Simulation of Negotiation Policies in Distributed Multiagent Resource Allocation

Hylke Buisman, Gijs Kruitbosch, Nadya Peek and Ulle Endriss

Artificial Intelligence Programme
University of Amsterdam
{hbuisman,gkruitbo,npeek,ulle}@science.uva.nl

Abstract. In distributed approaches to multiagent resource allocation, the agents belonging to a society negotiate deals in small groups at a local level, driven only by their own rational interests. We can then observe and study the effects such negotiation has at the societal level, for instance in terms of the economic efficiency of the emerging allocations. Such effects may be studied either using theoretical tools or by means of simulation. In this paper, we present a new simulation platform that can be used to compare the effects of different negotiation policies and we report on initial experiments aimed at gaining a deeper understanding of the dynamics of distributed multiagent resource allocation.

1 Introduction

Many complex application domains can be modelled as multiagent systems in which agents of varying capabilities interact. Building such artificial societies of autonomous software agents and devising suitable interaction mechanisms presents a formidable research challenge. Within such a society, agents will have to negotiate on a number of issues, including the best possible distribution of the resources available in the system amongst the individual agents. The field of *multiagent resource allocation* [1] is concerned with the design and analysis of mechanisms for finding a suitable assignment of resources to agents, given the individual interests of the agents as well as any technical constraints imposed by the system. In *distributed* approaches to multiagent resource allocation, the computational burden of the process of allocating resources is shared by all the agents in the society. In *centralised* approaches, notably combinatorial auctions [2], on the other hand, the task of computing the optimal allocation is relegated to an external entity (e.g. an auctioneer). Here we concentrate on distributed approaches, which provide a particularly rich setting in which to study interaction in multiagent systems.

The specific resource allocation framework we adopt has previously been studied by a number of authors [3–5]. It assumes that a finite number of indivisible goods needs to be allocated to a finite number of agents. Goods cannot be shared by more than one agent, and we assume that some initial allocation is given to begin with. Each agent expresses their preferences in terms of a valuation function mapping bundles of goods to the (positive) reals, and will only accept deals (possibly involving monetary side-payments) resulting in a strict increase in utility for themselves (so-called *myopic individual rationality*). Detailed definitions will be given in Section 2. We are interested in the effects such locally conducted and individually rational deals have on the agent society as a whole. In particular, we seek to understand under what circumstances a sequence of deals will converge to an allocation that would be considered optimal in view of a particular aggregation of the individual agent preferences. Here we consider both measures for economic *efficiency*, such as Pareto optimality or the sum of individual utilities, and notions of *fairness*, such as envy-freeness or the level of utility enjoyed by a society's poorest member [6, 7, 1].

Previous work has studied such convergence properties mostly from a theoretical point of view [3, 4]. Where it is *possible* to derive general theorems on (guaranteed) convergence to a socially desirable allocation, this seems indeed the best approach. However, for many realistic scenarios some of the assumptions on which the correctness of such theorems rests simply will not hold. The amount of time available to negotiate in will be limited, and therefore perhaps not sufficient to attain an optimal state. To allow any kind of deal includes allowing very complex deals involving many agents and resources, which is computationally expensive. To be able to find convergence trends, it then becomes interesting to simulate many runs of a distributed negotiation process, under similar conditions, to see whether it may be possible to make empirically founded predictions. Previous work along these lines includes that of Andersson and Sandholm [8] and Estivie and colleagues [9, 10]. The former have studied the effects of sequencing different types of deals (such as deals involving only a single resource at a time, or deals involving the swapping of two items), while the latter have concentrated on understanding under what circumstances we can expect to see fair allocations emerge when rational agents negotiate. These works offer interesting insights into the dynamics of distributed multiagent resource allocation. However, what has been missing so far is a generic simulation platform that would allow the experimenter to vary a wide range of parameters, to run simulations for different types of agent valuations and different negotiation policies, and to evaluate outcomes with respect to a range of different efficiency and fairness criteria.

In order to fill this gap, we have developed a simulation platform called the MultiAgent Distributed Resource Allocation Simulator (MADRAS). Using this platform, a user can easily generate a scenario with given numbers of agents and resources, in which the agents have their own preferences. The agents are able to negotiate amongst themselves to establish trades using money. Using such a scenario, the user is able to run a variety of experiments to see under what circumstances the agents most beneficially manage to reallocate their resources. Finally the platform provides possibilities for visualising several experiment statistics. In this paper, we introduce the MADRAS platform and report on a set of initial experiments that we have conducted using the platform.

The remainder of this paper is organised as follows: Section 2 briefly maps out the formal resource allocation framework we use and recalls a relevant result from the literature regarding the convergence of negotiation processes to a socially optimal allocation. Then Section 3 describes the MADRAS platform, which consists of three modules: the generation of resource allocation scenarios (in particular the generation of valuation functions); the simulation of negotiation processes conforming to a chosen negotiation policy for the agents; and an experimentation support module for evaluating and visualising the data produced during simulation. A selection of the experiments we have run using MADRAS are documented in Section 4. Section 5 concludes with a brief discussion of possible directions for future work.

2 Preliminaries

In this section, we briefly review the basic definitions of the resource allocation framework we adopt and we recall a fundamental convergence result linking the negotiation behaviour of individual agents and the emergence of optimal allocations at the societal level. Full details are available elsewhere [4].

