Egalitarian Allocations of Indivisible Resources: Theory and Computation

P.-A. Matt and F. Toni

Department of Computing, Imperial College London,
South Kensington Campus, London SW7 2AZ, UK
{pmatt,f.toni}@imperial.ac.uk

Abstract. We present a mechanism for collaboration and coordination amongst agents in multi-agent societies seeking social equity. This mechanism allows to compute egalitarian allocations of indivisible resources to agents, reached via progressive revisions of social consensus. Egalitarian allocations are allocations with maximal egalitarian social welfare, where the egalitarian social welfare is given by the minimum worth (utility) assigned by agents to the resources they are given by the allocation. Egalitarian allocations are useful in a number of applications of multi-agent systems, e.g. service agents, satellite earth observation and agent oriented/holonic manufacturing systems. The mechanism we propose is distributed amongst the agents, and relies upon an incremental construction whereby agents join progressively in, forcing a revision of the current set of agreements amongst the prior agents. The mechanism uses search trees and a reduction operator simplifying the search for egalitarian allocations. We finally show how to reduce the negotiation time using social order-based coordination mechanisms and make agents find consensus efficiently using well-suited resource-preference orders.

1 Introduction

The emergence of societies of artificial agents, such as software agents, domestic or industrial robots, is a development with huge potential significance in the near future. For this significance to materialise, it is important that agent-designers render agents as autonomous as possible, by specifying decision-making strategies and rules of interaction. It is also important, from a global perspective, to clarify the properties a society of agents should exhibit in order to benefit applications.

To tackle this issue, we follow a worth-oriented paradigm adapted from the area of social choice theories [1–3] and welfare economics [4–6]. In this paradigm, agents assign a measure of worth to each of the possible states of affairs they may encounter, to represent, intuitively, the notion of goal satisfaction. In particular, we focus on states of affairs resulting from the allocation of indivisible resources to agents. Resources serve as an abstraction for objects, commodities, tasks, services, computational power etc.

Research in multi-agent systems so far focused mostly on utilitarianism, i.e. selfish agents trying to maximise their own good without concern for the global good of the society. Although appropriate for applications such as combinatorial auctions [7], utilitarian principles dangerously threaten cooperation between agents working in teams, because they are highly elitist by often generating situations where only a small percentage of agents detain the totality of the resources, preventing the others to achieve their goals. Moreover, purely artificial 'servants' may have no particular need of behaving selfishly and agent-designers do have the power to enforce specific interaction protocols and chosen reasoning strategies, in particular cooperative ones.

In our opinion, self-interest, natural within human societies, may be counterproductive in artificial societies. By shifting from natural to artificial societies, it is possible, and beneficial in some applications of resource allocation for cooperative agents, to abandon the utilitarian model. Egalitarian allocations allow to maximise the welfare of the agent that is less 'well-off' in the society, in terms of the worth it assigns to resources it is allocated. When agents are endowed with egalitarian strategies, they combine their actions for the global good of the society. This can be a fruitful approach in the resource allocation case: agents' cooperation allows better repartition of the resources, resulting in simultaneous quality improvements in the wide panel of services offered or set of tasks performed by the agents [8].

A recent overview of socially optimal allocations of resources achieved by means of negotiation can be found in [9]. Interestingly, the paper proves that any sequence of strongly equitable deals (see [9] for a definition) will eventually result in an egalitarian allocation. This purely theoretical result provides however no indication to designers of multi-agent systems on how agents can compute these deals. In this paper, we provide a new negotiation mechanism for solving in a distributed manner, and without any approximation, indivisible resource allocation problems so that the egalitarian social welfare of the multi-agent system is maximised, in the case where the worth of resources to agents is represented in terms of semi-linear utility functions. It is assumed that agents are willing to fully cooperate when working in teams. One may think of agents serving or acting for the interest of a common symbolic 'master' that in practice could correspond to an individual, company or institution of any kind.

Egalitarian allocations correspond to natural solutions in a number of application areas, and in particular for service-oriented multi-agent systems, where for example control of or access to services needs to be negotiated amongst agents. In this application, worth and utilities may respectively correspond to agents' competence in managing services and using the services. Then, the egalitarian approach would allow maximising the minimum competence in managing/using services in the system. Another application is earth observation via satellites [18, 16], where an egalitarian approach is advisable as observation satellites are a scarce and highly expensive resource, usually co-exploited by several entities all expecting fairness in the exploitation process. A third field for application and perhaps the most obvious one, is agent-oriented [19, 20] or holonic manufacturing systems [21]: agents can play the role of flexible self-organizing production units in a factory, where production platforms, raw materials and semi-finished products are utilised in function of the market's conditions. Factories are typically environments where no agent should be under-exploited and it is thought egalitarian principles can bring better productivity. A very interesting list of other practical applications has been given in [13].

The remainder of this paper is structured as follows. In section 2, we give a formal introduction to the resource allocation problem we tackle. In section 3, we introduce a general method to solve the problem efficiently and distributedly. Section 4 is dedicated to experiments on the time complexity required for solving allocation problems and the comparison of several social order-based coordination techniques for conducting negotiations and decision-making heuristics based on resource-preference orders. Section 5 draws some conclusions, compares our approach with related work and identifies some directions for future research.

2 Preliminaries

When several agents within a multi-agent system compete for the same resources, they need to overcome a complex conflict of interest, especially when agents need (or lack) many resources and have similar preferences. Autonomous agents need to solve the resulting resource allocation problem on their own, but cooperative agents also aim at finding a solution which is socially acceptable.

