word application, namely the exploitation of Earth Observation Satellites (EOSs) [10, 59, 60]. This application pertains to the problem of allocating a set of indivisible goods to some agents with no possible monetary compensation between them. As we will see, this is a typical case of a sharing problem, different from an auction situation, especially because fairness is a key issue.

2.2.1 Problem Description

Due to their high cost, space projects such as EOSs are often co-funded and then exploited by several agents (countries, companies, civil or military agencies, etc.). The mission of an EOS is to acquire images (photos) of specified areas on the earth surface, in response to observation demands from users. Such a satellite is operated by an Image Programming and Processing Center. Each day, the Center collects a set of observation demands from agents. Usually a demand can be covered by a single image, but more complex demands may arise, as we will see below. Each demand is given a *weight* (a positive integer), reflecting the importance the requesting agent assigns to the satisfaction of the demand. The daily task of the Center is, amongst others, to build the imaging workload of the satellite for the next day, by selecting the images to be acquired from the set of agent demands.

Naturally, the exploitation of the satellite must obey a set of *physical constraints*, such as time window visibility constraints, minimum transition times between successive image acquisitions, or memory and energy management. Due to these exploitation constraints, and due to the large number of (possibly conflicting) demands, a set of demands, each of which could be satisfied individually, may not be satisfiable as a whole on a single day. All these physical constraints define the set of *admissible* allocations of images to agents. The exploitation of an EOS must also meet the following requirements:

- *Efficiency:* The satellite should not be underexploited.
- *Equity:* Each agent should get a return on investments that is proportional to its financial contribution.

2.2.2 Modelling

Let us first consider the simple problem where only one agent exploits the resource. In this case, the allocation problem consists of selecting, each day, an admissible sequence of images that will be acquired by the satellite over the next day (and allocated to the agent). This agent measures its satisfaction by a utility function which may be defined as the sum of the weights of the allocated images. The efficiency requirement comes down to a simple optimisation problem: the utility function of the agent must be maximised over the set of admissible allocations (see Lemaître et al. [61] for the description of some algorithms for solving this mono-agent allocation problem).

We now turn to the case where several agents exploit the satellite. For simplicity, we assume in this paper that the agents have equal rights over the resource (we may assume, for example, that they have funded the satellite equally). Of course, each agent wants to maximise its own utility function, but generally they are antagonistic: increasing the utility of one agent can lead to decreasing the utility of others. So a fair compromise must be found, the realisation of which is the role of a suitable preference aggregation mechanism. Such mechanisms will be discussed in detail in

Section 5. Here, the *min* function (egalitarian social welfare) fits our requirements, as it naturally conveys the equity requisite: we try to make the agent least happy as happy as possible (a refinement of this approach is given by the so-called *leximin* ordering; see Section 5.4).

As mentioned before, weights of demands are freely fixed by agents. In order to be able to compare individual utilities between agents, a common utility scale must be set and used; that is, the same number should express the same level of satisfaction. To this end, Lemaître et al. [59, 60] have adopted an approach known as the *Kalai-Smorodinsky* solution (see Section 5.6), whereby individual utilities are compared relative to the maximum utility that each agent can receive. It should be noted that, unlike for auction problems, there are no preemptive constraints in this application: the same image could be requested by several agents, and allocated to them all (*i.e.* resources are sharable).

This application is also of interest because it offers real-word examples of dependencies between demands. As a first example, a request may involve a pair of stereoscopic images; receiving only one image would result in a poor satisfaction level for the agent. A second example comes from the fact that, for earth areas situated in high latitudes, several images of the same area can be taken from distinct angles during the same day. Consider a stereoscopic demand concerning such an area, and suppose that it could be photographed from two angles. Let $o11$ and $o12$ be the pair of stereoscopic images from angle 1, $o21$ and $o22$ the images from angle 2. The demand can be quite naturally formulated as $(o11 \wedge o12) \vee (o21 \wedge o22)$.

To sum up, our EOS multiagent fair resource allocation problem can be formally stated in the following way. Agents express their (weighted) demands as simple logical propositions. An agent's individual utility is the sum of the weights of the satisfied demands. The global utility is an aggregation of normalised individual utilities, the aggregation function being the *min* function (or, better, the *leximin* ordering).

2.3 Manufacturing Systems

Since the second half of the 20th century, the organisation of mass production has been shifting towards flexible manufacturing and customised products. From a technological point of view, it has been observed that current manufacturing systems (e.g. computer-integrated manufacturing architectures) have several drawbacks, in particular excessive rigidity and centralisation [50, 71]. Furthermore, future manufacturing systems are expected to be characterised by globally distributed production units, small quantities of a large variety of products, the provision of individual solutions tailored to each customer's specific needs, and concurrent execution of all the activities in the manufacturing process [95].

2.3.1 Problems and Requirements

Future manufacturing systems therefore require coordination amongst production units and it is expected that rigid, static, and hierarchical manufacturing systems will give way to systems that are more adaptable to rapid change [15]. In order to overcome the identified problems with current manufacturing systems and prepare them for the expected future scenarios, the new generation of systems must possess such attributes as decentralisation, distribution, autonomy, adaptability, and incomplete information handling [88].

In manufacturing, the term *resource allocation* is usually synonymous for *task scheduling*. Furthermore, the term *resource* is understood as a physical resource, *i.e.* a machine, of the manufacturing plant. One of the problems in this area is that a task is a step of a production plan for a specific order (e.g. manufacture 100 chairs of type `P12-5`), and there are usually dependencies between tasks that must be obeyed (e.g. operation "drill hole 2" must be done before "cool surface" but after "drill hole 1").

To further complicate things, the tasks involved in a production plan will probably be done on different production resources, thus creating a network of dependencies amongst resources.

One issue in the manufacturing area is that the schedule itself is only valid until the first disturbance (e.g. machine or tool breakdown, rush order, etc.). Since manufacturing control and execution is a real time application, the need to find a *feasible* solution is much greater than to find one that is *optimal*. The system as a whole must reach a stable and feasible schedule without too much interruption of the shop floor.

2.3.2 Manufacturing and Agents

Physically, a manufacturing system involves several resources (numeric control machines, robots, automated guided vehicles, conveyors) and several tasks can be carried out at the same time. The number and configuration of these may change of the lifetime of the system. Since the manufacturing process is dynamic (e.g. suppliers and consumers in a supply chain may change many times) it is impossible to know the exact structure or topology of the system in advance. The number of products and orders, as well as different alternative production routes, account for the highly complex nature of manufacturing systems.

All of the above make the design of manufacturing systems an excellent candidate for the application of agent-based technology. In many implementations of multiagent systems for manufacturing scheduling and control, the agents model the resources of the plant and the scheduling and control of the tasks is done in a distributed way by means of cooperation and coordination of actions amongst agents [15, 53, 72]. As such, manufacturing scheduling and control touches the areas of distributed planning and distributed artificial intelligence. Nonetheless, there are also approaches that use a single agent for scheduling (usually with a well-known centralised scheduling algorithm) that dictates the schedules that the resource agents will execute [92]. The rationale for modelling resources as agents is to better mimic the actual real-world environment and to allow for the modelling of the characteristics of each resource (e.g. available operations, own agenda of tasks to execute, cost of performing each operation, etc.)

When responding to disturbances, the distributed nature of multiagent systems can also be a benefit to the rescheduling algorithm by involving only the agents directly affected, without disturbance to the rest of the community that can continue with their work. Typical approaches to rescheduling include the removal of a late order, reallocation of low priority orders to make room for rush orders, shifting of tasks from one resource to a similar one, etc.

An example for a MARA system for manufacturing control is the *Fabricare* prototype suite [89]. This a multiagent system for dynamic scheduling of manufacturing orders. The agents are modelled as extended logic programs with the ability to handle negative and incomplete knowledge [88]. The system is very dynamic in what concerns its agents, *i.e.* resource agents depend on the system description file; task agents depend on the existing tasks (dynamic events). Each negotiation uses the set of agents that are present and available at that time, thus giving the system a high degree of adaptability to the dynamic nature of the manufacturing arena.

2.4 Grid Computing

Perhaps one of the most pressing applications for MARA techniques is *grid computing* [40]. It is true that there are functioning systems for grid resource allocation, but these largely operate in benevolent, cooperative subnets where participants know and trust one another and there is typically no charge for the utilisation of resources, although perhaps some artificial accounting system is applied. Such frameworks are exactly what is needed in order to test out many grid middleware functions where the objective is to see a job executed across a range of grid resources. In many respects, grid resource allocation—as distinct from scheduling—and payment for resource usage is an orthogonal problem to the actual processing of a job.

However, at some stage, if the vision of grid computing as a commodity not unlike energy is to become reality, then resource allocation, payment and job processing will have to come together and current research in MARA technology aims to lay the foundations for this union.

2.4.1 Scalability Issues

If the benevolent, cooperative network of mutually trusting participants is discarded, the client is faced with the problem of piecing together a range of disparate resources that are required to complete the processing of a particular job. The parallels with markets, and especially commodity markets, as efficient (by economic measures) resource allocation mechanisms in the presence of large numbers of traders and where possibly complex packages of goods are required, are striking. Grid networks have not yet become so large as to make such approaches essential, but that time is

not far off, even in cooperative scientific research networks, if one considers the grid that is foreseen to support the analysis of results emerging from CERN's Large Hadron Collider [17].

The issues could be seen as a function of scale: Existing grids can handle resource allocation through single centralised mechanisms and (economic) efficiency of allocations may not be important. As grids become larger with a wider range of resources, and used for broader classes of tasks, centralised allocation and inefficient allocation of resources are likely to become less tolerable. In response to this, various approaches need to be evaluated and contrasted under carefully controlled conditions, from centralised systems seeking optimal allocation to distributed mechanisms involving bilateral negotiation. Intuition—which should of course be treated with circumspection—suggests that neither of these can be entirely satisfactory, but each may act in different ways as benchmarks against which to measure the rest:

- Centralised systems relying on combinatorial auction clearing algorithms can deliver optimal allocations, but are currently limited by computation costs to hundreds of items and thousands of bids [81].
- Distributed systems relying on bilateral negotiation between consumer and service provider for each component—that is, the consumer constructs their own bundle—will almost certainly scale, but the results are much less likely to be "good". The risks inherent in such an approach are significant: The order in which to undertake negotiation, the possibility that contracting for one resource constrains the choice of subsequent resources, perhaps leading to incomplete bundles, the difficulty in assessing the quality of a bundle or indeed the valuation of a bundle are all surrounded by uncertainty.

Implicit in both scenarios is that a client will need to combine a range of resources from the grid in order to carry out their computation.

2.4.2 Market-based Allocation

In between the two extremes of centralised and distributed lie the many variants of market-based allocation. And given the essentially decentralised nature of (geographically dispersed) grids, potentially with many administrative centres and relatively weak control over individual nodes, the grid seems well suited to market-based schemes, where the twin benefits of reputation and decentralised negotiation can facilitate the trading of computational resources.

Among the different market schemes that exist, one approach is to mimic ideas seen in commodity trading [48]. While analogies are both risky and seductive, there do seem to be sufficient parallels to make more detailed exploration—and simulation—desirable. Commodity markets are a blend of centralised and distributed in that there are many commodity markets around the world, such that at any one time a significant subset are trading, giving a 24/7 market, but within any given market trades take place through bilateral mechanisms, typically continuous double auctions. However, a trader may participate in more than one market at a time, giving rise to communication between markets as to current valuation trends along with the publication of "closing prices".

But, commodity markets typically trade in lots of a single kind and depending on the market, traders may be direct buyers and sellers with no middle-men or market-makers. Economic analyses and simulations indicate that market-makers increase liquidity and enable the market to remain (economically) efficient at lower levels of participation than in the presence of buyers and sellers alone [7]. Furthermore, in the case of bundles (lots of varying quantities of several kinds of goods), market-makers become repositories of market memory, learning what bundles work (potentially a combination of reputation and fit of resources) and identifying trends as new kinds of bundles emerge. Thus they become more than mediators between buyer and seller, fully justifying the epithet of "market-maker". A trading framework such as this seems highly applicable to grids and resource allocation within grid systems.

3 Types of Resources

A central parameter in any resource allocation problem is the nature of the resources themselves. In this section, we give a brief overview of the (abstract) properties of different types of resources. Some of these properties are characteristics of the

resources (such as being perishable rather than static, or continuous rather than discrete), while others are better understood as being characteristics of the chosen allocation system (for instance, whether or not a given item is sharable amongst several agents will typically depend on the allocation procedure rather than on characteristics of the item itself).

3.1 Continuous vs. Discrete

A resource may be either *continuous* (e.g. energy) or *discrete* (e.g. fruit). This "physical" property will typically influence how the resource is being traded, although this need not be the case. For instance, a continuous resource will typically be regarded as being (infinitely) divisible. Still, in a particular negotiation setting, it may only be possible to buy or sell a certain quantity of such a continuous resource as a whole. Individual units of a discrete resource, however, are always indivisible (an apple that can be sold in small pieces would not count as a discrete resource).