2.1 Formal Framework

Let $\mathcal{A} = \{1, \ldots, n\}$ be a set of *agents*, and let $\mathcal{R} = \{r_1, \ldots, r_m\}$ be a set of *resources* (or goods). An *allocation* $A : \mathcal{A} \to 2^{\mathcal{R}}$ is a division of the resources in $\mathcal{R}$ amongst the agents in $\mathcal{A}$. Any allocation A has to assign each resource to *exactly* one agent. Agents may have different preferences dictating which resources they want, and how much they want them. A valuation function $v : 2^{\mathcal{R}} \to \mathbb{R}$ maps any given bundle of resources to a value in real numbers (this may be restricted to the positive reals and zero). We write $v_i(A)$ for $v_i(A(i))$, the valuation assigned by agent i to the bundle it receives in allocation A.

A *deal* $\delta = (A, A')$ is defined by the transition between two allocations (before/after). This model allows for any number of resources being reallocated amongst any number of agents within a single deal. Deals may be paired with monetary *side payments*. These are modelled using a payment function $p : \mathcal{A} \to \mathbb{R}$, satisfying $\sum_{i \in \mathcal{A}} p(i) = 0$. A positive $p(i)$ indicates that agent i has to *pay*, while a negative $p(i)$ means that i will *receive* money. The *utility* enjoyed by an agent i in a given negotiation state is computed by subtracting the sum of previous payments made by i from the valuation i assigns to the *bundle* of resources it currently holds (*quasi-linear* utility).

Whether an agent will accept a given deal (including side payments) depends entirely on whether that deal seems rational to the agent. There are a number of different rationality criteria that we could consider [7]. In this paper we shall concentrate on a myopic form of *individual rationality* [3]. A deal $\delta = (A, A')$ is called individually rational (and considered acceptable) iff it increases the utility of each of the agents involved. That is, we require $v_i(A') - v_i(A) > p(i)$ for every agent i involved in the deal (non-involved agents may receive money, but cannot be required to pay anything).

In the most general case, we assume that there are no restrictions on time or computational resources: any deal that is individually rational may eventually be identified and implemented. For more realistic scenarios, besides the restriction imposed by the agents' rationality requirements, we may also impose structural restrictions on deals. In this paper, we are going to be interested in two such classes of deals. The class of *1-resource deals* is the class of deals involving the reallocation of a single item only (and hence only two agents). The class of *bilateral deals* is the class of deals involving only two agents (but any number of resources at a time).

2.2 Convergence

Given this framework, the question arises what kinds of allocations we can expect agents to negotiate. We are interested in assessing the quality of an allocation in terms of various criteria for economic efficiency and fairness, borrowed from the literature on social choice theory and welfare economics [6, 11]. Several examples will be given in Section 3.3. For now, let us just recall the notion of *utilitarian social welfare*. The utilitarian social welfare $sw_u(A)$ of an allocation A is given by the sum of individual agent valuations:

$$sw_u(A) = \sum_{i \in \mathcal{A}} v_i(A)$$

Observe that taking past payments into account does not affect this definition (as they always add up to zero). High social welfare implies high average utility, which justifies this as a metric for assessing the quality of an allocation.

Now, what is the connection between the *local* concept of individual rationality driving negotiation and the *global* concept of social welfare? An important result establishes that *any sequence of individually rational deals will eventually result in an allocation with maximal utilitarian social welfare* [3]. That is, no central point of control is required. We can let agents negotiate in a distributed manner following only their own selfish interests and still guarantee that the system will, at some point, reach a state that would be considered optimal from a social point of view. While this may seem surprising at first, it is actually not difficult to prove. They key insight is that, in fact, a deal turns out to be individually rational iff it increases social welfare [4]. However, we stress that the above convergence result holds only if we do not place any structural restrictions on deals. For instance, if agents will only negotiate individually rational *bilateral* deals, then the social optimum may not be reachable.

3 The MADRAS Platform

This section explains the functionality of the MADRAS simulation platform for distributed resource allocation. The platform consists of three modules:

1. The *scenario generator* is used to generate problem instances, characterised by sets of agents and resources, valuation functions for these agents, and an

initial allocation of resources. Scenarios may be defined manually or generated automatically (using user-defined constraints). We have also defined an XML-based language to store and communicate scenario descriptions. Section 3.1 discusses the most challenging task falling under this module, namely the automatic generation of valuation functions.

2. The module for *negotiation simulation* reads in a scenario description and then simulates a negotiation process. How this works exactly is determined by the chosen *negotiation policy*. Such a policy fixes choices regarding the rationality criterion used by the agents, structural restrictions imposed on deals, and the search algorithms used to identify the next deal meeting the specified requirements. This will be discussed in Section 3.2. The module can save a record of the resulting negotiation process on file.
3. The *experimentation support* module reads in one or several files documenting particular negotiation runs and can produce a wide range of experimentation statistics from this data. In particular, it can be used to plot how social welfare and similar metrics develop as negotiation progresses. Examples are given in Section 3.3.

