The distributed negotiation of egalitarian resource allocations

Paul-Amaury Matt, Francesca Toni and Dionysis Dionysiou

Abstract

We provide a sound theory for the computation of allocations of indivisible resources amongst cooperative agents, maximising the egalitarian social welfare of the overall multi-agent system, seen as a society. Agents' preferences over resources are captured by scalar utilities that we sum up to define the agents' individual welfare. The egalitarian social welfare is defined as the minimal individual welfare across the society.

From the proposed theory we derive a mechanism of negotiation distributed over the agents. This mechanism is defined by means of a public communication protocol and a private computational policy that have the advantage of integrating efficient coordination and computational heuristics.

1 Introduction

Equity and fairness [17] are social, economic and philosophical notions that can be transposed to artificial societies and serve as a basis for the design of complex agents systems [1].

The problem of reallocating resources amongst agents within a multi-agent systems can be understood as the problem of identifying socially optimal allocations of resources amongst the agents, by interpreting multi-agent systems as societies [8]. In this setting, allocations may be understood as fair if they are *egalitarian* [8], namely if these allocations render the least "well-off" agents in the society as "better-off" as possible, in terms of the individual welfare they obtain from the resources allocated to them.

In this paper, we are concerned with the computation of fair allocations of indivisible resources amongst cooperative agents in a society, where fairness is given this egalitarian interpretation. In particular, we provide a distributed mechanism for computation of egalitarian allocations, whereby agents in a distributed platform share the burden of the computation.

Maximising the egalitarian social welfare by allocating indivisible resources is a hard global optimisation problem, characterised by a discrete domain of exponential size, on which constraints exist and in which a non-linear and non-derivable function is to be optimised. Well known global optimisation and constraint satisfaction techniques (see [15] for a recent survey) cannot be applied.

As Golovin recently put it (see [9] and references therein): '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'. The computational aspects of fair allocations of indivisible goods have been studied by [11], but in that work fairness is achieved by minimising envy. Golovin [9] provides approximation algorithms for maximising the egalitarian social welfare, along with some complexity results (see also [2]). In the operations research community [12, 21] resources are allocated to activities instead of agents resulting in a different and simpler problem with fewer variables (linear instead of quadratic).

Endriss et al [8] prove that any sequence of strongly equitable deals (defined therein) will eventually result in an egalitarian allocation of indivisible goods. However, this is a purely theoretical result which provides no indication to agents designers on how to compute these deals and thus the allocations. Also, in the Mathematics community, advanced existence results have been provided [4] concerning fair sharing problems (where additivity of utilities of resources is not assumed). However, also these results do not address the problem of constructing optimal allocations.

In this paper we give a new negotiation mechanism for solving distributedly and without approximation the problem of allocating indivisible resources amongst cooperative agents whose preferences are modeled in terms of semilinear utility functions. This mechanism is based upon the algorithm described in [13], and is defined in terms of a communication protocol, formalised along the lines of [7], and a communication policy, formalised along the lines of [18].

2 Preliminaries

In this paper, we 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. The numbers of agents (n) and resources (m) are assumed to be strictly positive. We also assume that the resources are indivisible, so that each resource may be allocated to one agent at most. We will thus use the following definition of allocation of indivisible resources to agents.

Let $E_k = \{a_{i_1}, \ldots, a_{i_k}\}$ represent a group of k agents in the society [1]. An *allocation of resources* to E_k is a Boolean table of k lines and m columns:

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

such that A contains at most one element=1 per column. Given $a_i \in E_k$, we say that a_i *gets* r_j if and only if $A_{i,j} = 1$.

In our framework, agents in a multi-agent systems are abstractly characterised by their own preferences concerning the resources. These preferences are given by means of a *utility table*, defined as a matrix $U = ((U_{i,j}))_{n\times m}$ with n lines and m columns of real valued, positive coefficients. 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,

[1] Note that the entire society is given by E_n.

measuring the contribution of the resource to the agent's welfare. Each agent need only be aware of its own preferences, namely its own line in the utility table.

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: 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 is a real valued, positive coefficient, representing the welfare of agent a_i prior to any allocation of resources.

Let us now introduce an optimality criterion on allocations, borrowed from the areas of social choice theories [1, 19, 14] and welfare economics [17, 10, 6] and having an *egalitarian* flavour. We are after allocations that maximise the egalitarian social welfare, defined metaphorically as the welfare of the "unhappiest" or least "well-off" agent in the system. Formally, the *egalitarian social welfare* of an allocation A to the entire society E_n is:

$$sw_e(A) = Min\{w_i(A) | i = 1, \ldots, n\}$$

An *egalitarian allocation* is an allocation A^* maximising the egalitarian social welfare.