In this paper, we will refer to the agents and resources involved in a resource allocation problem as $a_1, a_2, \ldots, a_n$ and $r_1, r_2, \ldots r_m$, respectively, where the number of agents (n) and resources (m) are assumed to be strictly positive integers. We will assume that the resources are indivisible, so that a resource may be allocated only entirely and to one agent at most. We will use the following definition of allocation of resources to agents.

Definition 1. *Let $G = \{a_{i_1}, \ldots, a_{i_g}\}$ be a non-empty subset of $\{a_1, ..., a_n\}$ of cardinality $g \leq n$. G represents a* group *of g agents. An* allocation *for G is a Boolean table $A = ((A_{i,j}))_{g\times m}$ of g lines and m columns*

$$A^{\{i_1,\ldots,i_g\}} = \begin{pmatrix} i_1 : A_{i_1,1} & A_{i_1,2} & \ldots & A_{i_1,m} \\ \ldots & \ldots & \ldots & \ldots \\ i_g : A_{i_g,1} & A_{i_g,2} & \ldots & A_{i_g,m} \end{pmatrix}$$

such that

$$\forall j \in \{1, ..., m\} : \sum_{i, a_i \in G} A_{i,j} \leq 1.$$

We say that a_i gets *r_j if and only if $A_{i,j} = 1$.*

The inequalities in definition 1 sanction that each resource is allocated to one agent at most and indivisibility is expressed by the Boolean quantities. When clear from the context, we omit the agents indexes in the allocations, as done here-after. As an example, given the group $G = \{a_1, a_2, a_3\}$ and resources r_1, r_2, r_3, r_4, a possible allocation is

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

According to it, a_1 gets r_1, a_2 gets r_3 and r_4 and a_3 gets no resource.

In our framework, agents in a multi-agent systems are abstractly characterised by their own preferences concerning the resources to be distributed. These preferences are given via a global utility table:

Definition 2. *A* utility table *is a matrix* $U = ((u_{i,j}))_{n \times m}$ *with* n *lines and* m *columns of coefficients* $u_{i,j} \in I\!R^+$. *For each* $1 \leq i \leq n$ *and* $1 \leq j \leq m$, $u_{i,j}$ *is referred to as the* utility *of resource* r_j *for agent* a_i.

The utility of a resource for an agent provides a measure of its contribution to the agent's welfare.

A reasonable and convenient assumption is to consider that the welfare of an agent resulting from an allocation of resources is semi-linearly distributed over the resources, as given by the following definition:

Definition 3. *For any* $1 \leq i \leq n$, *the* welfare of agent a_i resulting from allocation A *is given by the equation* $w_i(A) = c_i + \sum_{j=1}^{m} u_{i,j} A_{i,j}$ *where* $c_i \in I\!R^+$.

The coefficient c_i intuitively represents the welfare of a_i prior to any allocation of resources. By our definition of welfare, basically all the resources that are given to a_i increase its welfare: the individual utilities of resources sum up. This simple model captures the idea that: the more resources agents get the higher their welfare, and agents prefer resources that contribute most to their welfare. However, this model does not capture the synergies resources may have when put together. This simplification enables us to avoid having to treat the allocation problem as a complex combinatorial one, as is for instance the case in combinatorial auctions.

Let us now introduce an optimality criterion on allocations, borrowed from the areas of social choice theories and welfare economics and having an *egalitarian* flavour. Informally, we are after allocations that maximise the egalitarian social welfare of the multi-agent system, defined metaphorically as the welfare of the 'unhappiest' agent in the system. Formally:

Definition 4. *The* egalitarian social welfare *of an allocation* A *for* $\{a_1, \ldots, a_n\}$ *is* $sw_e(A) = Min\{w_i(A) | i = 1, \ldots, n\}$. *An* egalitarian allocation *is an allocation* A^* *with maximal egalitarian social welfare.*

As an example, given agents a_1, a_2, a_3, resources r_1, r_2, r_3, r_4, and tables

$$U = \begin{pmatrix} 0.9\ 0.3\ 0.0\ 1.2 \\ 0.2\ 0.5\ 0.7\ 0.3 \\ 0.4\ 0.4\ 0.9\ 0.8 \end{pmatrix}, \quad \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 0.6 \\ 0.2 \\ 0.5 \end{pmatrix}$$

then, $A^* = \begin{pmatrix} 1\,0\,0\,0 \\ 0\,1\,1\,0 \\ 0\,0\,0\,1 \end{pmatrix}$ with $sw_e(A^*) = 1.3$, since $\begin{pmatrix} w_1(A^*) \\ w_2(A^*) \\ w_3(A^*) \end{pmatrix} = \begin{pmatrix} 1.5 \\ 1.4 \\ 1.3 \end{pmatrix}$.

We will use this example as a running example throughout the paper.

Note that there may be multiple egalitarian allocations within a society, but they all have the same egalitarian social welfare that we will denote sw_e^*.

Intuitively, by maximising the egalitarian social welfare in a society, we increase the chances that all agents are given enough resources to achieve their objectives. The notion of egalitarian social welfare contrasts with the more popular notion of utilitarian social welfare, amounting to the sum (or average) of all agents' welfare values. When utilitarian social welfare is maximised, resources are given to the agents that best use them, in terms of their individual welfare. But this may cause that other agents become inactive and incapable of achieving their objectives, which can be very ineffective, especially when the artificial society is supposed to provide a wide panel of services at the same time, as is the case e.g. in grid computing.