3.1 Generating Agent Valuations

We have opted for a logic-based representation of valuation functions based on weighted propositional formulas [12, 1]. In this representation, agents may express *goals* as propositional formulas over the set of atomic propositions given by the resource names $\{r_1, \ldots, r_m\}$. For example, the goal $r_1 \wedge (r_2 \vee r_3)$ indicates that the agent in question desires to obtain r_1 and at least one of r_2 and r_3. Furthermore, agents assign numerical weights to these goals. An agent's valuation for a given bundle R is then given by the sum of the weights of the goals that are satisfied by R.[1] For example, if an agent has the weighted goals $(r_1, 3)$ and $(r_1 \wedge r_2, 1)$, then they will assign value 3 to the bundle $\{r_1\}$, value 4 to the bundle $\{r_1, r_2\}$, and value 0 to both $\{r_2\}$ and the empty bundle. This logic-based representation is not only very flexible and natural, but also fully expressive and often allows for representing interesting valuation functions in a concise manner. As far as the *automatic* generation of valuation functions is concerned, the current implementation is restricted to goals that are conjunctions of atomic propositions. This is isomorphic to the so-called *k-additive form* of representing valuation functions [13, 1].

After having specified the number of agents and resources in the scenario generation module of MADRAS, the user can initiate the automatic generation of valuations. To this end, the user may manipulate the following parameters:

– The maximum length k (number of atoms in a conjunction) for all goals in the valuation function. Either a precise value can be given, or k can be taken from a user-specified uniform or normal distribution. This parameter determines the degree of synergy between different resources.

[1] Here we interpret bundles R as models of propositional logic: an atomic proposition r is taken to be *true* in a model characterised by R iff r is an element of R.

– A function specifying the number of the goals of a given length that will actually be generated. This parameter determines the range of different bundles that an agent may wish to obtain.
– A distribution from which to pick the numerical weights for our goals.

We stress that a choice of different parameters would have been possible as well. While the present implementation gives the experimenter a good degree of control and allows for the generation of a wide range of scenarios, further research is required to establish useful guidelines for generating interesting and application-relevant sets of valuations.

Similar problems have been addressed in the context of research on combinatorial auctions, in particular for the development of the combinatorial auction test suite CATS [14]. Like for our logic-based language, bids in combinatorial auctions are symbolic expressions for encoding valuation functions. CATS can generate such bids. It is intended to model realistic bidding behaviour, for different types of real-world scenarios (such as spectrum auctions or temporal scheduling), and has been developed for testing the performance of winner determination algorithms for combinatorial auctions. Unfortunately, this data cannot (at least not immediately) be used for simulating *distributed* multiagent resource allocation. One problem is the fact that CATS does not label bids with the name of the agent bidding (the reason being that this information is not relevant from the viewpoint of testing the performance of winner determination algorithms).[2]

3.2 Simulating Negotiation Policies

We emphasise that our aim has *not* been to build negotiating agents. We are only interested in *simulating* negotiation by generating sequences of deals that would be acceptable to the agents (given their valuation functions) and to evaluate how these deals affect social welfare. An important aspect for a simulation is the *negotiation policy* used. This is determined by the following parameters:

– *Rationality criterion:* What rationality criterion do agents use to decide whether a given deal is acceptable to them? At this stage, only individual rationality has been implemented.
– *Payment functions:* Are side payments allowed? If so, and if the payments are not uniquely determined by the rationality criterion, what are the exact payments that agents have to make for a given deal? At this stage, we have implemented two simple payment functions, the *globally uniform payment function* and the *locally uniform payment function* [10].
– *Structural restrictions:* What types of deals are possible? We have implemented *1-resource deals* and (a particular form of) *bilateral deals*.
– *Search algorithms:* Given the structural and rationality-related restrictions, how do we actually find a deal to implement? This requires a search algorithm. For 1-resource deals, this is not difficult: we simply search through

[2] For a discussion of exploiting CATS in the context of distributed multiagent resource allocation we refer to Estivie [9].

pairs of agents (i, j) and resources r (owned by i or j) and check whether reallocating r from from one to the other agent would be individually rational (or conformant to whichever rationality criterion we wish to apply). For bilateral deals (between two randomly chosen agents i and j), we have implemented a search algorithm that determines an *optimal partial reallocation* (OPR) of the union of the resources currently held by i and j amongst these two agents. This will be described in more detail below.

Running a simulation for a given scenario requires choosing a negotiation policy and specifying how long the simulation should run for. This could be done by providing a time limit, an upper bound on the number of new allocations, or an upper bound on the number of *attempts* of forging a deal (and hence moving to a new allocation). In MADRAS, we have opted for the latter. While running, the system will record the sequence of allocations encountered, as well as the payments made along the way. This data can later be used to calculate social welfare and other experiment statistics.

In the remainder of this section we shall outline our approach to implementing the search algorithm for the OPR negotiation policy. After having selected a pair of agents (i, j) at random, this policy attempts to find the best possible bilateral deal between i and j. That is, it will try to find a reallocation of the items held by i and j that would maximise the sum of the valuations of i and j. To find an optimal partial reallocation, we use the A* algorithm [15]. This approach is inspired by work of Sandholm on optimal algorithms for the winner determination problem in combinatorial auctions [16].

When using A*, one must define the set of states making up the search space, the range of moves between states, the goal states, and a heuristic for moving through the state space effectively. In our case, a state is characterised by the set of resources for which we have already made a decision as to which of the two agents should receive it. Initially, all resources are unallocated. Each move assigns another resource to one of the agents, and the goal state is reached when there are no more resources to allocate.