When building an egalitarian allocation, two problems need to be solved at once: a) finding the value sw_e^* of the optimal egalitarian social welfare and b) actually finding an egalitarian allocation, with social welfare sw_e^*. To solve the first problem, one can perform a dichotomous search. To solve the second problem, the agents will have to reason about sets of allocations, that we encode using *fuzzy allocations*, defined below. A *fuzzy allocation* F *to* E_k is a table with k lines, m columns and whose coefficients $f_{i,j}$ belong to $\{-1, 0, 1\}$:

$$F = \begin{pmatrix} i_1 : & f_{i_1,1} & f_{i_1,2} & \ldots & f_{i_1,m} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ i_k : & f_{i_k,1} & f_{i_k,2} & \ldots & f_{i_k,m} \end{pmatrix}$$

A fuzzy allocation F to E_k encodes the set of allocations to E_k according to which each agent a_i in the group gets r_j if $f_{i,j} = 1$ and does not get r_j if $f_{i,j} = -1$. The coefficients equal to 0 correspond to unspecified information about the allocation of the corresponding resources, and are the reason why fuzzy allocation do not simply denote singletons, but really sets.

We also define the *signature* $s(F)$ *of a fuzzy allocation* F as the allocation in the set encoded by F that allocates fewest resources. This allocation is obtained by replacing in F all the coefficients equal to -1 by 0.

The *social welfare corresponding to a fuzzy allocation* F, denoted $w(F)$, is the egalitarian social welfare of the signature of F, defined over E_k.

3 Computational strategy

In this section we revise the method of [13] that a) uses dichotomous search for finding the value sw_e^* of the optimal egalitarian social welfare and b) uses frugal reductions of allocations and fuzzy allocations for actually finding an egalitarian allocation, with social welfare sw_e^*.

Dichotomy is a simple and elegant mechanism guaranteeing arbitrary precision and enabling fast estimation of the optimal social welfare. In our dichotomous search, lower (L) and upper (U) bounds for this optimal value are updated iteratively. The upper bound corresponds to an allocation where after the allocation the unhappiest agent is given all the resources and the lower bound corresponds to an allocation where after the allocation the unhappiest agent is given no resource. Clearly, the value of the optimal egalitarian social welfare lies somewhere between those bounds. These are initialised as follows:

$$L = Min\{c_i \mid i = 1 \ldots n\},\ U = Min\{c_i + \sum_{j=1}^{m} u_{i,j} \mid i = 1 \ldots n\}$$

Assuming agents are endowed with an appropriate mechanism for checking the non-emptiness of the set of allocations with social welfare higher than an arbitrary value (the mean of the bounds), dichotomous search algorithm 1 can be used to determine in finite time the exact value of sw_e^*. Our only assumption here is that all agents internally represent their preferences $u_{i,j}$ with d digits of precision. The optimal egalitarian social welfare is rapidly found after $\texttt{floor}(\log_2 \frac{U-L}{10^{-d}}) + 1$ cycles only.

Algorithm 1 Dichotomous search. Inputs: precision d in digits, lower and upper bounds for sw_e^*. Output: optimal social welfare sw_e^*.

1: **repeat**
2:   **if** $\{A | sw_e(A) \geq (L+U)/2\} \neq \emptyset$ **then**
3:     $L \leftarrow (L+U)/2$
4:   **else**
5:     $U \leftarrow (L+U)/2$
6:   **end if**
7: **until** $U - L < 10^{-d}$
8: **return** round (L+U)/2 with d digits

The check at line 2 of the algorithm is highly complex, as the space of possible allocations is of exponential size $(n+1)^m$. We now discuss how to best handle this check. Basically, our idea is to use a space reduction operator that both eliminates inefficient allocations and redundancies. Indeed, after all, given L and U, all the agents need to do is find out if they can come up with some allocation A such that $sw_e(A) \geq (L+U)/2$.

The operator's definition is based on a special binary relation between pairs of allocations. Let A and B be two allocations to E_k. We say that B *minors* A

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

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

Figure 1: The frugal reduction operator F eliminates both redundancies (superfluous agreements) and inefficient allocations (over-consuming resources). The agents save memory and computational time and the society manages its resources better (here either resource r_1 or r_4 can be preserved).

and write $B \preceq A$ if and only if

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

The intuitive meaning of $B \preceq A$ is that whatever resource is allocated according to B, it is also allocated according to A. When considering sets of allocations for all of which $w(A) \geq (L+U)/2$ holds, the agents may perfectly treat non-minimal allocations as superfluous. Also, when two such allocations minor each other, one can be eliminated to avoid redundancy. This defines our reduction operator. Let S be a set of allocations for E_k. A *frugal reduction* $F(S)$ of S is a subset of S such that

- any allocation in S is minored by an allocation of $F(S)$
- no two allocations in $F(S)$ minor each other.