In a setting with several continuous resources, a bundle can be represented as a vector of non-negative reals (or, alternatively, numbers in the interval $[0, 1]$ to denote the proportion of a particular resource owned by the agent receiving the bundle). Bundles of discrete resources can be represented as vectors of non-negative integers. If there is just a single item of each resource in the system, then vectors over the set $\{0, 1\}$ suffice.

A continuous resource may be *discretised* by dividing it into a number of smaller parts to be traded as indivisible units. For instance, rather than treating 10.000 litres of orange juice as a truly continuous resource that could be divided into ever smaller subparts, we may agree to negotiate over 200 units of 50 litres each. This means that methods developed for discrete MARA are often also applicable in the continuous case (although they may not be as efficient as methods specifically tailored to continuous resources).

The allocation of continuous resources (often just a single continuous resource), has been studied in depth in the classical literature in Economics. More recent work in Computer Science and Artificial Intelligence, on the other hand, has often focussed on discrete resources. In this paper, we also concentrate on discrete resources.

3.2 Divisible or not

As discussed above, resources may treated as being either *divisible* or *indivisible*. While being either continuous or discrete is a property of resources themselves, the distinction between divisible and indivisible resources is made at the level of the allocation mechanism. In this survey, we concentrate on indivisible resources.

3.3 Sharable or not

A *sharable* resource can be allocated to a number of different agents at the same time. An example of such sharable resources can be found in the context of the Earth Observation Satellite application discussed earlier (see Section 2.2): A single picture taken by the satellite can be allocated to several different agents (no preemptive constraints). The canonical case, however, considers resources as being *non-sharable* and in the rest of this paper we also make this assumption.

3.4 Static or not

A resource may be *consumable* in the sense that the agent holding the resource may use up the resource when performing a particular action. For instance, fuel is consumable. Also, resources may be *perishable*, in the sense that they may vanish or lose their value when held over an extended period of time. Food is a classical example of a perishable resource.

We call resources that do not change their properties during a negotiation process *static* resources. In general, resources cannot be assumed to be static. In MARA however, it is often assumed that they are (that is, that resources are neither consumable nor perishable). The rationale behind this stance is the fact that the negotiation process is not really concerned with the actions agents may undertake outside the process itself. That is, even if a resource is either consumable or perishable, we can often assume that it remains static throughout a particular negotiation process. In this paper, in particular, we concentrate on static resources.

3.5 Single-unit vs. Multi-unit

In a *multi-unit* setting it is possible to have many resources of the same type and to refer to these

resources using the same name. Suppose, for instance, there are a number of bottles of champagne available in the system, but that agents cannot distinguish between these bottles. In a *single-unit* setting, on the other hand, every item to be allocated is distinguishable from the other resources and has a unique name.

The differentiation between single- and multi-unit settings is a matter of representation. Any multi-unit problem can, in principle, be transformed into a single-unit problem by introducing new names for previously indistinguishable items. Vice versa, clearly, any single-unit problem is also a (degenerate) multi-unit unit problem. An important advantage of working within a multi-unit setting is that it may allow for a more compact way of representing both allocations and the preferences of agents over alternative bundles. On the downside, a richer language (variables ranging over non-negative integers, rather than binary values) is required in this case.

3.6 Resources vs. Tasks

At a sufficiently high level of abstraction, a task allocation problem can be reduced to a resource allocation problem. Indeed, *tasks* may be considered *resources* to which agents assign a negative utility. However, an important characteristic of tasks as opposed to resources is the fact that tasks are often coupled with constraints regarding their coherent combination. For instance, a task may require the achievement of another task as a precondition. In this respect, treating allocations merely as assignments of bundles of items to agents (without associated time constraints, for instance) would be too simple a model.

In this paper, however, we concentrate on general resource allocation problems rather than issues that are specific to task allocation (and exception is our discussion in Section 2.3).

4 Preference Representation

Preferences express the relative or absolute satisfaction of an individual when faced with a choice between different alternatives.[2] In the context of MARA, these alternatives are the different potential allocations of resources, or more concretely, the bundle of resources received by an agent for each of the alternative allocations.

A *preference structure* represents an agent's preferences over a set of alternatives X. There are several choices that can be made regarding the definition of a mathematical model for preference structures (this is an important question that has been discussed by researchers in decision theory for a long time). We can distinguish four families of preference structures:

- A *cardinal* preference structure consists of an evaluation function (generally called *utility*) $u : X \to Val$, where Val is either a set of numerical values (typically, $\mathbb{N}$, $\mathbb{R}$, $[0, 1]$, $\mathbb{R}^+$, etc.), or a totally ordered scale of qualitative values (e.g. linguistic expressions such as "very good", "good", etc.). In the former case the preference structure is called *quantitative*, in the latter it is called *qualitative*.

- An *ordinal* preference structure consists of a binary relation on alternatives, denoted by $\preceq$, which is reflexive and transitive (and usually, although not necessarily, complete).[3]

  We write $x \prec y$ (strict preference) if and only if $x \preceq y$ but not $y \preceq x$, and $x \sim y$ (indifference) if and only if both $x \preceq y$ and $y \preceq x$.

- A *binary* preference structure is simply a partition of X into a set of *good* and a set of *bad* states. A binary preference structure can be seen as both a (degenerate) ordinal preference structure and a (degenerate) cardinal preference structure.

- A *fuzzy* preference structure is a fuzzy relation over X, *i.e.* a function $\mu : X \times X \to [0, 1]$. $\mu(x, y)$ is the degree to which x is preferred over y. Fuzzy preferences are more general than both ordinal and cardinal preferences.

Since fuzzy and qualitative preferences have not been used much as far as resource allocation is concerned, we are going to neglect these in this

[2] This is the decision-theoretic view of preferences, shared by many communities, from mathematical economics to multi-criteria decision making.

[3] Some work in preference modelling has also addressed non-transitive preference relations, arguing that humans often exhibit non-transitive preferences—for the sake of brevity we will omit this issue here.

survey, and focus on quantitative and ordinal preferences instead.

Observe that we have three "levels" for preference modelling, according to the possible operations allowed by the preference structure: Ordinal preferences allow only for comparing the satisfaction of a given agent for different alternatives, but cannot express preference intensity and do not allow for interpersonal comparison of preferences (that is, expressing statements such as "agent i is happier with x than agent j with y"). Qualitative preferences do allow for interpersonal comparison of preferences, and can express a weak form of intensity, but they do not allow for any "metric" use of preferences such as computing the difference between two utility degrees so as to allow for a monetary compensation—while quantitative preferences do.

Note that any cardinal preference induces an ordinal preference, namely for a utility function u we can define the complete weak order $\preceq_u$ given by $x \preceq_u y$ if and only if $u(x) \leq u(y)$.

The *explicit* representation of a preference structure consists of the data of all alternatives with their associated utilities (for cardinal preferences) or the whole relation $\preceq$ (for ordinal preferences). These representations have a spatial complexity in $O(|X|)$ for cardinal structures and $O(|X|^2)$ for ordinal structures, respectively.

In many real-world domains, the set of alternatives X is the set of assignments of a value to each of a given set of variables. In such cases, the alternatives are exponentially many. It is not reasonable to ask agents to report their preference in an explicit way when the set of alternatives is exponentially large, as this amounts to listing the exponentially many alternatives together with their utility assessment or their ranking. This is the case, in particular, when alternatives are allocations of resources (assignments of resources to agents). For this reason, the MARA project needs *languages for preference representation* aiming at enabling a *succinct* representation of the description of the problem, without having to enumerate a prohibitively large number of alternatives. Such preference representation languages often allow for a much more concise representation of the preference structure than an explicit enumeration.

In this section, we are going to give a brief survey of languages for preference representation. We begin by discussing several ways of representing compactly quantitative preferences (that is, utility functions), including languages specifically introduced for combinatorial auctions, and then we move on to languages for representing ordinal preferences.

4.1 Quantitative Preferences

Let $\mathcal{R} = \{r_1, \ldots, r_m\}$ be a set of indivisible resources. A quantitative preference structure for a resource allocation problem is a utility function $u : 2^{\mathcal{R}} \to \mathit{Val}$ mapping bundles of resources (subsets of $\mathcal{R}$) to numerical values (such as the reals). By defining utilities over bundles, we assume that the preferences of agents are free of so-called *allocative externalities*. That is, the value that an agent assigns to a bundle R does not depend on the allocation of the remaining resources amongst the other agents.

In the case of *task allocation* (as opposed to resource allocation), we may model the preferences of agents using *cost functions* rather than utility functions. At the level of abstraction being considered in the present survey, there is no effective difference in the representation of utility functions and cost functions. In the former case, agents would usually aim at maximising their utility, while in the latter case they would aim at minimising their costs.

Next we are going to review several languages for representing utility functions.

4.1.1 Bundle Enumeration

The most basic form of representing a utility function is to enumerate the bundles to which it assigns a non-zero value. That is, a utility function u can be presented as the set of pairs $\langle R, u(R)\rangle$ with R those bundles of resources for which $u(R) \neq 0$. We call this the *explicit form*, or the *bundle form*.

The bundle form is obviously *fully expressive* in the sense that any utility function may be so described. A serious drawback, however, is that the length of such descriptions will typically be exponential in the number of resources. This has prompted researchers to develop more succinct languages for utility representation.

4.1.2 The k-additive Form

For some (but not all) utilities it is possible to exploit *regularities* in the function structure in order to build succinct and efficiently computable descriptions. Given $k \in \mathbb{N}$, a utility function u is said to be *k-additive* if and only if there exists a coefficient α_T for each set of resources T of size at most k such that:

$$u(R) \quad = \quad \sum_{T \subseteq R} \alpha_T$$

The coefficient α_T represents the synergetic value of owning all of the items in T together, beyond the utility associated with any of its proper subsets. If a utility function is presented in terms of such coefficients, then we say that it is given in *k-additive form*.

The k-additive form is also fully expressive, but only in the sense that it can describe any utility function provided k is chosen large enough (for any k less than the overall number of resources there are functions that cannot be represented). It is typically considerably more succinct than the simple bundles form (think of a function mapping bundles to the number of items in a bundle), although there are also counterexamples (such as functions mapping only bundles with a single element to a non-zero utility value) [18].

In many application domains, it will be reasonable to assume that utility functions are k-additive with a relatively small value of k (which would allow for a very succinct representation). Indeed, the larger a bundle of resources, the more difficult for an agent to estimate the additional benefit incurred by owning all the resources in that bundle *together* (*i.e.* beyond the benefit incurred by the relevant subsets).

The k-additive form of representing utility functions is inspired by work in fuzzy measure theory [47]. It has been introduced into the MARA domain by Chevaleyre et al. [18] and, independently and in a combinatorial auction setting, by Conitzer et al. [23].

4.1.3 Weighted Propositional Formulas

Many languages for compact preference representation make an explicit use of *logic* (for a survey of such languages we refer to the work of Lang [57]). The basic idea of logic-based preference representation for MARA is that each resource r can be identified with a propositional variable p_r, which is *true* if the agent whose preferences we are modelling owns the corresponding resource, and *false* otherwise.[4] That is, every bundle R corresponds to a model. Agents can then express their preferences in terms of propositional formulas (or *goals*) that they want to be satisfied. We write $R \models G$ to express that the goal G is satisfied in the model corresponding to the bundle R.

The simplest (and prototypical) logical representation of preferences simply consists of giving a single propositional formula G (representing the agent's goal). The utility function u_G generated by G is extremely basic: $u_G(R) = 1$ if $R \models G$, $u_G(R) = 0$ if $R \models \neg G$. One possible refinement of this consists of considering a *goal base* $GB = \{G_1, \ldots, G_n\}$ and counting the number of goals satisfied by R.

In a further refinement, goals are associated with numerical weights, which tell how important the satisfaction of the goal is considered to be. Formally, the preferences of an agent are expressed by means of a finite set of such weighed goals: $GB = \{\langle G_1, \alpha_1\rangle, \ldots, \langle G_n, \alpha_n\rangle\}$, where each α_i is an integer and each G_i is a propositional formula. For every bundle R, we define the *penalty* of R as follows:

$$p_{GB}(R) \quad = \quad \sum\{\alpha_i \mid R \not\models G_i\} \qquad (1)$$

The penalty of R can be viewed as its disutility, that is, $u_{GB}(R) = -p_{GB}(R)$. Many other operators can be used, in place of the sum, for aggregating weights of violated (or symmetrically, satisfied) formulas [56].