In the next section we give a method for constructing egalitarian allocations. Note that finding an egalitarian allocation of resources is a very hard search problem since there are exactly $(n+1)^m$ possible allocations to explore (that is the number of Boolean matrices of size $n \times m$ with at most one 1 per column). This means, for example, that even for a small multi-agent system with 9 agents and 10 resources to share, there are ten billions possible allocations to consider! Thus, the complexity of the decision-making problem the agents are confronted with when trying to compute an egalitarian allocation is huge.

3 Computation of egalitarian allocations

In this section we present an algorithm allowing agents to compute egalitarian allocations by working together, while taking autonomous decisions. When building an egalitarian allocation, two problems need to be solved at once: 1) finding the value sw_e^* of the optimal egalitarian social welfare and 2) actually finding an egalitarian allocation for the whole set of agents, with welfare sw_e^*.

3.1 Computing the optimal egalitarian social welfare

To solve the first problem, we will perform a dichotomous search. Dichotomy is a simple and elegant mechanism guaranteeing arbitrary precision and enabling fast estimation of the optimal social welfare. In this dichotomous search, an upper bound (ub) and lower bound (lb) for this optimal value are updated iteratively. These bounds are initialised as follows:

$$ub_0 = Min\{c_i + \sum_{j=1}^{n} u_{i,j} | i = 1 \dots n\} \ ; \ lb_0 = Min\{c_i | i = 1 \dots n\}$$

Roughly, the upper bound corresponds to an allocation where the eventually unhappiest agent is given all the resources and the lower bound corresponds to an allocation where it is given no resource. Clearly, the value of the optimal egalitarian social welfare lies somewhere between those bounds. The imprecision of our estimation is equal to their difference.

Definition 5. *Let $k \in I\!N$. Let ub_k and lb_k denote estimated upper and lower bounds of sw_e^* at iteration k: $lb_k \leq sw_e^* \leq ub_k$. Let m_k be the mean of ub_k and*

lb_k, defined as $m_k = (ub_k + lb_k)/2$. A is a satisfying allocation *at iteration k iff its egalitarian social welfare is greater than m_k:*

$$sw_e(A) \geq m_k.$$

Assume now that the agents are endowed with reasoning capabilities that enable them to detect whether the set of satisfying allocations is empty or not. If this set is empty, i.e. the optimal egalitarian social welfare cannot be greater than the mean, then the upper bound can be updated with the value of the mean:

$$ub_{k+1} = m_k \ ; \ lb_{k+1} = lb_k \ ; \ m_{k+1} = (ub_{k+1} + lb_{k+1})/2$$

Otherwise, the optimal social welfare is at least equal to the mean and the lower bound can be assigned to the value of the mean:

$$ub_{k+1} = ub_k \ ; \ lb_{k+1} = m_k \ ; \ m_{k+1} = (ub_{k+1} + lb_{k+1})/2$$

In both cases, one of the bounds is updated and the estimation imprecision is divided by two. After k iterations, the imprecision on sw_e^* is divided by 2^k. The constructed sequences all converge to the same limit:

Theorem 1 (Convergence).

$$\underset{k\to\infty}{Lim}\, lb_k = \underset{k\to\infty}{Lim}\, ub_k = \underset{k\to\infty}{Lim}\, m_k = sw_e^*$$

Proof. The sequences $(lb_k)_{k\in\mathbb{N}}$ and $(ub_k)_{k\in\mathbb{N}}$ are adjacent sequences, respectively monotonic increasing and monotonic decreasing. This proves they converge to the same limit, so $(m_k)_{k\in\mathbb{N}}$ also converges and its limit is the same. The common value of these three sequences is obviously sw_e^*.

We therefore dispose of an efficient estimation procedure in which the imprecision converges towards zero at exponential speed. Note that theorem 1 can actually be strengthened. Indeed, in practice the agents would represent the utilities $u_{i,j}$ and the parameters denoted c_i by means of a finite number of digits d, i.e. with a precision of 10^{-d}. Then:

Theorem 2 (Fast termination). *The optimal egalitarian social welfare is computed after a number of dichotomous search steps equal to $floor(log_2 \frac{ub_0 - lb_0}{10^{-d}})+$ 1.*

Proof. Since sw_e^* is a sum of parameters represented with d digits it also has the same number of digits. The interval $I_k = [lb_k, ub_k]$ can contain a maximum of N_{max} such numbers, where $N_{max} = floor((lb - ub)/10^{-d})) + 1$. When N_{max} equals 1, the target sw_e^* can be precisely identified as the unique real number with d digits in this interval. As long as the length of I_k, i.e. $L(I_k) = ub_k - lb_k = 2^{-k}(ub_0 - lb_0)$, is strictly smaller than 10^{-d}, N_{max} is equal to 1. Basic calculus shows that this happens for the smallest integer k such that $k > log_2(\frac{ub_0 - lb_0}{10^{-d}})$.

3.2 Computing an allocation with optimal egalitarian social welfare

Having shown how to compute the value of the optimal egalitarian social welfare in a society, we now turn to how to best construct satisfying allocations for this value. These correspond to egalitarian allocations. We present a method whereby, at each iteration k of the dichotomous search, the agents will go through a multiple-phase process (up to n phases), by constructing groups of increasing size: the first group contains one agent, the second contains two agents, etc.

In the dichotomous search process, the set of satisfying allocations for the society, at any iteration, need not be explicitly constructed, as only the (non-)emptiness of the set of all satisfying allocations matters. Thus, trivially, the agents are allowed to restrict the search to any subset of the set of satisfying allocations obtained by applying an *invariance operator*, namely an operator on sets preserving the properties of emptiness and non-emptiness.