A* refers to two functions: The function g maps each state to the value (sum of valuations of the two agents) we get for the resources already allocated in that state. The heuristic function h estimates the additional value we can still expect to generate by allocating also the remaining items. A* maintains a so-called *fringe* of states in the search space, and will always pursue the state s from the fringe which maximises $g(s) + h(s)$. By a classical result, A* will be guaranteed to find the optimal allocation provided the heuristic function h is *admissible* [15]. In our context, admissibility means that h *never underestimates* the real additional value still obtainable. For the heuristic function we are using the following formula (for a state s and agents i and j):

$$h(s) = \left(\sum_{(G,\alpha) \in \Gamma_i(s)} \alpha \right) + \left(\sum_{(G,\alpha) \in \Gamma_j(s)} \alpha \right)$$

Here $\Gamma_i(s)$ is the set of weighted goals in the representation of the valuation function of agent i that are not yet satisfied in state s, but that could still be satisfied in a follow-up state (if i were to receive all remaining resources, for instance). Formally, if $s(i)$ is the set of resources allocated to i in state s and if $U(s)$ is the set of resources not allocated to anyone in state s, then $(G, \alpha) \in \Gamma_i(s)$ iff $s(i) \not\models G$ and $s(i) \cup U(s) \models G$. That is, for the heuristic we are computing the marginal valuation for each individual agent in the most optimistic manner and then sum these up without regard for possible conflicts. As we are restricting ourselves to positively weighted conjunctions of atomic propositions, it is not difficult to see that this constitutes an admissible heuristic for A*. While being simplistic (and certainly still subject to improvements), our heuristic already results in a very significant speed-up in comparison to a simple breadth-first search and allows us to run interesting experiments.

3.3 Evaluating and Visualising Results

The third module is a *grapher* that can be used to visualise the results obtained during simulation. Specifically, we can plot the social welfare of a sequence of allocations passed through during a simulation run. This allows the experimenter to evaluate and compare different negotiation policies in view of different desiderata. As far as the quality of allocations is concerned, MADRAS allows for plotting graphs visualising the following concepts:

- *Utilitarian social welfare:* As explained in Section 2.2, this is given by the sum of individual utilities, and is a good measure for economic efficiency.
- *Egalitarian social welfare:* This is an alternative way of defining social welfare, emphasising fairness rather than efficiency. The egalitarian social welfare of a negotiation state (possibly involving past payments) is the utility assigned to that state by the least happy agent [11].
- *Elitist social welfare:* This is defined as the utility of the happiest agent [7].
- *Envy:* Another fairness criterion is envy-freeness [17]. An agent i is said to *envy* another agent j iff agent i would prefer to own agent j's bundle of resources. Envy-free allocations are difficult to obtain through distributed negotiation, and may not even exist at all. MADRAS can plot how the *degree of envy* develops as negotiation progresses, for different interpretations of that concept. For instance, we can plot the *maximum* or the *average envy* experienced by any one agent, or we can plot the *number of envious agents* in the society.

MADRAS can also generate graphs showing the number of resources held by each agent across allocations. The closer the system gets to an optimal state, the more difficult it becomes to find a possible deal. To visualise such effects, MADRAS can plot graphs showing the number of implemented reallocations per amount of attempts at finding a deal between two randomly chosen agents.

Fig. 1 is an example for the kind of graphs generated by MADRAS. It shows how three different kinds of social welfare develop as negotiation progresses. For

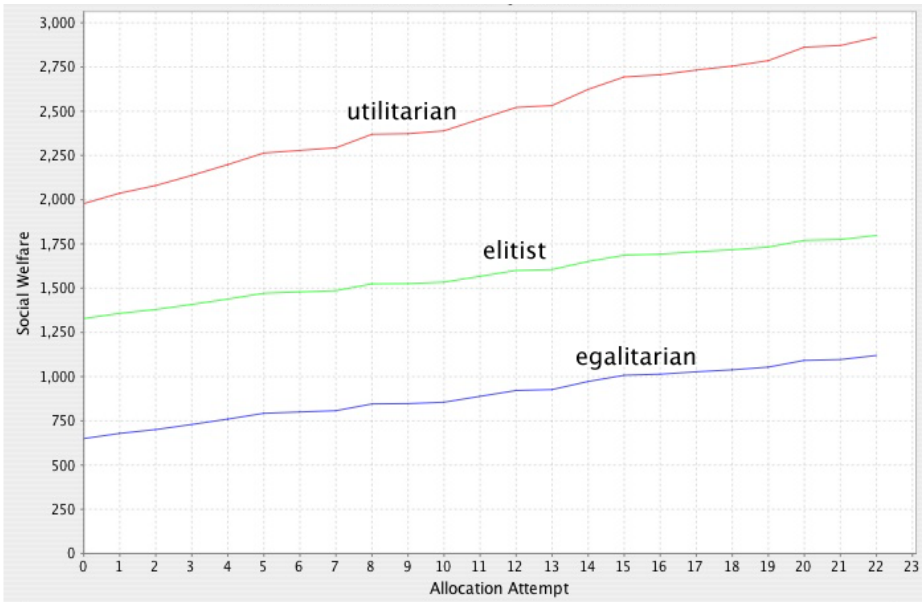

Fig. 1. Comparing utilitarian, egalitarian and elitist social welfare

this particular example, we have created 50 resources and only 2 agents, and the chosen negotiation policy requires agents to negotiate individually rational 1-resource deals using the locally uniform payment function (which means that payments are arranged so as to evenly distribute the social surplus generated by a deal amongst the participating agents [10]).