4.1.4 Straight-line Programs

A further representation form for utility functions is based on *straight-line programs* (SLPs). SLPs may be viewed as directed acyclic graphs consisting of two distinguished types of vertex: *inputs* which are sources (have in-degree 0) and *gates*, each of which has in-degree exactly 2. A subset of the gates (with out-degree 0) are distinguished as the program *outputs*. In addition to the graph structure an SLP is fully defined by associating a binary Boolean operation with each

[4] In a multi-unit setting (see Section 3.5), we would have to consider atomic sentences such as $x \geq 50$, signifying a bundle with at least 50 units of type x.

gate vertex. For an SLP, C, with m inputs, ordered as $\langle x_1, x_2, \ldots, x_m \rangle$ and p gates, a *topological labelling* of the vertices assigns a unique integer in the range $[1, m+p]$, $\lambda(v)$, to each vertex of C in such a way that: $\lambda(x_i) = i$; if v is a gate with $\langle w_1, v \rangle$ and $\langle w_2, v \rangle$ edges in C then $\lambda(v) > \max\{\lambda(w_1), \lambda(w_2)\}$. A topological labelling may be efficiently computed for C using depth-first search.

An SLP, C with inputs $\langle x_1, x_2, \ldots, x_m \rangle$ and p gates, t of which are outputs labelled $\langle s_0, s_1, \ldots, s_{t-1} \rangle$ computes its result by executing the program consisting of exactly $m + p$ lines, at each of which a single bit value ($res(i)$) is computed. Given an instantiation of the inputs $\langle \alpha_1, \ldots \alpha_m \rangle$, the ith line, l_i computes: $res(i) := \alpha_i$ if $1 \leq i \leq m$; and $res(j_1)\theta_i res(j_2)$ if $m + 1 \leq i \leq m + p$, where θ_i is the binary operation associated with the ith gate and whose inputs are the vertices labelled j_1 and j_2. The *numerical* value computed by C as a consequence of a particular instantiation $\underline{\alpha}$ of its inputs is $val(C, \underline{\alpha}) = \sum_{i=0}^{t-1} res(s_i) \cdot 2^i$.

This model provides an alternative representation for utility functions, $u : 2^{\mathcal{R}} \rightarrow \mathbb{N}$ by a suitable SLP, C: a subset S defines an instantiation of the inputs via its m-bit characteristic vector $\alpha(S)$; the value $u(S)$ is then simply $val(C, \alpha(S))$. It is noted that, although this definition uses $\mathbb{N}$ as the range, it is a trivial matter to extend to $\mathbb{Z}$ (allow an additional output to act as a *sign* bit) and to $\mathbb{Q}$ (interpret the output bits as two groups, one defining the numerator, the other the denominator). As with the bundle form, the SLP form has the property of being fully expressive. In addition, however, there are the following advantages:

- The number of bits needed to encode utility functions can be exponentially smaller than that required in the bundle form.
- If the function $u : 2^{\mathcal{R}_m} \rightarrow \mathbb{Q}$ is computable by a deterministic Turing Machine in time T, then u may be represented by an SLP, C containing $O(T \log T)$ lines.

The first of these is easily seen by considering the function with value 1 if $|S|$ is odd, and value 0 if $|S|$ is even: the number of bundles to be listed is exactly 2^{m-1}. The same function, however, is described by the program with $2m - 1$ lines corresponding to the computation $\oplus_{i=1}^{m} x_i$. The second property is a consequence of the constructions presented by Schnorr [85] and Fischer and Pippenger [39]. These simulations are effective (*i.e.* not simply existence arguments) and can be efficiently implemented.

In principle, other "program-based" formalisms can be defined, however, in order to be effective it must be possible efficiently to validate that a given bit-string does describe a syntactically correct program *and* to have an effective method of determining the program output. For the SLP approach above, both of these are satisfied, the latter since the runtime of a given SLP is exactly the number of program lines contained within it.

Extensive complexity-theoretic treatments of the SLP model (described in its usual terminology of *combinational logic networks*) may be found in the monographs of Savage [84], Wegener [96] and Dunne [28]. In the context of MARA, the SLP form has been considered by Dunne et al. [32].

4.1.5 Bidding Languages

Bidding languages are used in combinatorial auctions to allow agents to communicate their preferences to the auctioneer.[5] Preferences structures here are *valuation functions* or, equivalently, positive and monotonic utility functions on $2^{\mathcal{R}}$.

Bids are expressed as combinations of atomic bids of the form $\langle R, p \rangle$, where p is the amount the bidder is prepared to pay for the bundle R. Two prominent bidding languages are the OR and the XOR languages:

- The OR language is probably the most widely used bidding language. Here the valuation of a bundle is taken to be the maximal value that can be obtained when computing the sum over *disjoint* bids for subsets of the bundle. For instance, a bid of the form

  $$\langle \{a\}, 2 \rangle \text{ OR } \langle \{b\}, 2 \rangle \text{ OR } \langle \{c\}, 1 \rangle \text{ OR } \langle \{a, b\}, 5 \rangle$$

  expresses that the bidder is willing to pay 2 for a alone, 2 for b alone, 5 for both a and b, and 6 for the full set. Clearly, this language is not fully expressive since it cannot represent subadditive utility functions (for example, there is no way to specify that you would only be prepared to pay 4 for the full set).

[5] Of course, strategic considerations may cause agents not to report their *true* preferences, but this issue is not relevant from the viewpoint of preference *representation*.

– In the XOR language [80], atomic bids are assumed to be mutually exclusive. In this case, the valuation of a bundle is simply the highest value offered for any of its subsets. The XOR language can express any (normalised) monotonic utility function.

While the XOR language is more expressive than the OR language, it can also prove to be far less compact for certain types of preferences. For instance, the utility function $u(R) = |R|$ requires an exponential number of atomic bids in the XOR language, but only a linear number of OR bids.

Because the OR language is widely considered a simple and natural bidding language, there have been several attempts to extend its expressiveness without requiring an exhaustive listing of XOR bids. It is, for instance, possible to combine the types of bids, and to thus to obtain OR-of-XOR and XOR-of-OR bidding languages. For an extensive discussion of such languages we refer to the review article by Nisan [68].

An interesting alternative is to simulate XOR bids by means of OR bids. The idea is simply to introduce "fake" resources (or phantom goods, or dummy items), that have no function other than making bundles mutually exclusive, because the resource appears in both bundles [41]. For instance, if one wanted to express that the set $\{a, b, c\}$ should be valued at 4, it would be possible to add the fake resource f to obtain both $\langle\{c, f\}, 1\rangle$ and $\langle\{a, b, f\}, 5\rangle$, and to bid in addition on $\langle\{a, b, c\}, 4\rangle$. This language, known as the OR$*$ bidding language (or OR with dummy items), is as expressive as the XOR language.

4.2 Ordinal Preferences

Next we are going to discuss the representation of ordinal preferences. Again, let $\mathcal{R} = \{r_1, \ldots, r_m\}$ be a set of indivisible resources. An ordinal preference structure $\preceq$ is a binary relation over $2^{\mathcal{R}}$. Here, logic-based languages play a central role (see also our discussion of weighted propositional formulas in Section 4.1.3).

4.2.1 Prioritised Goals

Prioritised goals are the ordinal counterpart to weighted goals: instead of numerical weights attached to goals (expressed as propositional formulas), we have a priority relation on goals, from which a preference relation on the set of bundles can be drawn.

While some approaches make use of partial priority preorders, most of them make the assumption that the priority relation is complete. When this is the case, then priorities on formulas can be expressed by a function r from integers to integers. A goal base is then a finite set of formulas with an associated function: $GB = \langle\{G_1, \ldots, G_n\}, r\rangle$. If $r(i) = j$, then j is called the rank of the formula G_i. By convention, a lower rank means a higher priority. The question is now how to extend the priority on goals to a preference relation over alternatives. The following three choices are the most frequent ones:[6]

– *best-out ordering* [5]: $R \preceq^{bo}_{GB} R'$ iff
$\min\{r(i) \mid R \not\models G_i\} \leq \min\{r(i) \mid R' \not\models G_i\}$

– *discrimin ordering* [5, 14, 43]:
Let $d(R, R') = \min\{r(i) \mid R \not\models G_i \ \&\ R' \models G_i\}$.
$R \preceq^{dis}_{GB} R'$ iff $d(R, R') < d(R', R)$ or
$\{G_i \mid R \models G_i\} = \{G_i \mid R' \models G_i\}$

– *leximin ordering* [5, 58]:[7]
Let $d_k(R) = |\{G_i \mid R \models G_i \ \&\ r(i) = k\}|$.
$R \prec^{lex}_{GB} R'$ iff there exists a k such that
$d_k(R) < d_k(R')$ and $\forall j < k$, $d_j(R) = d_j(R')$
$R \preceq^{lex}_{GB} R'$ iff $R \prec^{lex}_{GB} R'$ or $\forall j$, $d_j(R) = d_j(R')$

Note that $\preceq^{lex}_{GB}$ and $\preceq^{bo}_{GB}$ are complete preference relations while $\preceq^{dis}_{GB}$ is generally not. We moreover have the following chain of implications: $R \prec^{bo}_{GB} R'$ entails $R \prec^{dis}_{GB} R'$ entails $R \prec^{lex}_{GB} R'$.

4.2.2 Ceteris Paribus Preferences

In this language, preferences are expressed in terms of statements like: "all other things being equal, I prefer these alternatives over those other ones." Formally, let C, G and G' be three propositional formulas and V a set of propositional variables including those occurring in G and G'. The *ceteris paribus* desire $C : G > G'[V]$ means: "when C is true, all irrelevant things being equal, I prefer $G \wedge \neg G'$ to $\neg G \wedge G'$", where the "irrelevant things" are the variables that are not in V. The preference relation induced by a set

[6] We are using the convention $\min(\emptyset) = +\infty$.

[7] Not to be confused with (although related to) the *leximin* ordering for the aggregation of individual preferences in a society of agents presented in Section 5.4.

of such preference statements is then the transitive closure of the union of preference relations induced by individual preference statements. This language can also be extended so as to allow for indifference statements.

An important sublanguage of *ceteris paribus* preferences is the language of (binary) *CP-nets* [9], which is obtained by imposing the following syntactical restrictions:

- Goals G and G' are literals speaking about the same propositional variable.
- The variables mentioned in the context C of a preference statement about variable p must belong to a fixed set, called the *parents* of p.
- For each variable p and each possible assignment π of the parents of p, there is one and only one preference statement $C : p > \neg p$ or $C : \neg p > p$ such that $\pi \models C$.

Various extensions of CP-nets have been proposed so as to be more expressive. For instance, TCP-nets [12] are CP-nets with a dominance relation between variables. Languages for *cardinal* preference representation in the style of CP-nets have been defined as well, for instance UCP-nets [8], which are based on generalised additive independence.

4.3 Discussion

At least five very important issues should be addressed when investigating preference representation languages:

- *Elicitation:* How hard is is to elicit preference from an agent so as to obtain a statement expressed in a given preference language L?
- *Cognitive relevance:* How close is a given language L to the way in which humans would express their preferences?
- *Expressive power:* Given a representation language L, a relevant question is whether L can express all preorders and/or all utility functions, or only complete preorders, or only a strict subclass of them, etc.
- *Computational complexity:* For a given language L, what is the computational complexity of comparing two alternatives, of deciding whether a given alternative is optimal, or of finding an optimal alternative?
- *Comparative succinctness:* Given two languages L and L', determine whether every preference structure that can be expressed in L can also be expressed in L' without a significant (that is, supra-polynomial) increase in size (in which case L' is said to be at least as succinct as L).

A detailed discussion of these issues in view of all the different representation languages we have covered would be beyond the scope of this survey. We limit ourselves to a few indicative remarks.

With the exception of bidding languages, all the languages for quantitative preferences presented above are fully expressive and we have already discussed several examples of comparative succinctness results for such languages. A problem with quantitative preferences in general is the well-known difficulty of eliciting numerical preferences from agents. *Ceteris paribus* preferences, being rather close to human intuition and comparatively easy to elicit, are interesting from a cognitive point of view. However, they have a high computational complexity in the general case, and furthermore, they generally leave many pairs of alternatives incomparable. As for prioritised goals, their lack of expressive power (no compensation allowed between goals) somewhat limits their range of use.

5 Social Welfare

A typical objective in MARA is to find an allocation that is optimal with respect to a metric that depends, in one way or another, on the preferences of the individual agents in the system. The aggregation of individual preferences can often be modelled using the notion of *social welfare* as studied in Welfare Economics and Social Choice Theory. This view is in line with the widely used metaphor of multiagent systems as "societies of agents". For instance, assuming that individual agents model their preferences using utility functions mapping bundles of resources to numerical values, the concept of *utilitarian social welfare*, defined as the sum of individual utilities, can be used to measure the quality of an allocation from the viewpoint of the system as a whole. This is probably

the most widely used interpretation of the term "social welfare" in the multiagent systems literature [79, 97].