Then, basically, our idea is to use an invariance operator that reduces as much as possible the search space. We will use an invariance operator defined in terms of the following binary relation on allocations for groups:

Definition 6. *Let A and B be two allocations for a group G. Then $B \preceq A$ if and only if for all resources j, $1 \leq j \leq m$:*

$$\sum_{i,a_i \in G} B_{i,j} \leq \sum_{i,a_i \in G} A_{i,j}$$

The inequality means that if a resource j is allocated according to B then it is also allocated according to A. The relation $\preceq$ is reflexive and transitive. Also, not all allocations can be compared with this relation (i.e. $\preceq$ is not a total relation). Moreover, $\preceq$ is not anti-symmetrical, so $\preceq$ is not an order relation.

Definition 7. *Let A and B be two allocations for a group G. Then B* minors *A if and only if $B \preceq A$ and A is* equivalent *to B if and only if $B \preceq A$ and $A \preceq B$.*

When considering the satisfying allocations, we eliminate every allocation that is either minored by or equivalent to another one. We repeat these simplifications until a fix point is reached. This process enables to construct/implement a reduction operator with the property of being an invariance operator:

Definition 8. *Let S be a set of allocations. A* frugal reduction *$F(S)$ of S is a subset of S such that (i) any allocation in S is minored by a allocation in $F(S)$, and (ii) no allocation in $F(S)$ is minored by another one in $F(S)$.*

Note that frugal reductions are not guaranteed to be unique, but the frugal reduction operator has the merit of being an invariance operator.

Theorem 3 (Invariance of F). *The frugal reduction operator F is an invariance operator.*

Proof. Since $F(S) \subseteq S$ trivially holds for all S, the frugal reduction of the empty set is the empty set. If S is not empty, it contains an element that is minored by an element in $F(S)$, so $F(S)$ is not empty either.

The problem of determining an egalitarian allocation then amounts to determining whether the frugal reduction of the satisfying set for the final iteration of the dichotomous search, giving the optimal egalitarian social welfare, is empty or not and, if not, to find one of its elements. In fact, we compute all of them.

Note also that, by using frugal reductions, resources are not wasted since only minimal allocations are kept (see example in figure 1). This is particularly useful when resources are scarce or expensive because leftovers can be re-used for other allocations. Note that utilitarianism do not have this property: they systematically consume any available resource. In a nutshell, strict egalitarianism implicitly captures resource management policies.

$$F(\{\begin{pmatrix}1\,0\,1\,0\\0\,1\,0\,0\\0\,0\,0\,1\end{pmatrix},\begin{pmatrix}0\,1\,0\,0\\0\,0\,1\,0\\0\,0\,0\,1\end{pmatrix},\begin{pmatrix}0\,1\,0\,0\\0\,0\,1\,0\\1\,0\,0\,0\end{pmatrix},\begin{pmatrix}0\,0\,1\,0\\0\,1\,0\,0\\1\,0\,0\,0\end{pmatrix}\}) = \{\begin{pmatrix}0\,1\,0\,0\\0\,0\,1\,0\\0\,0\,0\,1\end{pmatrix},\begin{pmatrix}0\,1\,0\,0\\0\,0\,1\,0\\1\,0\,0\,0\end{pmatrix}\}$$

Fig. 1. The frugal reduction operator filters both redundancies (superfluous agreements) and allocations that over-consume resources (inefficient solutions). The agents save memory and time and the society manages its resources better (here either r_1 or r_4 is preserved).

Below, the term *(minimal collection of) agreements* for a group G, denoted $Ag(G)$, will stand for a frugal reduction of the set of satisfying allocations for G. Now we present an efficient decision-making procedure to build minimal collections of agreements for the full set of agents at each iteration of the dichotomous search, via a multi-phase process whereby agents progressively join in, starting from an initial group consisting of a single agent. At each phase, a minimal collection of agreements is built for the current group, if possible, and, when a new agent joins the group, the prior set of agreements is revised to provide a minimal collection for the newly formed group. If no agreements can be found the search is abandoned. In order to collect the minimal collection of agreements for a group to which a new agent has been added, we will use sets of trees, or in other words forests.

Suppose an agent $a_{i'}$ wants to start a new group or join an existing group $G = \{a_{i_1}, a_{i_2}, \ldots, a_{i_g}\}$ to form the new group $G' = \{a_{i'}\}$ or $G' = G \cup \{a_{i'}\}$, respectively. In order to build the minimal collection of agreements for G', $a_{i'}$ constructs a forest of trees whose nodes are pairs of the form $(Z, w(Z))$, where Z is a *fuzzy allocation* for G' and $w(Z)$ is its welfare, defined below.

Definition 9. *A* fuzzy allocation *is a table* $Z = ((z_{i,j}))_{(g+1)\times m}$ *with* $g+1$ *lines and* m *columns and whose coefficients* $z_{i,j}$ *belong to* $\{1, 0, -1\}$*:*

$$F = \begin{pmatrix} i_1 : z_{i_1,1} \; z_{i_1,2} \; \cdots \; z_{i_1,m} \\ \cdots \;\; \cdots \;\;\; \cdots \; \cdots \; \cdots \\ i_g : z_{i_k,1} \; z_{i_k,2} \; \cdots \; z_{i_g,m} \\ i' : z_{i',1} \; z_{i',2} \; \cdots \; z_{i',m} \end{pmatrix}$$

The set of allocations encoded by a fuzzy allocation Z *is the set of allocations for* G' *according to which each agent* a_i *in the group* G' *gets* r_j *if* $z_{i,j} = 1$ *and does not get* r_j *if* $z_{i,j} = -1$.