Note that for the special case of a society with only two agents, the utilitarian social welfare is actually the sum of the egalitarian and the elitist social welfare, and this is clearly visible in Fig. 1. Furthermore, we can see that utilitarian social welfare monotonically increases over time, as predicted by the aforementioned result linking individual rationality and utilitarian social welfare [4]. Egalitarian and elitist social welfare are computed with respect to *utility* (rather than valuation, meaning that previous side payments are taken into account). Hence, as each deal is individually rational, also these must increase monotonically. Due to our particular choice of payment function, they furthermore increase at exactly the same rate. Hence, while egalitarian social welfare does increase, negotiation does not affect the relative fairness of the allocation: the difference in utility between the two agents does not change.

4 Experiments

In this section we report on a couple of initial experiments which we have carried out using MADRAS.

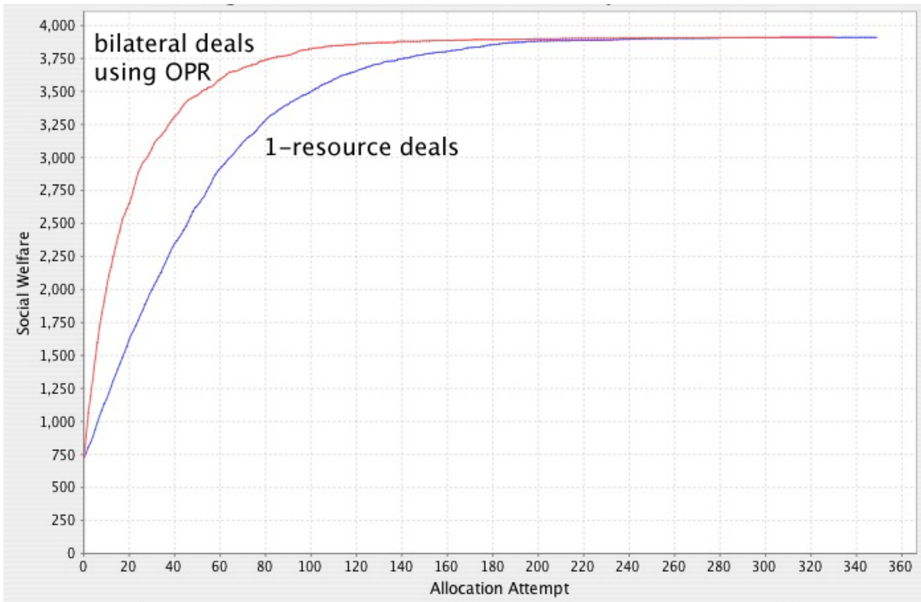

Fig. 2. Social welfare using 1-resource vs. OPR deals in modular domains

4.1 Comparing Negotiation Policies in Modular Domains

This first experiment is aimed at comparing the two negotiation policies currently implemented in MADRAS in view of reaching an allocation that maximises utilitarian social welfare when all agents are known to have *modular* valuation functions. Recall that a valuation function v is called modular iff it satisfies $v(R_1 \cup R_2) = v(R_1) + v(R_2) - v(R_1 \cap R_2)$ for all $R_1, R_2 \subseteq \mathcal{R}$. That is, in modular domains an agent's valuation for a given bundle R can be computed by adding up its valuations for the elements of R.

It is known that any sequence of individually rational 1-resource deals will eventually result in an allocation with maximal utilitarian social welfare, provided that all agents use modular valuation functions [4]. Given that the bilateral OPR policy subsumes the 1-resource deal policy,[3] the same must be true for the former. That is, both negotiation policies guarantee optimal outcomes in modular domains. The question is which policy does so *faster*.

Intuition suggests that the OPR policy should be faster in the sense that fewer deals are required to reach the optimum (as each individual deal can be expected to result in a greater increase in overall utility). What is not clear is how significant the difference is, and whether that advantage would not be outweighed by the fact that finding an individual deal under the OPR policy is

[3] The bilateral OPR policy subsumes the 1-resource deal policy in the sense that whenever there is a 1-resource deal that would be applicable between two agents, the OPR policy will either implement that same deal or a deal that is even better.

considerably more complex than under the 1-resource deal policy (NP-complete as opposed to linear).

Fig. 2 confirms our intuitions. This experiment involves 10 agents with modular valuations over 50 resources, with each agent assigning a positive weight drawn from a uniform distribution over [1..100] to 20 randomly selected resources. The graphs show an average of 20 experiment runs from one scenario description. Fig. 2 shows that convergence is in fact *much* faster for full bilateral negotiation using the OPR policy than for 1-resource deals, at least if "time" is measured in terms of the number of attempts made at forging an acceptable deal. Additionally, data not shown in Fig. 2 suggests that the real time required for reaching the optimum is of a similar order of magnitude for both negotiation policies. It appears that the high complexity of the search involved in computing an optimal partial reallocation in the bilateral scheme is traded off against the overhead in search required to find matching trading partners under the 1-resource policy. Of course, our findings regarding real-time performance need to be interpreted with some care: they are strongly dependent on the specific implementation choices made in the MADRAS system.

4.2 Comparing Negotiation Policies for Varying Degrees of Synergy

Our second set of experiments is aimed at comparing the performance of our two negotiation policies for varying degrees of synergy in the agent valuations. Modular valuations (as studied in Section 4.1) are representable as sets of weighted goals, each of which has length $k = 1$. If we allow proper conjunctions in the goals (of length $k > 1$), then this may be understood as synergies between the items occurring together in the same conjunction. For instance, if an agent has the goal $(r_1 \wedge r_2, 5)$, they will only receive the value of 5 if they own both of r_1 and r_2 *together*; the individual items by themselves may have no value at all.