In Welfare Economics and Social Choice Theory, on the other hand, many different *notions of social welfare* and related concepts have been studied [2, 65, 86] and many of these are also applicable to MARA systems [33]. In the context of an *e*-commerce application, our aim may be to maximise the average profit generated by the negotiating agents. In this case, utilitarian social welfare provides a suitable metric for assessing system performance. In an application such as that introduced in Section 2.2, where agents need to agree on the access to an Earth Observation Satellite that has been jointly funded by the owners of these agents, on the other hand, it is important that each agent receives a *fair* share of the common resource (possibly reflecting the size of the financial contribution made by its owner). In this case, average utility is clearly not a good indicator of performance.

Generally speaking, before sending a software agent into a system to negotiate on our behalf, we would like to know under what (social) rules that system operates. If these rules are not satisfactory, we may not be prepared to agree to be bound by the outcome of a negotiation.

In this section, we are going to review some of the notions of social welfare proposed in the literature on Welfare Economics and Social Choice Theory that are relevant to MARA. More specifically, we are going to present and discuss different approaches to defining a *social welfare ordering*, *i.e.* a mapping from the preferences of the agents in a society to the "preferences" of society as a whole. Good references in this area are the *Handbook of Social Choice and Welfare*, edited by Arrow, Sen and Suzumura [2], and the textbook by Moulin [65]. We are going to cover preference aggregation mechanisms for both ordinal and cardinal agent preferences (utility functions). Given that every utility function also induces an ordinal preference relation, any concept defined for ordinal preferences also extends to the cardinal case.

5.1 Notation

Let $\mathcal{A} = \{1, \ldots, n\}$ be a set of agents. Depending on whether we assume cardinal or ordinal preference structures, each of these agents i is equipped with either a utility function u_i or a preference relation $\preceq_i$. An allocation P is a mapping from agents to bundles of resources; that is, $P(i)$ is the bundle held by agent i in allocation P.

Our presentation is independent from the exact nature of the resources used (divisible or not, sharable or not, etc.). In most cases, we only assume that agents have preferences over alternative *allocations* (only in the case of envy-freeness, discussed in Section 5.7, we need to assume that agents have preferences over alternative *bundles*). For instance, $P \preceq_i Q$ states that agent i likes allocation P no more than allocation Q. Despite such generality, it makes sense to think of preferences as being defined over bundles of resources (as discussed in Section 4), *i.e.* to assume that there are no allocative externalities. That is, $P \preceq_i Q$ may be considered an abbreviation for $P(i) \preceq_i Q(i)$ and $u_i(P)$ is short for $u_i(P(i))$.

5.2 Pareto Optimality

An allocation P is *Pareto-dominated* by another allocation Q if and only if the following hold:

- $P \preceq_i Q$ for all agents $i \in \mathcal{A}$; and
- $P \prec_i Q$ for at least one agent $i \in \mathcal{A}$.

An allocation is *Pareto optimal* (or Pareto efficient) if and only if it is not Pareto-dominated by any other allocation. That is, an allocation is Pareto optimal if and only if it is not possible to (strictly) improve the individual welfare of an agent without making any of the others worse off.

Pareto optimality is generally regarded as the most fundamental criterion for efficiency. Note that the concept of Pareto optimality is purely ordinal: It does not require preferences to be numerical, not even interpersonally comparable. Also observe that the notion of Pareto dominance only gives rise to a partial (rather than a complete) ordering over alternative allocations.

5.3 Collective Utility Functions

If individual agents use utility functions to represent their preferences, then every allocation P gives rise to a utility vector $\langle u_1(P), \ldots, u_n(P)\rangle$. A *collective utility function* (CUF) is a mapping from such vectors to numerical values (e.g. the reals). Given that every allocation P determines a

utility vector, a CUF may also be regarded as a function from allocations P to numerical values. Every CUF sw induces a *social welfare ordering:* The allocation Q is socially preferred over allocation P if and only if $sw(P) \leq sw(Q)$.

In the sequel, we list several examples for such CUFs and indicate the kind of MARA applications where they may be useful.

5.3.1 Utilitarian Social Welfare

The *utilitarian social welfare* is defined as the sum of individual utilities:

$$sw_u(P) \quad = \quad \sum_{i \in \mathcal{A}} u_i(P) \tag{2}$$

The utilitarian CUF is independent of the zeros of individual utilities. It can provide a suitable metric for overall (as well as average) profit in a range of *e*-commerce applications.

5.3.2 Egalitarian Social Welfare

The *egalitarian social welfare* is given by the utility of the agent that is currently worst off:

$$sw_e(P) \quad = \quad \min\{u_i(P) \mid i \in \mathcal{A}\} \tag{3}$$

This CUF offers a level of *fairness* and may be a suitable performance indicator when we have to satisfy the minimum needs of a large number of customers. Fair division [13, 66, 101] is an important area with many potential applications in the field of MARA.

5.3.3 Nash Product

The *Nash product* is defined as the product of individual utilities:

$$sw_N(P) \quad = \quad \prod_{i \in \mathcal{A}} u_i(P) \tag{4}$$

This notion of social welfare favours both increases in overall utility and inequality-reducing redistributions. In this sense, it may be regarded as a good compromise between the utilitarian and the egalitarian agendas. Another interesting aspect of this CUF is that it is independent of the individual scales of agent utility functions.

Observe that the Nash product can only provide a meaningful metric of social welfare if all individual utilities are non-negative (or better even, if they are all positive).

5.3.4 Elitist Social Welfare

The *elitist social welfare* is given by the utility of the agent that is currently best off:

$$sw_{el}(P) \quad = \quad \max\{u_i(P) \mid i \in \mathcal{A}\} \tag{5}$$

The elitist CUF is clearly *not* a fair measure for social welfare, but it can be useful in cooperation-based applications where we require only one agent to achieve its goals.

5.3.5 Rank Dictators

The egalitarian and the elitist CUFs are both representatives of the family of *k-rank dictator* CUFs, which we are going to define next. Let $(v_P^{\uparrow})_k$ denote the kth smallest utility assigned to allocation P by any of the agents in $\mathcal{A}$ (this is the kth coordinate in the *ordered utility vector* for allocation P; see also Section 5.4). Then the k-rank dictator CUF sw_k is defined as follows:

$$sw_k(P) \quad = \quad (v_P^{\uparrow})_k \tag{6}$$

A special case of particular interest is the *median rank dictator* CUF which is defined as sw_k with $k = \frac{n}{2}$ in case n is even and $k = \frac{n+1}{2}$ in case n is odd. Indeed, for certain applications the individual level of welfare on an agent that does at least as well as half of the agents in the system but not better than the other half may be considered as suitable indicator for overall system performance.

5.4 The *leximin* Ordering

The *leximin* ordering is a social welfare ordering that refines egalitarian social welfare. It works by comparing first the utilities of the least satisfied agents, and in case these utilities coincide, compares the utilities of the next least satisfied agents, and so on. This idea is formalised as follows.

Suppose agents use utility functions to express their preferences. Then every allocation P gives rise to an *ordered utility vector* $v_P^{\uparrow}$, which is the result of first computing $u_i(P)$ for every agent $i \in \mathcal{A}$ and then arranging these values in ascending order. For example, $v_P^{\uparrow} = \langle 3, 5, 20 \rangle$ means that the agent worst off enjoys utility 3, the one best off utility 20, and the third one utility 5.

Then Q is *leximin*-preferred to P if and only if there exists an integer $k \in \{1, \ldots, n\}$ such that:

– $(v_P^{\uparrow})_i = (v_Q^{\uparrow})_i$ for all $i < k$; and

– $(v_P^{\uparrow})_k < (v_Q^{\uparrow})_k$.

In other words, the *leximin* ordering is the lexicographic ordering over ordered utility vectors. It favours the reduction of inequalities between agents. An allocation is *leximin-optimal* if and only if it is not *leximin*-preferred by any other allocation.

5.5 Generalisations

It is possible to build families of parametrised CUFs able to induce a continuous collection of social welfare orderings, including most of those defined above. Let us describe briefly two such families. The first one is defined by the following additive CUF [65]:

$$sw_{(p)}(P) \quad = \quad \sum_{i \in \mathcal{A}} g_{(p)}(u_i(P)) \tag{7}$$

The parameter p is a real number, $p \neq 0$, and $g_{(p)}(x) = \mathrm{sgn}(p) \cdot x^p$ (where $\mathrm{sgn}(p) = 1$ if $p > 0$ and $\mathrm{sgn}(p) = -1$ if $p < 0$), with the convention $g_{(0)}(u) = \log u$. Obviously, $sw_{(1)}$ measures utilitarian social welfare, and $sw_{(0)}$ induces the same social welfare ordering as the Nash product. The *leximin* ordering is the limit of the social welfare ordering induced by $sw_{(p)}$ as p goes to $-\infty$.

The other family of CUFs is a particular case of what is known as *ordered weighted averaging* (OWA) operators [99]. With the notation introduced above, let us define:

$$sw_w(P) \quad = \quad \sum_{i \in \mathcal{A}} w_i \cdot (v_P^{\uparrow})_i \tag{8}$$

Here, $w = (w_1, w_2, \ldots, w_n)$ is a vector of real numbers. Let us consider the vector w such that $w_i = 0$ for all $i \neq k$ and $w_k = 1$, then we have exactly the k-rank dictator CUF (including the egalitarian and the elitist CUFs, which are special cases of rank dictators). Consider now the vector w such that $w_i = \alpha^{i-1}$, with $\alpha > 0$, then the case $\alpha = 1$ corresponds to the utilitarian CUF, and the *leximin* ordering is the limit of the social welfare ordering induced by sw_w as α goes to 0.

5.6 Normalised Utility

It can often be necessary to *normalise* utility functions before aggregating individual preferences using any of the methods presented here, because many of them require individual utilities to be intercomparable. For instance, if P_0 is the initial allocation of resources, then we may restrict our attention to allocations P that Pareto-dominate P_0 and use the utility gains $u_i(P) - u_i(P_0)$ rather than the utilities $u_i(P)$ themselves as input to either a collective utility function or the *leximin* ordering.

A further normalisation step would be to evaluate an agent's utility gains *relative* to the gains it could expect in the best possible case. More precisely, let us define the maximum individual utility for each agent as:

$$\widehat{u}_i \quad = \quad \max\{u_i(P) \,|\, P \in Adm\} \tag{9}$$

Here, Adm is the set of admissible allocations. That is, $\widehat{u}_i$ is the utility that agent i could enjoy if it were the sole agent exploiting the available resources. Then we define the normalised individual utility of an agent i as follows:

$$u'_i(P) \quad = \quad \frac{u_i(P)}{\widehat{u}_i} \tag{10}$$

Observe that $\max\{u'_i(P) \,|\, P \in Adm\} = 1$, for all agents i. In other words, the maximum normalised utility is the same for all agents.

The optimum of the *leximin* ordering with respect to normalised utilities is known as the *Kalai-Smorodinsky* solution [66].

5.7 Envy-freeness

An allocation is *envy-free* if and only if each agent is at least as happy with its share than it would be with any of the bundles allocated to one of the other agents [13]. That is, an allocation P is envy-free if and only if $P(j) \preceq_i P(i)$ holds for all agents i and j. Envy-freeness is a property that does not require the intercomparability of the utilities of different agents.

If we require all items to be allocated, then an envy-free allocation does not always exist (consider, say, a an allocation problem with a single resource that is desired by all agents in the system). But even when not all items need to be allocated, it is well-known that there are allocation problems for which there exists no allocation that is both Pareto optimal and envy-free. One could therefore aim at finding (Pareto optimal) allocations that would, at least, minimise the overall "degree of envy" as much as possible. There

are several candidate definitions for *minimal envy*. Two possible approaches would be the following:

- Minimise the number of envious agents.
- Minimise the average degree of envy (the distance to the most envied competitor) of all envious agents.

5.8 Example

To exemplify some of the concepts introduced in this section, consider a scenario with two agents, 1 and 2, and a set of three resources $\{a, b, c\}$ that are indivisible and cannot be shared. Suppose the preferences of the two agents are represented by the utility functions u_1 and u_2:

$$\begin{array}{lll} u_1(\{a\}) = 18 & u_1(\{b\}) = 12 & u_1(\{c\}) = 8 \\ u_2(\{a\}) = 15 & u_2(\{b\}) = 8 & u_2(\{c\}) = 12 \end{array}$$

Furthermore, suppose u_1 and u_2 are *additive*, *i.e.* $u_i(R) = \sum_{r \in R} u_i(\{r\})$, and thereby fully specified by the above values. Let P be the allocation giving a to 1 and b and c to 2.

Allocation P has maximal egalitarian social welfare (18). Utilitarian social welfare, on the other hand, is not maximal for this allocation (38 rather than 42) , and neither is elitist social welfare (20 rather than 38).

P is Pareto optimal as well as *leximin*-optimal, but not envy-free, since agent 1 would be happier with the share of 2 than with its own. In fact, there is no allocation that would be both Pareto and and envy-free for this problem. On the other hand, for the slightly different problem where $u_1(\{a\}) = 20$ instead of 18 (leaving the rest unchanged), allocation P would be both Pareto optimal and envy-free.