The coefficients equal to 0 in a fuzzy allocation leave the information as to which agents gets the corresponding resource unspecified.

Definition 10. *The* signature of a fuzzy allocation Z *is obtained by replacing in* Z *all the coefficients equal to* -1 *by* 0.

Intuitively, the signature of a fuzzy allocation is the allocation in the set encoded by Z that allocates fewest resources.

Definition 11. *The* welfare of a fuzzy allocation Z*, denoted* $w(Z)$*, is the egalitarian social welfare of the signature of* Z*. If* $w(Z)$ *is greater than* m_k*, at some iteration* k*, then* Z *is said to be* satisfying*.*

Definition 12. *A node* $(Z, w(Z))$ *in a tree is called*

- positive *iff* $w(Z) \geq m_k$ *(i.e.* Z *is satisfying)*
- open *iff it is not positive but the allocation in the set encoded by* Z *in which all the resources not used by the agents in* G *are used by the new agent* $a_{i'}$ *is satisfying (if* $G' = \{a_{i'}\}$ *any resource can be used)*
- negative *iff it is neither positive nor open.*

The trees are constructed as follows. The roots of the trees constituting the forest of a phase are constructed from the positive leaves of the trees in the previous phase. More precisely, $Ag(G')$, the minimal collection of agreements for G', is computed from the positive leaves of the trees in the forest for G' (as we will see below). The roots of the trees in the forest for G' are pairs $(Z, w(Z))$ where the first g lines of Z take their values in (one of) the agreements in $Ag(G)$ and all the coefficients in the last line (corresponding to the newly added agent $a_{i'}$) are equal to zero. Negative and positive nodes have no children, only open nodes do. Consider an open node $N = (Z, w(Z))$. Let j_0 be the index of a resource r_j that $a_{i'}$ could use, i.e. $z_{i',j_0} = 0$ and that does not have a null utility. Such an index exists since the node is open. Then the left and right children of N, denoted $(Z_L, w(Z_L))$ and $(Z_R, w(Z_R))$, respectively, are defined as follows:

$$Z_{L;i',j_0} = 1,\ Z_{R;i',j_0} = -1 \textit{ and } \forall j \neq j_0 :\ Z_{L;i',j} = Z_{R;i',j} = Z_{i',j}$$

The agents build the tree by constructing the descendants of all open nodes. Thus, the trees all have a strictly binary structure. The process terminates finitely because there is a finite number of resources. In fact, the depth of a tree is bounded by the number of resources $a_{i'}$ can use.

Figures 2 and 3 illustrate the construction of trees in a forest, for our running example at step $k = 1$, where the allocations must have a welfare greater than $m_1 = 1.225$. Here and in the rest of the paper, we ignore the index of agents in fuzzy allocations if clear from the context.

We will refer to fully constructed trees (namely trees whose only leaves are positive or negative nodes) as *frugal trees*. Frugal trees have an interesting property: their leaves 'hide' a frugal reduction of their root:

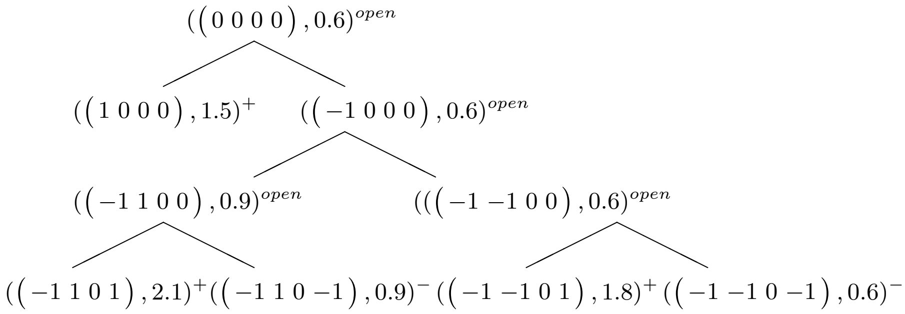

Fig. 2. a_1 finds first three satisfying allocations (1 0 0 0), (0 1 0 1) and (0 0 0 1). The second one, which is minored by the third one, will be eliminated by frugal reduction.

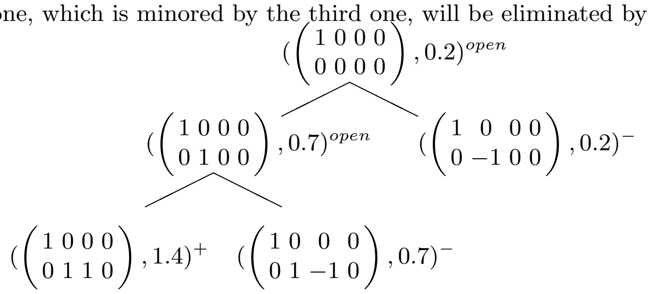

Fig. 3. a_2 examines the first allocation found by a_1 and this search leads to one agreement between the two agents: a_1 takes r_1 and a_2 takes r_2 and r_3.