We have produced two groups of experiments for valuation functions represented by sets of goals of length $\leq k$, with k ranging from 1 to 6. The results are shown in Figures 3 and 4, respectively. As before, there are 10 agents and 50 resources. For each value of k, we have generated 3 different scenarios and run 10 simulations for each of the two negotiation policies for each such scenario (so each of the curves shown represents the average of 30 runs). The only difference between the two groups of experiments, corresponding to Figures 3 and 4, concerns the number of weighted goals generated for each agent. In the case of Fig. 3, we have generated 20 goals of each of the required lengths for each agent. So, for instance, if $k = 3$ then an agent will have 20 goals of length 1, 20 goals of length 2, and 20 goals of length 3. All weights are drawn independently from a uniform distribution over [1..100]. In the case of Fig. 4, we have generated 30 goals *in total* for each agent (32 in the case of $k = 4$). For instance, for $k = 2$ we have generated 15 goals of length 1 and 15 goals of length 2; while for $k = 3$ we have generated only 10 goals of each length. For each pair of curves, the upper curve (better performance) corresponds to the OPR policy, and the other one to the 1-resource policy. In Fig. 4 the pairs are clearly visible as such; in Fig. 3 we

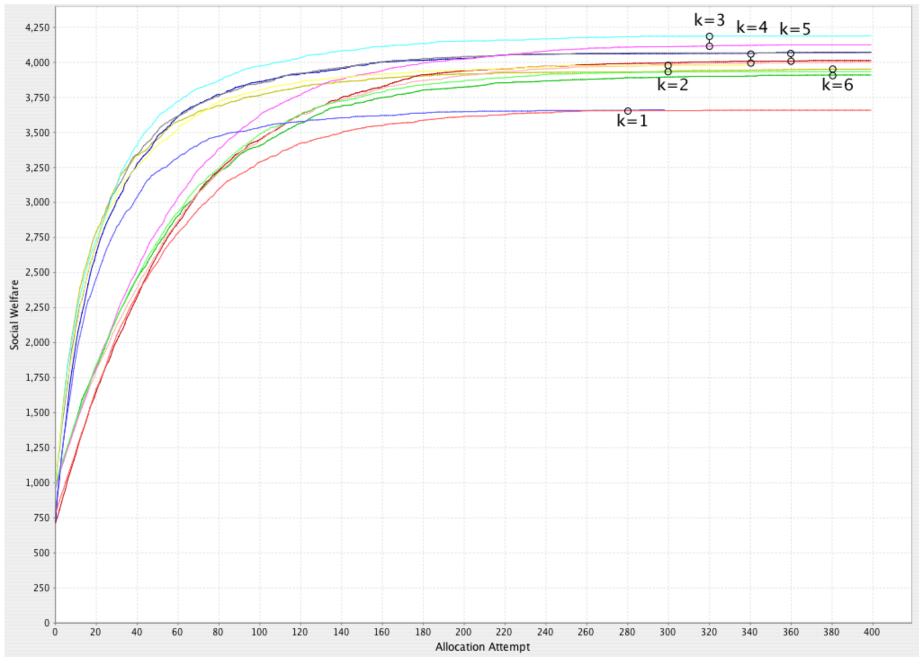

Fig. 3. Results when the number of goals per agent is proportional to k

have included some additional markers, which also indicate the maximum level of utilitarian social welfare achieved by each policy.

The experiments reveal some very interesting, and arguably surprising, effects. We know that for $k = 1$ (modular valuations), both negotiation policies will reach the same (optimal) state and that the OPR policy can be expected to get there in fewer steps than the 1-resource deal policy. This is visible in both figures. Now, as k increases (as valuations move further away from the simple modular case), we would have expected that the much more sophisticated OPR policy would outperform 1-resource negotiation even more significantly. For both policies, we would not expect to be able to reach an optimal state anymore (and this is indeed the case; data not shown here), but we would expect OPR deals to typically converge to a state with (maybe much) higher utilitarian social welfare than is attainable through 1-resource deals alone. As it turns out, this is the case only to a very limited extent. In Fig. 3, we can see that the gap between OPR and 1-resource increases as k increases up to $k = 3$, but then it becomes smaller again. So besides the expected trend described above, there must also be a second trend causing this gap to decrease again for larger values of k.

Our hypothesis is that this trend can be explained by the fact that the longer a goal, the lower the probability that all the required resources can be found in the set of items owned by the two agents supposed to forge a deal. Hence, having large amounts of long goals available in addition to the short goals present in all