5.9 Welfare Engineering

The insight that very different notions of social welfare may be appropriate for different applications of MARA has provided the impetus for the development of the *Welfare Engineering* framework [19, 33], which addresses two issues:

- the systematic choice of suitable social welfare orderings for a given application of MARA (and possibly the application-driven design of new orderings); and
- the design of appropriate rationality criteria and social interaction mechanisms for negotiating agents in view of different notions of social welfare.

By "appropriate" we mean criteria and mechanisms that ensure the convergence of the negotiation process to an allocation that is optimal with respect to the chosen social criterion (see also Section 6.4). Of course, depending on the application in question, such criteria need to be balanced with the autonomy requirements of individual agents.

An example for the first aspect of Welfare Engineering would be the *elitist* collective utility function discussed earlier, which seems unethical for human society, but it may be just the right performance indicator for a distributed computing application where several agents are working towards their own goals, but the system designer is only interested in (at least) one of them achieving their objective as quickly as possible. This aspect of Welfare Engineering may be characterised as "welfare economics for *artificial* agent societies".

An example for the second aspect would be the following convergence result: To achieve Pareto optimal outcomes in negotiation without monetary side payments, ask agents to negotiate mutually beneficial deals involving any number of agents or resources, but also to participate in deals that do at least not lower their own level of utility [35]. This aspect of Welfare Engineering can be summarised as "*inverse* welfare economics", alluding to the characterisation of mechanism design as "inverse game theory" [70].

6 Allocation Procedures

Generally speaking, the allocation procedure used to find a suitable allocation of resources could be either *centralised* or *distributed*. In the centralised case, a single entity decides on the final allocation of resources amongst agent, possibly after having elicited the preferences of the other agents in the system. Typical examples for the centralised approach are combinatorial auctions [24]. Here the central entity is the auctioneer and the reporting of preferences takes the form of bidding. In truly distributed approaches, on the other hand, allocations emerge as the result of a sequence of local negotiation steps. Such local negotiation is

often restricted to bilateral trading as in the classical *Contract-Net* approach [87], but systems allowing for multilateral exchanges of resources between more than two agents are also possible.

A comprehensive survey on allocation procedures for MARA would be beyond the scope of this paper. Any such survey would have to address at least the following three issues:

- *Protocols:* At this level, we need to address ontological issues (what types of deals are possible?) and devise communication protocols accordingly (what messages do agents have to exchange to agree on one such deal?).

- *Strategies:* When designing individual agents, we need to devise strategies for agents that allow them to best exploit a given negotiation protocol. This can also provide feedback to the first level: Where possible, protocols should be designed in such a way that they provide incentives to the negotiating agents to adopt a particular desirable profile of behaviour (mechanism design).

- *Algorithms:* At this level, we need to provide algorithms to solve the computational problems faced by agents when engaged in negotiation. This includes both algorithms to decide how to respond to a proposal in a distributed negotiation scenario and winner determination algorithms for combinatorial auctions. Again, this level may provide feedback to the other two levels: If a particular computational problem proves too hard to be solved in a reasonable amount of time then this may call for a simplification of the negotiation protocol (or strategy).

In this paper, we concentrate on the first of these issues. The most fundamental question to consider before devising a protocol for a MARA system is whether to adopt a centralised or a distributed design. We therefore start with a short discussion of the respective merits and drawbacks of centralised and distributed approaches to MARA. This is followed by an introduction to protocols for combinatorial auctions and an overview of the Contract-Net and related protocols for distributed resource allocation. Finally, we make a connection to our discussion of social welfare measures in Section 5 and review a number of results concerning the convergence to a socially optimal allocation for different protocols in the distributed setting.

6.1 Centralised vs. Distributed

Both the centralised and the distributed approach to MARA have their advantages and disadvantages. Possibly the most important argument in favour of auction-based mechanisms concerns the simplicity of the communication protocols required to implement such mechanisms. Another reason for the popularity of centralised mechanisms is the recent push in the design of powerful algorithms for combinatorial auctions that, for the first time, perform reasonably well in practice [41, 80]. Of course, such techniques are, in principle, also applicable in the distributed case, but research in this area has not yet reached the same level of maturity as for combinatorial auctions. An important argument *against* centralised approaches is that it may be difficult to find an agent that could assume the role of an "auctioneer" (for instance, in view of its computational capabilities or in view of its trustworthiness).

The distributed model seems also more natural in cases where finding optimal allocations may be (computationally) infeasible, but even small improvements over the initial allocation of resources would be considered a success. Step-wise improvements over the *status quo* are naturally modelled in a distributed negotiation framework.

6.2 Auction Protocols

Auctions [24, 54, 55, 94, 98] are centralised mechanisms for the allocation of goods amongst several agents. Agents report their preferences and wait for the final allocation to be made by the auctioneer (whether there is an initial allocation of goods, as in combinatorial exchanges, or not, as in regular combinatorial auctions). The act of reporting preferences is called *bidding* and, naturally, agents are not required to reveal their true preferences during bidding, but they may submit whatever bid(s) they believe to best serve their own interests.

Bidding may be public (*open-cry*) as in the well-known English auction model or private (*sealed bids*). In the case of open-cry bidding,

we can further distinguish between *ascending* bids (English auction) and *descending* bids (Dutch auction) [94]. In combinatorial domains, which is what we are interested in here (*i.e.* there are many goods and agents can submit bids for different combinations of goods), typically, most auction protocols foresee only a single round of bidding using sealed bids. The *bidding language* (see Section 4.1.5) determines what types of bids are admissible (and how to interpret them).

The auction protocol also specifies which agent would be awarded which goods, based on the bids received in time, and what price they should pay for the bundles allocated to them. In some cases, this decision can be left entirely to the auctioneer (who will seek to maximise her revenue). In other cases, it is important that the auctioneer follows the rules specified by the protocol, as these rules have been designed in such a way as to provide incentives to the bidders to bid truthfully. This is the case for the Vickrey auction model [94], and its extensions to combinatorial scenarios, where the winning agents pay less then the prices they specified in their bids.

For an extensive review of different auction models for resource allocation in combinatorial domains we refer to the forthcoming book on *Combinatorial Auctions*, edited by Cramton, Shoham and Steinberg [24], and the review article on the same topic by Kalagnanam and Parkes [54].

6.3 Negotiation Protocols

We now give a brief overview of some of the protocols developed for negotiation over resources in a distributed setting.

6.3.1 Contract-Net

Perhaps the most popular negotiation protocol is the *Contract-Net* protocol [87]. Although the protocol was primarily designed for *task* allocation, it is also perfectly suited to MARA. The protocol consists in four interaction phases, involving two roles (manager and bidder):

- *Announcement* phase: The manager advertises the resource to a number of partner agents (the bidders).
- *Bidding* phase: The bidders send their proposals to the manager.
- *Assignment* phase: The manager elects the best bid and assign the resource accordingly.
- *Confirmation* phase: The elected bidder confirms its intention to obtain the resource.

Any agent can initiate an interaction following the protocol by assuming the adequate role. The protocol is really a one-to-many protocol, leading to the assignment of a single task (or resource) to a single contractor (that is, the resulting deal is a one-to-one agreement regarding a single item).

6.3.2 Extensions

Many different extensions to this protocol have been proposed and we briefly review some of these here. The TRACONET system developed by Sandholm [77], for instance, uses a variant of the classical Contract-Net protocol to allow negotiation over the exchange of *bundles* of resources.

Golfarelli et al. [46] have proposed an extension where the bidders have no explicit mechanism for utility transfer (in other words, they cannot use money). The first phase remains the same as in the original Contract-Net: the manager announces a (bundle of) resource(s). But the protocol is based on exchanges: instead of bidding money, the agents will bid for one or more resources they are interested in exchanging. This extension allows agents to agree on *swapping* resources (rather than buying them from each other).

Sousa et al. [89] have designed a version of the Contract-Net protocol where bidders first propagate *constraints* between them in order to guarantee the coherence of different operations related to the same task.

6.3.3 Concurrent Contract-Net

As pointed out by Aknine et al. [1], when many managers negotiate simultaneously with many contractors, using the Contract Net protocol can lead to unsatisfactory results. In particular, because contractors are required to answer a single bid at a time, they may miss some contracts. To overcome this, they have proposed an extension to in which a pre-bidding and a pre-assignment

phase are added before the final bidding and assignment phase of classical Contract-Net protocols. During the pre-bidding and pre-assignment phases, which can last a long time, agents propose temporary bids and managers temporarily accept (or reject) these bids. These new phases have several positive effects:

- After a deal has been temporarily accepted, if the manager receives a better offer, this deal can be turned into temporarily rejected offer. It turns out that when many negotiations are conducted simultaneously, by delaying the final acceptance, better deals (from the manager's point of view) may be negotiated.
- Contractors can modify their offers many times by making temporary offers. If the contractor receives a better new offer from another manager, it can modify its temporary bids before sending a definitive bid.
- The pre-bidding phase may be quite long. This has the positive effect of reducing the risk of decommitment.

An alternative way to tackle this latter problem is to allow agents to decommit, but to apply penalties when they do so. This route has been followed in the *levelled commitment* approach proposed by Sandholm and Lesser [82].

6.4 Convergence Properties

As discussed earlier, once a particular negotiation protocol has been fixed, we need to devise strategies for the agents using that protocol. Work in this area is often of a game-theoretical nature. A different line of research has analysed how the negotiation behaviour of individual agents affects the quality of the overall distribution of resources (with respect to some of the social welfare measures introduced in Section 5) by abstracting away from the details of individual negotiation strategies [35, 78].

For instance, a *rational* agent may be defined as an agent that will only agree to deals that result in a positive payoff for itself. That is, a set of rational agents will only agree on mutually beneficial deals. Which of the possibly many mutually beneficial deals agents will actually agree on depends on the concrete strategies they use, and overly aggressive negotiation strategies may even prevent agents from identifying any mutually beneficial deal at all [67]. However, in cases where it is admissible to assume that agents will agree on *some* deal meeting certain rationality criteria (such as resulting in a strictly positive payoff for everyone involved) whenever such a deal exists, it is sometimes possible to prove so-called *convergence properties* of a negotiation framework.

For instance, in the context of negotiation over finitely many indivisible resources, an important result, due to Sandholm [78], states that *any* sequence of deals that are mutually beneficial will eventually result in an allocation with maximal utilitarian social welfare, provided that agents can use monetary side payments to compensate their trading partners for otherwise disadvantageous deals (and each agent's payoff is linear in the amount of money received). That is, there can be no infinite sequence of mutually beneficial deals, and if agents keep on making such deals the system will converge to an allocation that maximises the sum of individual utilities. A similar result states that any sequence of mutually beneficial deals without side payments will converge to a Pareto optimal allocation [35].

An important caveat is that these results apply to negotiation settings where agents can agree on truly *multilateral* deals: A single deal may involve any number of agents (as well as any number of resources). Decomposing such a multilateral deal into a sequence of bilateral deals is not always possible, because some of the bilateral deals making up the overall deal may not be mutually beneficial to both agents. Hence, myopic agents that require a positive payoff for every single deal they take part in will not accept such a deal.

Given the difficulty of implementing such general deals, it is important to understand under what circumstances sequences of structurally simple deals suffice to guarantee convergence to a socially optimal allocation of resources. Recent results in this area show that mutually beneficial deals with side payments that involve only a single resource each (and thereby only two agents at a time) suffice to reach allocations with maximal utilitarian social welfare in case all agents use *modular* utility functions [35].[8] In fact, the class

[8] A utility function u is said to be *modular* if and only if we have $u(R_1 \cup R_2) = u(R_1) + u(R_2) - u(R_1 \cap R_2)$ for all bundles R_1 and R_2. This means that the utility assigned

of modular utility functions is also *maximal* in the sense that for no class of functions strictly including that class it would still be possible to guarantee that agents using utility functions from this larger class and negotiating only mutually beneficial deals over single resources will eventually reach an allocation with maximal utilitarian social welfare in all cases [21]. Related work has also identified classes of utility functions (and ordinal preference relations) that guarantee the convergence to optimal allocations for sequences of deals involving at most k resources each [20].

7 Complexity Results

A growing body of work within the study of MARA considers various concepts of complexity, not only in the standard sense of *computational complexity theory* but also in terms of concepts such as *communication complexity*. Such work comprises both *positive* results—e.g. algorithms with provably efficient performance characteristics, properties of restricted classes of allocation settings, etc.—and a large collection of *negative* results that suggest many naturally arising decision and optimisation problems are unlikely to admit generally applicable algorithmic solutions. Within this section our aim is to review extant work that has addressed such questions and to catalogue related open problems.