Theorem 4 (Frugal tree). *Given a fuzzy allocation Z, let L be the set of positive leaves of a frugal tree with root $(Z, w(Z))$, and let S be the set of signatures for all elements of L. Then, there exists a frugal reduction $F(\Sigma)$ of the set Σ of satisfying allocations encoded by Z such that $F(\Sigma) \subseteq S$.*

Proof. By construction of the tree, the union of the sets encoded by its leaves is equal to the set encoded by the root. A property of negative nodes is that they encode sets that do not contain satisfying allocations. So, all the satisfying allocations are in the union of the positive nodes (recall that open nodes are not leaves). By $\preceq$-minimality of the elements in the frugal reduction, they can only be the allocation in the positive node consuming least resources, i.e. be signatures of positive nodes. Hence the inclusion.

Applying the frugal reduction operator after having collected the signatures for the leaves enables the agents to ignore superfluous agreements. The reason why we do not lose any useful agreement by working only on the positive nodes is justified by the following lemma, where the role of the set of signatures is played by S and the set of satisfying allocations for the root is played by Σ:

Lemma 1 (Elimination). *If $F(\Sigma) \subseteq S \subseteq \Sigma$ then $F(S) = F(\Sigma)$ (namely, a frugal reduction of S is also one of Σ).*

Proof. Any allocation in Σ is minored by an allocation in $F(\Sigma)$ which in turn is minored by an allocation in $F(A)$. No allocation in $F(A)$ is minored by another in $F(A)$.

This lemma justifies the idea that the agents should apply the reduction operator on the positive leaves of the frugal trees so as to minimise the computational effort in the next phase, by minimising the number of trees to explore/construct and therefore save both time and memory.

By virtue of the following theorem, harvesting the positive leaves of a frugal tree and filtering through the frugal reduction operator, one ends up with a minimal collection of agreements for the new group.

Theorem 5 (Agreements coverage). *There exists a frugal reduction of the minimal collection of agreements for G' that is included in $F(X)$, where X is the union of all positive nodes in all trees in the forest constructed for G'.*

Proof. We will prove that indeed, the frugal reduction of the union is also a frugal reduction of the satisfying agreements for G'. We simply check that the two points defining a frugal reduction hold. The second point is trivial since the frugal reduction of the union is a frugal reduction. The first point holds because we started from a complete set of minimal agreements for G and have taken a complete set of their minimal extensions.

As an illustration of the overall mechanism, consider our running example with three agents, four resources and the tables in section 2. Then

– at iteration $k = 0$: $lb_0 = 0.2$ $ub_0 = 1.9$ $m_0 = 0.55$

$$Ag(\{1\}) = \{\begin{pmatrix} 0\,0\,0\,0 \end{pmatrix}\}\ ;\ Ag(\{1,2\}) = \{\begin{pmatrix} 0\,0\,0\,0 \\ 0\,1\,0\,0 \end{pmatrix}, \begin{pmatrix} 0\,0\,0\,0 \\ 0\,0\,1\,0 \end{pmatrix}, \begin{pmatrix} 0\,0\,0\,0 \\ 1\,0\,0\,1 \end{pmatrix}\};$$

$$Ag(\{1,2,3\}) = \{\begin{pmatrix} 0\,0\,0\,0 \\ 0\,1\,0\,0 \\ 1\,0\,0\,0 \end{pmatrix}, \begin{pmatrix} 0\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,1\,0 \end{pmatrix}, \begin{pmatrix} 0\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,0\,1 \end{pmatrix} \begin{pmatrix} 0\,0\,0\,0 \\ 0\,0\,1\,0 \\ 1\,0\,0\,0 \end{pmatrix}, \begin{pmatrix} 0\,0\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,1 \end{pmatrix}\}$$

– at iteration $k = 1$: $lb_1 = 0.55$ $ub_1 = 1.9$ $m_1 = 1.225$

$$Ag(\{1\}) = \{\begin{pmatrix} 1\,0\,0\,0 \end{pmatrix}, \begin{pmatrix} 0\,0\,0\,1 \end{pmatrix}\}\ ;\ Ag(\{1,2\}) = \{\begin{pmatrix} 1\,0\,0\,0 \\ 0\,1\,1\,0 \end{pmatrix}, \begin{pmatrix} 0\,0\,0\,1 \\ 0\,1\,1\,0 \end{pmatrix}\};$$

$$Ag(\{1,2,3\}) = \{\begin{pmatrix} 1\,0\,0\,0 \\ 0\,1\,1\,0 \\ 0\,0\,0\,1 \end{pmatrix}\}$$

– at iteration $k = 2$: $lb_2 = 1.225$ $ub_2 = 1.9$ $m_2 = 1.5625$

$$Ag(\{1\}) = \{\begin{pmatrix} 0\,0\,0\,1 \end{pmatrix}\}\ ;\ Ag(\{1,2\}) = \{\begin{pmatrix} 0\,0\,0\,1 \\ 1\,1\,1\,0 \end{pmatrix}\}\ ;\ Ag(\{1,2,3\}) = \{\}$$

– at iteration $k = 3$: $lb_3 = 1.225$ $ub_3 = 1.5625$ $m_3 = 1.39225$

$$Ag(\{1\}) = \{\begin{pmatrix} 1\,0\,0\,0 \end{pmatrix}, \begin{pmatrix} 0\,0\,0\,1 \end{pmatrix}\}\ ;\ Ag(\{1,2\}) = \{\begin{pmatrix} 1\,0\,0\,0 \\ 0\,1\,1\,0 \end{pmatrix}, \begin{pmatrix} 0\,0\,0\,1 \\ 0\,1\,1\,0 \end{pmatrix}\};$$

$$Ag(\{1,2,3\}) = \{\}$$

– at iteration $k = 4$: $lb_4 = 1.225$ $ub_4 = 1.39225$ $m_4 = 1.308625$

$Ag(\{1\}) = \{\begin{pmatrix}1\,0\,0\,0\end{pmatrix}, \begin{pmatrix}0\,0\,0\,1\end{pmatrix}\}$; $Ag(\{1,2\}) = \{\begin{pmatrix}1\,0\,0\,0\\0\,1\,1\,0\end{pmatrix}, \begin{pmatrix}0\,0\,0\,1\\0\,1\,1\,0\end{pmatrix}\}$
$Ag(\{1,2,3\}) = \{\}$