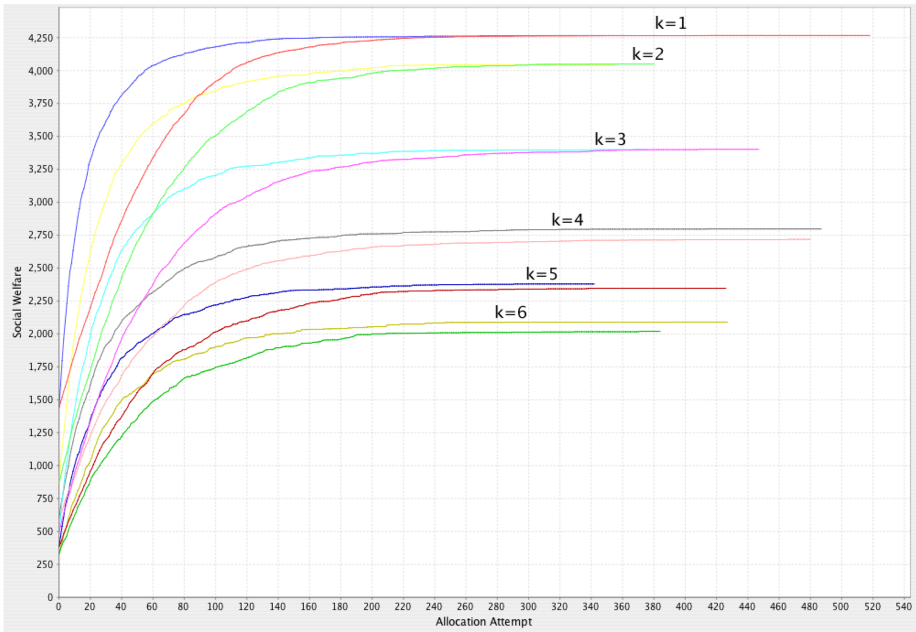

Fig. 4. Results when the total number of goals per agent is constant

the scenarios actually has very little effect on the outcomes. In fact, the presence of long goals may even be detrimental to achieving high social welfare (at least if the weights for goals of any length are drawn from the same distribution, as is the case for our experiments). The reason is that satisfying a single long goal may prevent a whole set of shorter goals (of other agents) from being satisfied.

In Fig. 4, the reduction in the gap between the two policies is less visible, but in any case it is still surprisingly small for larger values of k. Here we can also clearly observe a second effect: the attainable social welfare goes down as k increases. We expect this to be a consequence of there being fewer short goals in the scenarios with larger k (for Fig. 4 the *total* number of goals is constant, so the more different lengths there are, the fewer goals there are per length). These short goals are the easiest to satisfy, so the more there are the higher the sum of utilities. Indeed, further analysis of our data reveals that goals of length greater than 3 practically never get satisfied in the final allocation, and that for goals of length 3 typically no more than 1–2% get satisfied. This means that, really, what matters are the short goals of length 1 and 2.

A tentative conclusion based on these experiments would be that *any* form of bilateral negotiation (even if as seemingly sophisticated as OPR) is unlikely to be able to reach allocations that would satisfy goals that involve three or more resources. The reason for this is that chances are low that all the required resources would be present in the set of items jointly held by a particular pair of agents before negotiation between them starts. And those improvements over the *status quo* that *are* possible by means of bilateral negotiation then also seem to be achievable by means of its most basic form, namely 1-resource negotiation. Still, the OPR policy tends to achieve those moderate results in significantly fewer negotiation steps than the 1-resource policy (in terms of the number of deals attempted).

5 Conclusion

Dividing resources amongst a society of agents who have varying preferences can become a very complex task. Approaching this problem in a distributed manner and having the agents share the computational burden of the task seems promising on the one hand, but also raises serious challenges in terms of designing suitable interaction protocols. To be able to let the agents find an optimal allocation, there are many practical issues to consider. For instance, which negotiation policies are the fastest and still guarantee convergence to an optimum? How do behavioural criteria of individual agents influence the evolution of the system? A simulation platform such as MADRAS can be useful to test hypotheses about these issues. In this paper we have presented the basic functionality of MADRAS and explained the underlying principles. We have also reported on a number of experiments carried out using MADRAS. These experiments were aimed at comparing the performance of two negotiation policies in view of reaching a state with high utilitarian social welfare. In the first policy, agents negotiate individually rational deals that involve reallocating a single resource at a time. In the second policy, pairs of agents negotiate the best possible reallocation of the resources they own together amongst themselves. Despite the limited scope of these experiments, we can offer two tentative conclusions:

– Optimal partial reallocations between two agents tend to achieve the same or a higher level of social welfare than one-resource-at-a-time negotiation, and the former tend to do so in fewer steps than the latter.
– Even sophisticated forms of bilateral negotiation (such as optimal partial reallocations) are not well adapted to negotiation in domains with high degrees of synergies between large numbers of resources. In fact, in such domains the most basic form of negotiation (1-resource deals) can often achieve results very similar to those achieved by more sophisticated bilateral negotiation (although requiring a higher number of negotiation steps).

Even when studying the theoretical aspects of multiagent resource allocation closely, we are often surprised by the data that MADRAS generates. To fully understand the implications of varying any of the parameters incorporated into MADRAS we have to both analyse them theoretically and be able to explain the behaviour they generate in practice.

Our approach may be described as a middle-way between purely theoretical studies of convergence in multiagent resource allocation [3, 7, 4, 5] and work in agent-based computational economics [18, 19]. Epstein and Axtell [18], for instance, also study the emergence of various phenomena, but they do not specifically seek to understand the mathematical laws underlying such phenomena (and indeed, these may often be too complex to be easily understood or described). Here, on the contrary, we still see a mathematical explanation of emergent phenomena as an important goal, but the simulation of negotiation processes can serve as a tool for discovering the laws of distributed resource allocation mechanisms. Understanding these laws, in turn, will allow us to build better and more robust multiagent systems.