7.1 Models and Assumptions

The structure we consider in the subsequent text will be referred to as a *resource allocation setting*, by which we mean a triple $\langle \mathcal{A}, \mathcal{R}, \mathcal{U} \rangle$ where:

- $\mathcal{A} = \{1, 2 \ldots, n\}$ is a set of n agents;
- $\mathcal{R} = \{r_1, r_2, \ldots, r_m\}$ is a collection of m resources; and
- $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$ describes the utility function $u_i : 2^{\mathcal{R}} \to \mathbb{Q}$ for the agent $i \in \mathcal{A}$.

We assume that each $r \in \mathcal{R}$ is *indivisible* and *non-shareable*, *i.e.* at most one agent at a time will "own" r (see also Section 3). An allocation of the resources in $\mathcal{R}$ among the agents in $\mathcal{A}$ is a mapping $P : \mathcal{A} \to 2^{\mathcal{R}}$ with $P(i) \cap P(j) = \emptyset$ for any $i \neq j$. The set of all allocations of $\mathcal{R}$ among $\mathcal{A}$ will be denoted by $\Pi_{n,m}$. From the fact that there are n choices of agent for each of the m resources, it is easily seen that $|\Pi_{n,m}| = n^m$.

7.2 Computational vs. Communication Complexity

In very informal terms, traditional computational complexity theory is concerned with the issue of classifying computational problems with respect to how much of a particular computational resource is required for their solution. Typically, *computational problems* are phrased as *decision questions*, *i.e.* given an input instance I, is it the case that some property ϕ holds true of I? For example, given a directed graph $H(V, E)$ and a vertex s in V, is it the case that every vertex in V can be reached by some path that starts in s? The concept of *computational resource* is modelled via some formal *model of computation*. Thus, *time* (*space*) as the (worst-case) number of *moves* (*tape cells*) made by a (deterministic) 2-tape Turing machine (DTM) that correctly classifies input instances, *i.e.* accepts if $\phi(I) = \top$, rejects if $\phi(I) = \bot$. For further introductions to computational complexity theory we refer the reader to the textbook by Papadimitriou [69].

In the context of MARA problems, computational complexity results have tended to address what might be termed *"global"* properties of given resource allocation settings, e.g. whether allocations satisfying particular criteria exist. Recent work, however, has begun to address computational properties of abstract high-level *negotiation protocols* as reviewed in Section 6.4 above, e.g. given some constraint, χ, that allowed deals must satisfy, a number of decision problems may be formulated regarding allocations that are reachable from a starting allocation via sequences of χ-*deals*.

This view of complexity has not, in general, needed to be concerned with *"localised"* questions, e.g. the overheads involved in describing and implementing proposed deals; how many deals may be needed in order to reach an allocation with desirable properties, etc. In the work of Endriss and Maudet [34] the term *communication complexity*, deriving from the model put forward by Yao [100], is introduced to capture the

to a bundle of resource can be computed as the sum of the utilities of the individual resources in that bundle, *i.e.* the classes of modular and 1-additive functions coincide.

combination of *number of deals* and *communication to agree a deal* that could be needed in order for an allocation to be finalised. While the bulk of the survey below is concerned with complexity issues from the perspective of computational complexity, we also discuss some results related to communication from the works of Endriss and Maudet [34] and Dunne [29], that consider upper and lower bounds on the *number of deals* needed in various contexts.

7.3 Allocations with Given Properties

Given a resource allocation setting, $\langle \mathcal{A}, \mathcal{R}, \mathcal{U} \rangle$, the agents concerned seek to bring about an allocation that will satisfy certain criteria. As discussed in Section 5, such criteria may be purely quantitative (e.g. the sum of the individual utility valuations (utilitarian social welfare) is maximal (or is above a given amount), but so-called qualitative properties (Pareto-optimal or envy-free outcomes, for instance) are also of interest.

7.3.1 Representation Issues

Standard computational complexity theory considers properties of algorithms implemented within some "well-defined" model of computation, e.g. Turing machines. In order sensibly to consider the performance of a specific algorithm, this is reported as a function of the algorithm's *input length*. This convention presumes that, in comparing different algorithmic approaches to a particular problem, such comparisons are only "reasonable" if the representation of input instances is similar, or that (at worst) different formats can be translated between efficiently.

In considering how instances are to be represented in the case of decision problems concerning resource allocation settings, a significant issue that arises is the encoding of the collection of utility functions $\mathcal{U}$. The domain of a utility function is $2^{\mathcal{R}}$: thus (from the viewpoint of upper bounds on complexity) the characteristics of algorithms employing an enumerative form (listing all subset/value pairs) may not be comparable with algorithms employing some *compact* representation. We therefore give complexity results for the three different forms of representing utility functions discussed in Section 4.1: the bundle form, the SLP form, and the k-additive form (here the 2-additive form is of particular interest).

7.3.2 Quantitative Criteria

Two natural decision questions regarding the measure sw_u of utilitarian social welfare, have been considered with regard to each of the three formalisms for representing utility functions:

Welfare Optimisation (WO)	
Instance:	$\langle \mathcal{A}, \mathcal{R}, \mathcal{U} \rangle$; $K \in \mathbb{Q}$
Question:	$\exists\, P \in \Pi_{n,m} : sw_u(P) \geq K$?

Welfare Improvement (WI)	
Instance:	$\langle \mathcal{A}, \mathcal{R}, \mathcal{U} \rangle$; $P \in \Pi_{n,m}$
Question:	$\exists\, Q \in \Pi_{n,m} : sw_u(Q) > sw_u(P)$?

WI and WO are both NP-complete for the representation of utility functions in bundle form (reduction from Set Packing [18]); for the SLP form (reduction from 3-Sat [32]); and for 2-additive functions (the simplest proof is via a reduction from Max-2-Sat [18]). Both the 2-additive and SLP results apply even in systems containing only 2 agents; the SLP reduction shows that the problems remain NP-complete when (both) utility functions are monotonic.

7.3.3 Qualitative Criteria

The qualitative measures of Pareto optimality and envy-freeness give rise to the following decision problems:

Pareto Optimality (PO)	
Instance:	$\langle \mathcal{A}, \mathcal{R}, \mathcal{U} \rangle$; $P \in \Pi_{n,m}$
Question:	Is P Pareto optimal?

Envy-Freeness (EF)	
Instance:	$\langle \mathcal{A}, \mathcal{R}, \mathcal{U} \rangle$
Question:	$\exists\, P \in \Pi_{n,m} : P$ is envy-free?

Deciding PO is coNP-complete for both SLP and 2-additive utility functions. The former was shown by Dunne et al. [32] (reduction from 3-UnSat restricted to instances with clause and variable numbers equal); the latter, although not explicitly stated by Chevaleyre et al. [18], is an immediate consequence of their proof that WI is NP-complete. Again both continue to hold in 2-agent contexts, with the SLP reduction also applying to monotonic utility functions.

EF is examined in a variety of cases in the work of Bouveret and Lang [11]. They consider a representation based on concise logic-based descriptions of agent preferences (as discussed in Section 4.1.3 above). In addition to the basic question of whether envy-free allocations are possible—shown to be NP-complete even within 2-agent settings—the question of allocations that combine envy-freeness with Pareto optimality is examined (termed *efficient envy-free*, or EEF, allocations). For such decision problems they demonstrate completeness results ranging from NP-complete up to Σ_2^p-complete, depending on the restrictions placed on the preference relations. That NP-completeness also holds for the question EF within the SLP model in 2-agent settings has been shown by Dunne [30] (reduction from 3-SAT).

7.4 Path and Convergence Properties

The collection of results referred to above, hold independently of the regime used to negotiate allocations. There are, however, a number of questions that arise specifically in the context of distributed negotiation when the structure of admissible deals is constrained. Thus suppose that only *individually rational* deals may be used, *i.e.* deals that are beneficial to all the agents involved. If monetary side payments are allowed, then individually rational deals are deals $\langle P, Q\rangle$ under which $sw_u(Q) > sw_u(P)$ [35]. As has been shown by Sandholm [78], if additional constraints, such as "all deals are bilateral and involve exactly one resource changing" (sometimes called the class of O-contracts), then there are cases where some rational deals cannot be implemented. A further problem that arises is that, even when there is a *rational O-contract path* to go from P to Q this may involve an agent repeatedly making deals involving the same resource, *i.e.* such paths may contain more than m distinct deals.

In general, given some predicate Φ on deals, the following decision problem arises:

Φ-Path	
Instance:	$\langle \mathcal{A}, \mathcal{R}, \mathcal{U}\rangle$; $P^{(s)}, P^{(t)} \in \Pi_{n,m}$ with $sw_u(P^{(t)}) > sw_u(P^{(s)})$
Question:	$\exists\, \Delta = \langle P^{(0)}, P^{(1)}, \ldots, P^{(r)}\rangle$ s.t. $P^{(0)} = P^{(s)}$ and $P^{(r)} = P^{(t)}$ and $\forall 1 \leq i \leq r$, $\Phi(P^{(i-1)}, P^{(i)})$?

Dunne et al. [32] consider the complexity of Φ-Path where $\Phi(P, Q)$ holds only if the deal is individually rational and involves *at most* some given number k of the resources being passed from one agent to another. Within 2-agent settings using SLP representation it is shown that Φ-Path is NP-hard for all $k \leq \frac{m}{3}$ (and for the case $k = \frac{m}{2}$). In the special case of O-contracts (*i.e.* $k = 1$) NP-hardness holds with both utility functions being monotonic. Recent work, presented in Dunne and Chevaleyre [31], improves this NP-hardness lower bound and obtains an exact complexity classification: Φ-Path is PSPACE-complete for $\Phi(P, Q)$ holding if the deal is an individually rational O-contract.

Introducing the idea of a *maximal* Φ-path (from an initial allocation P) as a sequence of deals every one of which satisfies Φ and with the final allocation, $P^{(t)}$ being such that for every Q it is the case that $\neg\Phi(P^{(t)}, Q)$, leads to the following related problem:

Φ-Convergence	
Instance:	$\langle \mathcal{A}, \mathcal{R}, \mathcal{U}\rangle$
Question:	Is it the case that $\forall P \in \Pi_{n,m}$, for all maximal Φ-paths Δ starting from P, the allocation $last(\Delta)$ these terminate in, is one which maximises sw_u?

For instance, the basic convergence result first proved by Sandholm [78] (discussed in Section 6.4) shows that the answer to the above question is *always* "yes" when Φ does not restrict the range of admissible deals in any way. Φ-Convergence is the subject of ongoing work which has already established the following: For Φ corresponding to individually rational O-contracts, Φ-Convergence is coNP-complete for the SLP model and for 4-additive utility functions [31] Both results hold in 2-agent settings. If all utility functions are modular (*i.e.* 1-additive), then the answer to Φ-Convergence is always "yes" [21].

We now return to an issue relating to the ideas of *communication complexity* discussed above. The question of interest also has a bearing on establishing *upper bounds* on the complexity of Φ-Path. Given a resource allocation setting and Φ, consider the (rational) deals that *can* be implemented by Φ-paths. Dunne [29] has introduced the following measures:

- $L^{opt}(P,Q)$: the length of the *shortest* Φ-path realising $\langle P,Q\rangle$.
- $L^{\max}(\mathcal{A},\mathcal{R},\mathcal{U})$: the maximum value of $L^{opt}(P,Q)$ over those deals for which a Φ-path exists.
- $\rho^{\max}(n,m)$: The maximum value (taken over all choices of utility function) of $L^{\max}(\mathcal{A},\mathcal{R},\mathcal{U})$.
- $\rho^{\max}_{C}(n,m)$: As $\rho^{\max}$, but with the maximisation taken over utility functions belonging to some class C.

A related study (employing different terminology) is given by Endriss and Maudet [34], where attention is focussed on utility functions which allow any rational deal to be implemented via some sequence of rational O-contracts; the main case being considered in this respect is that of 1-additive functions.

Let $\Phi(P,Q)$ hold if and and only if the deal $\langle P,Q\rangle$ is an individually rational O-contract:

- $\rho^{\max}(n,m) \leq n^m - m(n-1)$ [78]
- $\rho^{\max}(n,m) \geq \frac{77}{256}2^m - 1$ [29]
- $\rho^{\max}_{1-add}(n,m) = m$ [34]
- $\rho^{\max}_{mono}(n,m) \geq \frac{77}{128}2^{\frac{m}{2}} - 3$ [29]

The latter two results pertain to the classes of 1-additive and monotonic utility functions, respectively. The constructed rational paths in the general and monotonic lower bound cases are unique.

7.5 Open Problems and Conjectures

Given the existing results concerning the measure sw_u wherein exact complexity classifications have been derived for each of the three representation styles for utility functions, the following conjectures seem plausible and ought to be straightforward to verify.