– at iteration $k = 5$: $lb_5 = 1.225$ $ub_5 = 1.308625$ $m_5 = 1.2668125$

$Ag(\{1\}) = \{\begin{pmatrix}1\,0\,0\,0\end{pmatrix}, \begin{pmatrix}0\,0\,0\,1\end{pmatrix}\}$; $Ag(\{1,2\}) = \{\begin{pmatrix}1\,0\,0\,0\\0\,1\,1\,0\end{pmatrix}, \begin{pmatrix}0\,0\,0\,1\\0\,1\,1\,0\end{pmatrix}\}$
$Ag(\{1,2,3\}) = \{\begin{pmatrix}1\,0\,0\,0\\0\,1\,1\,0\\0\,0\,0\,1\end{pmatrix}\}$

At this point the algorithm terminates (as $ub_5 - lb_5 < 10^{-1}$) correctly computing A^* given in section 2.

4 Consensus search and coordination heuristics

Two issues have been left open regarding the computation of egalitarian allocations: (i) the choice of resource indexes for splitting open nodes (in the *search for consensus*) and (ii) the election criterion for an agent to join the current group (referred to as *coordination* criterion). In this section we provide solutions to these issues in an experimental setting, as they play no role at the theory level. The solutions will be stated in the form of heuristics.

The way an agent performs consensus search is determined by a node-splitting strategy, i.e. a rule for choosing the resource index on which to split an open node in the search tree. In our experiments, the reference strategy consists in following the order of appearance of the (yet un-allocated) resources in their natural order r_1, r_2, r_3, etc. We call this the *random node strategy (RN)*. This strategy is compared to two other strategies based on resource preferences. These strategies are called *least-useful strategy (LU)* and *most-useful strategy (MU)*. The splitting index chosen when following LU (resp. MU) is the one of the unused resource having the lowest (resp. greatest) utility for the new agent.

Coordination is determined by the election criterion/order applied for selecting the next agent to join the group and enter in negotiation with it. The simplest order, that we call *random order (RO)*, is the natural order of the agents $a_1, a_2, ..., a_n$. This criterion serves as a reference for performance comparisons. We define two other orders, based on the welfare of the agents, which enables to order them socially. The *lowest welfare strategy (LW)* gives priority to the remaining agent with lowest welfare, whereas the *highest welfare strategy (HW)* gives priority to the remaining agent with highest welfare.

Figure 4 summarises some experiments comparing, in the first three columns, the consensus search strategies (RN, LU and MU) while keeping the coordination heuristic fixed to RO, and, in the last two columns, the social orders LW and HW for coordination, while keeping MU for the consensus search. We evaluate in seconds the average computational time needed to solve randomly generated problems whose sizes are indicated in the (n, m) column. The number of agents is taken equal to the number of resources and this number varies from 1 to 7.

(n,m)	$RO-RN$	$RO-LU$	$RO-MU$	$HW-MU$	$LW-MU$
$(1,1)$	0.0945	0.1065	0.1010	0.0110	0.0060
$(2,2)$	0.3635	0.4075	0.3145	0.0245	0.0180
$(3,3)$	0.9600	1.1345	0.8640	0.0320	0.0365
$(4,4)$	2.4090	2.8825	2.0175	0.1000	0.1110
$(5,5)$	7.4655	9.6015	6.0310	0.2450	0.1405
$(6,6)$	20.7670	27.6690	15.6435	0.8785	0.7125
$(7,7)$	74.6614	105.4960	51.6125	4.4810	2.2935

Fig. 4. Influence of the nodes-splitting strategy for social consensus (RN, LU, MU) and the agents selection order for coordination (RO, HW, LW) on the total decision-making time. The combined LW-MU heuristic divides the negotiation time by nearly 30.

The table U and coefficients c_i are generated according to a uniform distribution between 0 and 1. We use 2 digits of precision for the utilities ($d = 2$). The computational time is averaged over 20 problems for each dimension and the different strategies are systematically assessed with respect to the same cases for fair comparisons. The experiments have been carried with Maple 10 on a 1.07 GHz G4 processor. The first three columns show that for a fixed random order of negotiation, following the MU strategy gives the best results. When splitting open nodes with respect to most useful resources first, the depth of the search tree's branches is minimised. Then keeping the MU strategy (last two columns), we discover that using monotonic social orders to coordinate negotiations leads to solutions very fast, with a slight superiority of the increasing order over the decreasing one. The agents that must join the group first are those with lowest welfare (LW). When these two best strategies are combined (LW-MU), the improvement is considerable: the total time required for the negotiations is divided by nearly 30.

5 Conclusion

We presented a sound method that guarantees agents to find an allocation of resources that exactly maximises the egalitarian social welfare of the society they constitute. The method relies upon a dichotomous search terminating after a 'small' number of steps. In the search process, agents examine and update the value of the optimal egalitarian social welfare that can be collectively achieved given their personal preferences, expressed in terms of utilities they assign to sets of resources. Our method uses binary search trees and forests of Boolean fuzzy allocations as well as a frugal reduction operator that simplifies the reasoning process of the agents by eliminating appropriately any superfluous agreements they might come up with. The solutions are efficient as far as they never over-consume resources.