We should stress that certain design choices and features of the implementation of MADRAS are likely to have influenced (some of) our experimental results. To what extent this is the case will require further analysis. For instance, the heuristic we use for optimal partial reallocations is very important. Using the A* algorithm does not *per se* determine how to allocate goods that are not desired by either one of the agents involved in a bilateral deal. In our current implementation these uncontested resources remain with the agent they were initially allocated to, but other solutions such as random redistribution over the two agents involved are possible as well. The specific choices made during implementation in this regard may unwillingly influence not only the runtime of the algorithm but also the quality of the final allocation. Aspects such as these will require further study before we can fully bridge the gap between theoretical findings and implementation.

In addition to the above, a large number of interesting experiments remain to be done. Future work should further explore the constraints on preferences and agent rationality that are necessary to guarantee social optima. We conclude by giving three examples for specific directions of research that are being made possible by the availability of a simulation platform such as MADRAS:

– Pigou-Dalton transfers [11, 7, 20] are deals used in attempts at reducing inequality between agents.[4] However, it is known that, in the case of indivisible resources, using Pigou-Dalton transfers alone cannot guarantee convergence to an allocation with maximal egalitarian social welfare [4]. Using MADRAS would allow us to conduct research aimed at identifying constraints under which an egalitarian optimum *will* be found.
– MADRAS provides extensive possibilities for customising the agents' preferences. An interesting course of research would be to systematically examine the influence of certain classes of valuation functions on the reachability of certain social optima. For instance, while theoretical research has provided a good understanding of convergence behaviour in either the fully general case or the very simple case of modular valuations [4], little is known about convergence by means of structurally simple deals in case of valuations that are subjected to severe restrictions other than modularity.
– MADRAS also provides for another research approach which would not be possible without such a platform. This approach is to run many "arbitrary" experiments and examine these to form hypotheses (possibly using machine learning techniques). An approach of this type may produce findings which are not intuitive and would otherwise not be encountered easily.

[4] A Pigou-Dalton transfer is a deal between two agents that results in a transfer of utility from the stronger to the weaker agent, without reducing their sum of utilities.

References

1. Chevaleyre, Y., Dunne, P.E., Endriss, U., Lang, J., Lemaître, M., Maudet, N., Padget, J., Phelps, S., Rodríguez-Aguilar, J.A., Sousa, P.: Issues in multiagent resource allocation. Informatica **30** (2006) 3–31
2. Cramton, P., Shoham, Y., Steinberg, R., eds.: Combinatorial Auctions. MIT Press (2006)
3. Sandholm, T.W.: Contract types for satisficing task allocation: I Theoretical results. In: Proc. AAAI Spring Symposium: Satisficing Models. (1998)
4. Endriss, U., Maudet, N., Sadri, F., Toni, F.: Negotiating socially optimal allocations of resources. Journal of Artificial Intelligence Research **25** (2006) 315–348
5. Dunne, P.E., Wooldridge, M., Laurence, M.: The complexity of contract negotiation. Artificial Intelligence **164** (2005) 23–46
6. Arrow, K.J., Sen, A.K., Suzumura, K., eds.: Handbook of Social Choice and Welfare. North-Holland (2002)
7. Endriss, U., Maudet, N.: Welfare engineering in multiagent systems. In: Engineering Societies in the Agents World IV. LNAI 3071, Springer-Verlag (2004)
8. Andersson, M., Sandholm, T.W.: Contract type sequencing for reallocative negotiation. In: Proc. 20th International Conference on Distributed Computing Systems (ICDCS-2000), IEEE Press (2000)
9. Estivie, S.: Allocation de Ressources Multi-Agent: Théorie et Pratique. PhD thesis, Université Paris-Dauphine (2006)
10. Estivie, S., Chevaleyre, Y., Endriss, U., Maudet, N.: How equitable is rational negotiation? In: Proc. 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2006), ACM Press (2006)
11. Moulin, H.: Axioms of Cooperative Decision Making. Cambridge University Press (1988)
12. Lang, J.: Logical preference representation and combinatorial vote. Annals of Mathematics and Artificial Intelligence **42**(1–3) (2004) 37–71
13. Grabisch, M.: k-order additive discrete fuzzy measures and their representation. Fuzzy Sets and Systems **92** (1997) 167–189
14. Leyton-Brown, K., Pearson, M., Shoham, Y.: Towards a universal test suite for combinatorial auction algorithms. In: Proc. 2nd ACM Conference on Electronic Commerce, ACM Press (2000)
15. Hart, P., Nilsson, N., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics **4**(2) (1968) 100–107
16. Sandholm, T.W.: Optimal winner determination algorithms. In Cramton, P., Shoham, Y., Steinberg, R., eds.: Combinatorial Auctions. MIT Press (2006)
17. Brams, S.J., Taylor, A.D.: Fair Division: From Cake-cutting to Dispute Resolution. Cambridge University Press (1996)
18. Epstein, J.M., Axtell, R.L.: Growing Artificial Societies: Social Science from the Bottom Up. MIT Press (1996)
19. Tesfatsion, L., Judd, K., eds.: Handbook of Computational Economics: Agent-Based Computational Economics. Elsevier (2006)
20. Dunne, P.E.: Extremal behaviour in multiagent contract negotiation. Journal of Artificial Intelligence Research **23** (2005) 41–78

This work has been carried out in the context of the BSc Artificial Intelligence Honours Programme at the University of Amsterdam. MADRAS is available at `http://madras.infosyncratic.nl`.