Conjecture 1 *Deciding if there is an allocation P with $sw_e(P) \geq K$ is NP-complete (whether $\mathcal{U}$ is given in bundle form, SLP form, or is 2-additive).*

Conjecture 2 *Given $K \in \mathbb{Q}$, deciding if $\max\{sw_u(P) \mid P \in \Pi_{n,m}\} = K$ is D^P-complete (again in all three representation formalisms).*

Conjecture 3 *EF is NP-complete for 2-additive utility functions.*

8 Simulation Platforms

Theoretical work in Microeconomics and Auction Theory provides a very strong foundation for analysing many resource allocation problems. However, on occasion we may be faced with a problem in which some of the assumptions underlying the theory are violated. This is especially the case in MARA scenarios where computational concerns are prominent [25]. For example, mechanism design as originally developed in Economics is not concerned with computational issues such as algorithmic or communication complexity. In a conventional auction design scenario issues such as the speed of winner determination and the communication costs of submitting bids are often not of significant concern since they are not typically a bottleneck with respect to the entire auction process which can involve protracted and lengthy decision making by human traders. However, in a market place run entirely by automated trading agents, such issues are likely to be of more concern since their performance costs can sometimes be of similar order of magnitude as the overall computational costs of running the auction. Once these costs are taken into account many of the results in auction theory become somewhat brittle. For example, the revelation principle no longer applies when transaction-throughput and reduction in communication complexity are adopted as design goals [74].

In such cases experimental work using simulations of agent-based market places—*Agent-based Computational Economics* (ACE) [93]—can shed light on some of the grey areas that are difficult to analyse using existing theoretical tools.

As with any software engineering problem, in choosing an appropriate software framework in which to implement an ACE simulation it is important to consider the requirements that the software needs to meet. In this section, we give an overview of the typical requirements addressed by ACE software, and we then proceed to give an overview of some commonly-used simulation frameworks.

8.1 Simulation vs. Implementation

Software for *simulating* multiagent systems typically addresses different requirements from that designed to *implement* multiagent systems. Al-

though it is natural to view a MAS implementation as its own simulation, there are a number of problems with such an approach, which we shall address in turn.

Firstly, ideally we would like the outcome of a simulation experiment to be exactly reproducible given the initial conditions of the experiment. This is not always possible in a MAS implementation since many environmental factors will be beyond the experimenter's control. For example, the precise outcome of an experiment may depend on the exact timing with which an agent responds to a particular message, and this time interval will depend on factors beyond the experimenter's control, such as the memory and CPU currently available to the agent.

Secondly, when we come to analyse the results of a simulation, we often need to generalise beyond a single run of an experiment with a single set of initial conditions. Typically, we generalise by taking many samples of free initial variables and running the experiment many times for each sample. Simulation frameworks are equipped to log data from the outcome of each experiment to a format suitable for analysis using statistical analysis software, such as MATLAB.

Thirdly, the performance considerations of a simulation are qualitatively different to that of an implementation. The software architecture of a MAS implementation is driven by real-world requirements that do not always hold in a simulation context. For example, trading agents need to be able to run on different machines due to commercial and practical considerations. This distributed parallelism is detrimental to raw system-level performance however, since inter-host network communication overheads dominate other performance considerations. By running all agents on the same host we can achieve several orders of magnitude performance increase. This would be an impractical solution for a real MAS trading implementation. However, such considerations do not apply in a simulation context, and by relaxing these constraints we can achieve a significant gain in performance.

Similarly, much of the technical complexity of a real MAS implementation addresses requirements that are not present in a simulation context. For instance, MAS implementations need to be robust against system failures, and they need to respond quickly to real-time asynchronous events. This necessitates a highly parallel software architecture, involving, for example, many threads of execution running simultaneously. Such considerations do not apply in agent-based simulation, since real-time parallelism can be simulated using a sequential program, and this greatly reduces the complexity of the software (and hence the potential for bugs).

Finally, any MAS interacts at some point with the environment. In a MARA scenario, for example, the environment might constitute economically relevant characteristics of the human owners of agents, such as their utility functions. Unlike the agents in a MAS implementation, the environment is not a software entity in a MAS implementation, and cannot be directly ported to an agent-based simulation. Rather, the environment itself must be simulated. Agent-based simulation toolkits allow for the abstract statistical simulation of environmental factors. Hence a key feature of any simulation toolkit is a library of pseudo-random number generators (PRNGs). A good simulation toolkit will provide high quality PRNGs, such as the Mersenne Twister PRNG [64], with extremely large periods, low statistical correlation, and the ability to produce random numbers according to arbitrary (non-uniform) distributions.

In summary, when developing a system to simulate a MARA scenario, it is important to choose a framework or toolkit that is specifically designed for agent-based *simulation*, as opposed to toolkits such as JADE [51] that are designed for *implementing* multiagent systems.

8.2 Simulating Time

For practical purposes we often prefer to simulate the parallelism of events using sequential computation, rather than execute the simulation of multiple simultaneous events in parallel in real-time. This necessitates a framework for computing the outcome of events that occur simultaneously. There are several approaches to simulating time in a model.

8.2.1 Continuous Time Models

Many physical processes are characterised by smooth and continuous changes in time-dependent variables. Differential equation models

are common in analytical microeconomics. Such models are applicable approximations of real market places when there are very large numbers of participants in the market since individual characteristics of the participants play a less significant role and the entities in the system can be treated as simple and homogeneous particles. However, these models break down when the number of participants becomes very small and the individual and strategic characteristics of the participants become more prominent.

Agent-based models address this issue by providing a richer structure with which to model market participants. In such models, macro-level variables describing the ensemble of agents no longer vary smoothly with time. This necessitates alternative approaches to temporal modelling.

8.2.2 Discrete-event Simulation

Discrete-event simulation frameworks [4, 42] model time in discrete quanta called *ticks*. Intuitively, a tick can be thought of as an "instant" of time. During the simulation of a tick—the *tick cycle*—entities (agents) in the simulation signal which agents they interact with during that instant of time by sending *events* to each other. Individual events specify the exact nature of the interaction between agents. In an auction simulation, for example, an auctioneer agent may send an *end-of-auction* event to all trading agents in the auction when it has closed. At the end of a tick cycle, once events have been exchanged, each entity updates its internal state in response to any events it has received.

8.3 Agent Modelling

In a MARA simulation, agents often need to make intelligent decisions in their resource utilisation and acquisition behaviour. The intelligent agents community has traditionally favoured symbolic approaches, such as the class of BDI (Belief-Desire-Intention) models. In a MARA scenario, however, an agent's goals are often quantitative in nature; for example, agents act to maximise their expected utility. In the field of agent-based electronic commerce, this has led to the adoption of Bayesian approaches to agent's decision problems such as (multiagent) reinforcement learning.

Many agent-based simulation frameworks have been developed by the Artificial Life (ALife) community. Agents in ALife models are imbued with very little intelligent behaviour at the outset; rather, intelligent behaviour emerges collectively from the complex interactions between agents equipped with relatively crude decision making machinery. Connectionist approaches such as neural networks and evolutionary approaches such as genetic algorithms, are popular in such models.

Since simulation is the main methodology used in ALife research, ALife software toolkits tend to be the most mature in terms of simulation functionality. Correspondingly, since empirical methods are relatively rare in MAS research, there are few frameworks for *simulating* BDI agents, as opposed to implementing BDI agents.

8.4 Extensibility and Integration

When conducting research via simulation it is often necessary to *extend* the existing functionality of the system. Although all frameworks provide the ability to configure simulations, the desired behaviour cannot always be implemented by configuring the existing components provided by the framework. In this case it is necessary for the researcher to implement the desired behaviour by writing their own code. Toolkits take two main approaches to allowing extensibility: They allow either for scripting in a custom language or for the introduction of new classes and methods via inheritance.

8.5 Software Listing

We are now going to give a brief overview of some commonly used general-purpose simulation frameworks that might be suitable for analysing MARA problems.

8.5.1 Swarm

Swarm [91] is one of the most famous ALife software toolkits and has been continually improved by an active community of users and developers since the early 1990s. It provides an API for discrete-event simulation, uses high-quality PRNGs, allows for spatial modelling, and includes real-time visualisation tools. Swarm is an open-source project written in the Objective-C programming language.

8.5.2 Extensions to Swarm

The Evo toolkit [36] is an extension to Swarm that provides agents with the ability to mate and evolve new behaviour over time using a system similar to genetic programming.

MAML [63] is an extension to Swarm that provides a higher-level scripting language that is simpler to use than Objective-C. The goal is to allow researchers from the social sciences, who are not necessarily skilled programmers, to quickly develop simulations.

8.5.3 RePast

RePast [75] is another toolkit inspired by Swarm, but is written entirely in Java, and the ultimate design goals of this system are more MAS- rather than ALife-oriented. It offers similar features as Swarm (discrete-event simulation, high-quality PRNGs, spatial modelling, visualisation tools) and it is also open-source and extensible.

The core simulation functionality of RePast is particularly mature and robust (it use the COLT library for high-performance scientific computing). The MAS-oriented features, on the other hand, are still relatively immature (no explicit reinforcement learning, no BDI support).

8.5.4 Desmo-J

Desmo-J [26] is implemented in Java and provides raw discrete-event simulation functionality. While only providing minimal functionality, the system comes with a highly flexible, well-designed API. It uses the standard Java PRNG, but the API should allow other (more advanced) PRNGs to be plugged in as well.

8.5.5 AScape

AScape [3] is a Java-based discrete-event simulation framework with an emphasis on spatial modelling of agents.

8.5.6 DEx

DEx [27] is a high-performance toolkit providing high-quality PRNGs, discrete-event simulation, spatial modelling, and real-time visualisation tools (including 3D representation).

9 Conclusion

We have presented a survey of salient issues in Multiagent Resource Allocation (MARA), a timely and fast-developing area of research at the interface of Computer Science and Economics. Naturally, the choice of topics selected for detailed presentation has been driven, at least in part, by personal interests and preferences. Nevertheless, we are confident that this material will prove useful to many researchers working on different aspects of MARA and related disciplines.

In the first part of the paper, after a short introduction to the field, we have highlighted four major application domains, which together both demonstrate the wide scope of MARA and underline the urgent need to further advance the field to meet the enormous challenges still posed by these applications. The second part of the paper serves as a catalogue of fundamental concepts in MARA: generic properties of resources characterising a MARA problem at hand; languages for preference representation to model the interests of individual agents; and social welfare measures and related tools to assess the overall quality of an allocation of resources. The third part of the paper then addresses actual MARA techniques. This includes, in particular, an introduction to allocation procedures and a selection of relevant complexity results. Where theoretical results alone are not sufficient, our survey of simulation platforms can serve as a starting point for experimental work.

Two important issues that we have *not* covered are the *algorithmics* of MARA and the *game-theoretical analysis* of negotiation (and bidding) strategies. The former includes the design of algorithms for the winner determination problem in combinatorial auctions, and a survey of recent work in this area is available elsewhere [81]. The literature on game-theoretical issues in negotiation, multiagent systems, and Computer Science in general is vast and fast-developing. A good starting point for readers interested in the computational approach to Game Theory (and the game-theoretic approach to Computer Science) is the short paper by Papadimitriou [70].

Acknowledgements. This survey has been written in the context of the activities of the AgentLink Technical Forum Group on Multiagent Resource Allocation (TFG-MARA).

References

[1] S. Aknine, S. Pinson, and M. F. Shakun. An extended multi-agent negotiation protocol. *Journal of Autonomous Agents and Multi-Agent Systems*, 8(1):5–45, 2004.

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

[3] AScape System. `http://www.brook.edu/es/dynamics/models/ascape/main.htm`.

[4] J. Banks, J. Carson, B. Nelson, and D. Nicol. *Discrete-event System Simulation*. Prentice Hall, 4th edition, 2005.

[5] S. Benferhat, C. Cayrol, D. Dubois, J. Lang, and H. Prade. Inconsistency management and prioritized syntax-based entailment. In *Proc. 13th International Joint Conference on Artificial Intelligence (IJCAI-1993)*, pages 640–647. Morgan Kaufmann, 1993.

[6] M. Bichler, J. Kalagnanam, and H. S. Lee. RECO: Representation and evaluation of configurable offers. In H. K. Bhargava and N. Ye, editors, *Computational Modeling and Problem Solving in the Networked World: Interfaces in Computing and Optimization*. Kluwer, 2003.

[7] R. A. Bourne and R. Zaidi. A quote-driven automated market. In *Proc. AISB Symposium on Information Agents for E-Commerce*. AISB, 2001.

[8] C. Boutilier, F. Bacchus, and R. I. Brafman. UCP-networks: A directed graphical representation of conditional utilities. In *Proc. 17th Annual Conference on Uncertainty in Artificial Intelligence (UAI-2001)*, pages 56–64. Morgan Kaufmann, 2001.

[9] C. Boutilier, R. I. Brafman, C. Domshlak, H. H. Hoos, and D. Pool. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research*, 21:135–191, 2004.

[10] S. Bouveret, H. Fargier, J. Lang, and M. Lemaître. Allocation of indivisible goods: A general model and some complexity results. In *Proc. 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2005)*, pages 1309–1310. ACM Press, 2005.