The proposed mechanism allows consensus to be found with 'minimal' disclosure of information about the agents' preferences. Also, the mechanism can be nicely be distributed over the agents, with important computational advantages: the agents themselves carry the computational burden and the agents' master is relieved of all supervising work.

We proved empirically that the agents reason collectively much faster when giving priority to the most useful resources and can efficiently coordinate the sequence of their negotiations by using monotonic increasing social orders.

Dall'Aglio and Maccheroni [10] recently proved the existence of fair divisions between agents in the case of strongly subadditive and strongly continuous utility functions. In this paper, we have assumed additivity (semi-linearity) of the utility functions but have considered a finite set of indivisible resources. As Golovin [12] puts it: 'little is known about the computational aspects of finding [...] fair allocations [...] with indivisible goods' and 'early work in operations research focused on special cases that are tractable, or on exponential time algorithms for general models [13]'. In this family of models [13, 14], resources are allocated to activities, not agents. Under such formalism, the problem comes down to determining the appropriate levels of activity (represented as simple scalar variables). But when resources are allocated to agents, this notion collapses and one is led to handle vectorial variables with components for each resource, as in our case. Hence, although those models and our own all aim at solving the same ultimate application, they are not formalised as equivalent mathematical problems. Our intuition is that agents should first be assigned to tasks considering their capabilities (e.g. via coalition formation [8]) and then be allocated the resources.

The computational aspects of fair allocations of indivisible goods have been studied by [11]. This work however differs from ours in that in [11] fairness is achieved by minimising envy. According to Brams and King [15], 'while envy may be ineradicable if one desires to help the worst off, it is not clear that abandoning the maximin criteria to avoid it is a better alternative'.

[12] and [16] investigated the complexity of finding fair allocations of indivisible goods. [17] considered the problem of finding approximate max-min fair allocations for agents with additive utilities. In this paper, we have given a new negotiation mechanism for solving in a distributed manner, and without any approximation, indivisible resource allocation problems in the extended case of semi-linear utility functions.

Future work will be dedicated to the design of protocols and policies for implementing the allocation mechanism in distributed settings, to the study of complexity and scalability issues and to an experimental comparison to other methods.

Acknowledgements

This work was partially funded by the Sixth Framework IST programme of the EC, under the 035200 ARGUGRID project. The second author has also been supported by a UK Royal Academy of Engineering/Leverhulme Trust senior fellowship.

References

1. K. J. Arrow: Social Choice and Individual Values. John Wiley and Sons (1963)
2. A. K. Sen: Collective Choice and Social Welfare. Holden Day (1970)
3. H. Moulin: Axioms of Cooperative Decision Making. Cambridge University Press (1988)
4. J. Rawls: A Theory of Justice. Harvard University Press (1971)
5. J. C. Harsanyi: Can the Maximin Principle Serve as a Basis for Morality? American Political Science Review **86:2** (1996) 269–357
6. U. Endriss, N. Maudet, F. Sadri and F. Toni: Resource Allocation in Egalitarian Agent Societies. MFI (2003) 101–110
7. T. Sandholm: Algorithms for Optimal Winner Determination in Combinatorial Auctions. Artificial Intelligence **135** (2002)
8. O. Shehory and S. Kraus: Methods for task allocation via agent coalition formation. Artificial Intelligence **101:1–2** (1998) 165–200
9. U. Endriss, N. Maudet, F. Sadri and F. Toni: Negotiating Socially Optimal Allocations of Resources. Journal of Artificial Intelligence Research **25** (2006) 315–348
10. M. Dall'Aglio and F. Maccheroni: Fair division without additivity. The American Mathematical Monthly **112** (2005)
11. R. Lipton, E. Markakis, E. Mossel and A. Saberi: On approximately fair allocations of indivisible goods. Proceedings of EC'04 (2004)
12. D. Golovin: Max-Min Fair Allocations of Indivisible Goods. CMU-CS-05-144 (2005)
13. H. Luss: On equitable resource allocation problems: A lexicographic minimax approach. Operations Research **47:3** (1999) 361–378
14. G. Yu: On the max-min 0-1 knapsack problem with robust optimization applications. Operations Research **44** (1996) 407–415
15. S.J. Brams and D.L. King: Efficient Fair Division: Help the Worst Off or Avoid Envy? Rationality and Society **17:4** (2005) 387-421
16. S. Bouveret, M. Lemaître, H. Fargier and J. Lang: Allocation of indivisible goods: a general model and some complexity results. AAMAS (2005) 1309–1310
17. I. Bezakova and V. Dani: Allocating indivisible goods. ACM SIGecom Exchanges **5:3** (2005) 11–18
18. M. Lemaître, G. Verfaillie and N. Bataille: Exploiting a Common Property Resource under a Fairness Constraint: a Case Study. IJCAI (1999) 206–211
19. S. Bussmann: Agent-Oriented Programming of Manufacturing Control Tasks. ICMAS (1998) 57–63
20. S. Bussmann and K. Schild: Self-Organizing Manufacturing Control: An Industrial Application of Agent Technology. ICMAS (2000) 87–94
21. R. Babiceanu and F. Chen: Development and Applications of Holonic Manufacturing Systems: A Survey. Journal of Intelligent Manufacturing **1:17** (2006) 111–131