[11] S. Bouveret and J. Lang. Efficiency and envy-freeness in fair division of indivisible goods: Logical representation and complexity. In *Proc. 19th International Joint Conference on Artificial Intelligence (IJCAI-2005)*, pages 935–940. Morgan Kaufmann, 2005.

[12] R. I. Brafman and C. Domshlak. Introducing variable importance tradeoffs into CP-nets. In *Proc. 18th Conference in Uncertainty in Artificial Intelligence (UAI-2002)*, pages 69–76. Morgan Kaufmann, 2002.

[13] S. J. Brams and A. D. Taylor. *Fair Division: From Cake-cutting to Dispute Resolution*. Cambridge University Press, 1996.

[14] G. Brewka. Preferred subtheories: An extended logical framework for default reasoning. In *Proc. 11th International Joint Conference on Artificial Intelligence (IJCAI-1989)*, pages 1043–1048. Morgan Kaufmann, 1989.

[15] 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.

[16] E. Cantillon and M. Pesendorfer. Auctioning bus routes: The London experience. In P. Cramton et al., editors, *Combinatorial Auctions*. MIT Press, 2006. To appear.

[17] CERN. LHC: The Large Hadron Collider. `http://lhc.web.cern.ch`.

[18] Y. Chevaleyre, U. Endriss, S. Estivie, and N. Maudet. Multiagent resource allocation with k-additive utility functions. In *Proc. DIMACS-LAMSADE Workshop on Computer Science and Decision Theory*, Annales du LAMSADE 3, pages 83–100, 2004.

[19] Y. Chevaleyre, U. Endriss, S. Estivie, and N. Maudet. Welfare engineering in practice: On the variety of multiagent resource allocation problems. In *Engineering Societies in the Agents World V*, pages 335–347. Springer-Verlag, 2005.

[20] Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet. Negotiating over small bundles of resources. In *Proc. 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2005)*, pages 296–302. ACM Press, 2005.

[21] 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.

[22] V. Conitzer and T. W. Sandholm. Complexity of mechanism design. In *Proc. 18th Annual Conference on Uncertainty in Artificial Intelligence (UAI-2002)*, pages 103–110. Morgan Kaufmann, 2002.

[23] V. Conitzer, T. W. Sandholm, and P. Santi. Combinatorial auctions with k-wise dependent valuations. In *Proc. 20th National Conference on Artificial Intelligence (AAAI-05)*, pages 248–254. AAAI Press, 2005.

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

[25] R. K. Dash, D. Parkes, and N. R. Jennings. Computational mechanism design: A call to arms. *IEEE Intelligent Systems*, 18(6):40–47, 2003.

[26] Desmo-J: Discrete-Event Simulation and Modelling in Java. `http://www.desmoj.de`.

[27] DEx: Dynamic Experimentation Toolkit. `http://dextk.org`.

[28] P. E. Dunne. *The Complexity of Boolean Networks*. Academic Press, 1988.

[29] P. E. Dunne. Extremal behaviour in multiagent contract negotiation. *Journal of Artificial Intelligence Research*, 23:41–78, 2005.

[30] P. E. Dunne. Multiagent resource allocation in the presence of externalities. In *Proc. 4th International Central and Eastern European Conference on Multi-Agent Systems (CEEMAS-2005)*, pages 408–417. Springer-Verlag, 2005.

[31] P. E. Dunne and Y. Chevaleyre. Negotiation can be as hard as planning: Deciding reachability properties of distributed negotiation schemes. Technical Report ULCS-05-009, Dept. of Computer Science, University of Liverpool, 2005.

[32] P. E. Dunne, M. Wooldridge, and M. Laurence. The complexity of contract negotiation. *Artificial Intelligence*, 164(1–2):23–46, 2005.

[33] U. Endriss and N. Maudet. Welfare engineering in multiagent systems. In *Engineering Societies in the Agents World IV*, pages 93–106. Springer-Verlag, 2004.

[34] U. Endriss and N. Maudet. On the communication complexity of multilateral trading: Extended report. *Journal of Autonomous Agents and Multi-Agent Systems*, 11(1):91–107, 2005.

[35] U. Endriss, N. Maudet, F. Sadri, and F. Toni. On optimal outcomes of negotiations over resources. In *Proc. 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2003)*, pages 177–184. ACM Press, 2003.

[36] Evo Artificial Life Framework. `http://sourceforge.net/projects/evo/`.

[37] P. Faratin. *Automated Service Negotiation Between Autonomous Computational Agents*. PhD thesis, Dept. of Electronic Engineering, Queen Mary College, University of London, 2000.

[38] 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 Science (MFCS-2003)*, pages 21–45. Springer-Verlag, 2003.

[39] M. Fischer and N. J. Pippenger. Relations among complexity measures. *Journal of the ACM*, 26:361–381, 1979.

[40] I. Foster and C. Kesselman, editors. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 2nd edition, 2004.

[41] Y. Fujishima, K. Leyton-Brown, and Y. Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *Proc. 16th International Joint Conference on Artificial Intelligence (IJCAI-1999)*, pages 548–553. Morgan Kaufmann, 1999.

[42] J. M. Garrido. *Object-oriented Discrete-event Simulation with Java: A Practical Introduction*. Kluwer, 2001.

[43] H. Geffner. *Default Reasoning: Causal and Conditional Theories*. MIT Press, 1992.

[44] A. Giovannucci, J. A. Rodríguez-Aguilar, A. Reyes, F. X. Noria, and J. Cerquides. *i*Bundler: An agent-based decision support service for combinatorial negotiations. In *Proc. 19th National Conference on Artificial Intelligence (AAAI-2004)*, pages 1012–1013. AAAI Press, 2004.

[45] A. Giovannucci, J. A. Rodríguez-Aguilar, A. Reyes, F. X. Noria, and J. Cerquides. Towards automated procurement via agent-aware negotiation support. In *Proc. 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2004)*, pages 244–253. ACM Press, 2004.

[46] M. Golfarelli, D. Maio, and S. Rizzi. A task-swap negotiation protocol based on the contract net paradigm. Technical Report 005-97, CSITE, University of Bologna, 1997.

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

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

[49] P. J. 't Hoen and J. A. La Poutré. A decommitment strategy in a competitive multi-agent transportation setting. In *Agent-Mediated Electronic Commerce V*, pages 56–72. Springer-Verlag, 2004.

[50] M. Höpf. Holonic manufacturing systems: The basic concept and a report of IMS test case 5. In J. K. H. Knudsen et al., editors, *Sharing CIM Solutions*. IOS Press, 1994.

[51] JADE: Java Agent Development Framework. `http://jade.tilab.com`.

[52] 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.

[53] B. Kádár, L. Monostori, and S. Szelke. An object-oriented framework for developing distributed manufacturing architectures. *Journal of Intelligent Manufacturing*, 9(2):173–179, 1998.

[54] J. Kalagnanam and D. C. Parkes. Auctions, bidding and exchange design. In D. Simchi-Levi et al., editors, *Handbook of Quantitative Supply Chain Analysis: Modeling in the E-Business Era*. Kluwer, 2004.

[55] V. Krishna. *Auction Theory*. Academic Press, 2002.

[56] C. Lafage and J. Lang. Logical representation of preferences for group decision making. In *Proc. 7th International Conference on Principles of Knowledge Representation and Reasoning (KR-2000)*, pages 457–468. Morgan Kaufmann, 2000.

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

[58] D. J. Lehmann. Another perspective on default reasoning. *Annals of Mathematics and Artificial Intelligence*, 15(1):61–82, 1995.

[59] 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 Intelligence (IJCAI-1999)*, pages 206–211. Morgan Kaufmann, 1999.

[60] 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.

[61] M. Lemaître, G. Verfaillie, F. Jouhaud, J.-M. Lachiver, and N. Bataille. Selecting and scheduling observations of agile satellites. *Aerospace Sciences and Technology*, 6:367–381, 2002.

[62] 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 Intelligence (CCIA-2003)*. IOS Press, 2003.

[63] MAML: Multi-Agent Modeling Language. `http://www.maml.hu`.

[64] M. Matsumoto and T. Nishimura. Mersenne Twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, 8(1):3–30, 1998.

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

[66] H. Moulin. *Fair Division and Collective Welfare*. MIT Press, 2003.

[67] R. B. Myerson and M. A. Satterthwaite. Efficient mechanisms for bilateral trading. *Journal of Economic Theory*, 29(2):265–281, 1983.

[68] N. Nisan. Bidding languages for combinatorial auctions. In P. Cramton et al., editors, *Combinatorial Auctions*. MIT Press, 2006. To appear.

[69] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[70] C. H. Papadimitriou. Algorithms, games, and the Internet. In *Proc. 33rd Annual ACM Symposium on Theory of Computing (STOC-2001)*, pages 749–753. ACM Press, 2001.

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

[72] H. Van Dyke Parunak, A. D. Baker, and S. J. Clark. The AARIA agent architecture: An example of requirements-driven agent-based system design. In *Proc. 1st International Conference on Autonomous Agents (Agents-1997)*, pages 482–483. ACM Press, 1997.

[73] S. Phelps, P. McBurney, S. Parsons, and E. Sklar. Co-evolutionary auction mechanism design: A preliminary report. In *Agent-Mediated Electronic Commerce IV*, pages 123–142. Springer-Verlag, 2002.

[74] S. Phelps, S. Parsons, and P. McBurney. An evolutionary game-theoretic comparision of two double auction market designs. In *Agent-Mediated Electronic Commerce VI*. Springer-Verlag, 2005. To appear.

[75] RePast: Recursive Porus Agent Simulation Toolkit. `http://repast.sourceforge.net`.

[76] A. Reyes-Moro and J. A. Rodríguez-Aguilar. *i*AuctionMaker: A decision support tool for mixed bundling. In *Agent-Mediated Electronic Commerce VI*. Springer-Verlag, 2005. To appear.

[77] 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.

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

[79] T. W. Sandholm. Distributed rational decision making. In G. Weiß, editor, *Multiagent Systems: A Modern Approach to Dis-*

tributed Artificial Intelligence. MIT Press, 1999.

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

[81] T. W. Sandholm. Optimal winner determination algorithms. In P. Cramton et al., editors, *Combinatorial Auctions*. MIT Press, 2006. To appear.

[82] T. W. Sandholm and V. R. Lesser. Leveled commitment contracts and strategic breach. *Games and Economic Behavior*, 35(1–2):212–270, 2001.

[83] T. W. Sandholm and S. Suri. Side constraints and non-price attributes in markets. In *Proc. IJCAI Workshop on Distributed Constraint Reasoning*, 2001.

[84] J. E. Savage. *The Complexity of Computing*. John Wiley and Sons, 1976.

[85] C. P. Schnorr. The network complexity and Turing machine complexity of finite functions. *Acta Informatica*, pages 95–107, 1976.

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

[87] 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.

[88] P. Sousa, C. Ramos, and J. Neves. Manufacturing entities with incomplete information. *Studies in Informatics and Control*, 9(2):79–88, 2000.

[89] 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.

[90] Stephens Inc. Internet Supply Chain Team. *Strategic Sourcing: Applications to Turn Direct Materials Procurement into Competitive Advantage*. Industry Report, 2001.

[91] Swarm. `http://www.swarm.org`.

[92] K. Sycara, S. F. Roth, N. Sadeh, and M. S. Fox. Resource allocation in distributed factory scheduling. *IEEE Expert*, 6(1):29–40, 1991.

[93] L. Tesfatsion. Agent-based computational economics: Growing economies from the bottom up. *Artificial Life*, 8(1):55–82, 2002.

[94] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16(1):8–37, 1961.

[95] Visionary manufacturing challenges for 2020. Committee on Visionary Manufacturing Challenges, National Research Council. National Academic Press, 1999.

[96] I. Wegener. *The Complexity of Boolean Functions*. John Wiley and Sons, 1987.

[97] M. Wooldridge. *An Introduction to Multiagent Systems*. John Wiley and Sons, 2002.

[98] P. R. Wurman, M. P. Wellman, and W. E. Walsh. A parametrization of the auction design space. *Games and Economic Behaviour*, 35(1–2):304–338, 2001.

[99] R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1):183–190, 1988.

[100] A. C.-C. Yao. Some complexity questions related to distributive computing (preliminary report). In *Proc. 11th Annual ACM Symposium on Theory of Computing (STOC-1979)*, pages 209–213. ACM Press, 1979.

[101] H. P. Young. *Equity in Theory and Practice*. Princeton University Press, 1994.

[102] X. Zhang and V. Lesser. Multi-linked negotiation in multi-agent systems. In *Proc. 1st International Joint Conference on Autonomous Agents And Multiagent Systems (AAMAS-2002)*, pages 1207–1214. ACM Press, 2002.

The validity of all cited URLs has been verified at the time of writing (August 2005).