Note that frugal reductions are not guaranteed to be unique, but the frugal reduction operator has a remarkable property: $S \neq \emptyset \Leftrightarrow F(S) \neq \emptyset$.

Intuitively, we can forsee that $F(S)$ is statistically much smaller than S itself (cf figure 1), so in a way, using frugal reductions simplifies the search process for allocations. Moreover, frugal reductions can be computed using an incremental negotiation mechanism summarised in algorithm 2. At each step k, one new agent joins a group E_k, forcing a revision of the set of agreements amongst these prior agents. The newly formed group then eliminates superfluous agreements using a frugal reduction or abandons the search step when no agreements can be found (fail).

In order to build the minimal collection of agreements for a group to which a new agent has just been added, we consider a forest (set of trees). In each phase, specific leaves (termed *positive leaves*) of the trees in a forest constitute a collection (not yet minimal) of agreements for the group. The roots of the trees constituting the forest of a phase are simply the signatures of the positive leaves of the forest in the previous phase.

Algorithm 2 Incremental construction of a frugal reduction $F_n(x)$ of $\{A|sw_e(A) \geq x\}$. Input: x. Output: $F_n(x)$.

$E_0 \leftarrow \{\}; k \leftarrow 0; F_0(x) \leftarrow \{\}$
repeat
  $k \leftarrow k + 1$
  $E_k \leftarrow E_{k-1} \cup \{k\}$
  a_k tries to find a consensus Ext_k with the prior agents (E_{k-1}) of welfare at least equal to x
  if $Ext_k \neq \emptyset$ (i.e. a consensus can be found) **then**
    $F_k(x) = F(Ext_k)$ (reduce the set frugally)
  else
    return $\emptyset$ (failure)
  end if
until $k = n$
return $F_n(x)$

Suppose an agent $a_{i'}$ wants to join a group $G = \{a_{i_1}, a_{i_2}, \ldots, a_{i_k}\}$ to form the group $G' = \{a_{i_1}, a_{i_2}, \ldots, a_{i_k}, a_{i'}\}$. The group G then starts constructing a new forest whose trees' nodes N are pairs of the form $(F, w(F))$ where F is a fuzzy allocation for G'.

The root of any tree in the constructed forest at iteration $k + 1$ is a pair $(F, w(F))$ where the first k lines of F take their values in $Ag(G)$ (the minimal collection of agreements for G), where G is the group of agents at iteration k (consisting of k agents), and all the coefficients in the last line (corresponding to the newly added agent $a_{i'}$) are equal to zero. The trees are constructed top-down from their root and all have a strictly binary structure.

A node $(F, w(F))$ in a tree is called

- *positive* iff F is satisfying, i.e. $w(F) \geq (L + U)/2$
- *open* iff it is not positive but the allocation in the set encoded by F in which all the resources not used by an agent in G are used by the new agent $a_{i'}$ is satisfying
- *negative* iff it is neither positive nor open.

Negative and positive nodes have no children, only open nodes do.

Consider an open node $N = (F, w(F))$. Let j_0 be the index of a resource r_j that $a_{i'}$ could use, i.e. $f_{i',j_0} = 0$. Such an index exists since the node is open. Then the left and right children of N, denoted $(F_L, w(F_L))$ and $(F_R, w(F_R))$, are defined as follows:

$$\begin{cases} f_{L;i',j_0} = 1, \\ f_{R;i',j_0} = -1, \\ \forall j \neq j_0 : \; f_{L;i',j} = f_{R;i',j} = f_{i',j} \end{cases}$$

The agents build the tree by constructing the descendants of all the open nodes cf figure 2. The process terminates finitely because their is a finite number of resources. In fact, the depth of a tree is equal to the number of resources $a_{i'}$ can use.

$$(\begin{pmatrix} 0 & 0 & 0 \end{pmatrix}, 0)^{open}$$

$$(\begin{pmatrix} 1 & 0 & 0 \end{pmatrix}, 0.5)^{+} \quad (\begin{pmatrix} -1 & 0 & 0 \end{pmatrix}, 0)^{open}$$

$$(\begin{pmatrix} -1 & 1 & 0 \end{pmatrix}, 0.3)^{open} \quad (\begin{pmatrix} -1 & -1 & 0 \end{pmatrix}, 0)^{-}$$

$$(\begin{pmatrix} -1 & 1 & 1 \end{pmatrix}, 0.4)^{+} \quad (\begin{pmatrix} -1 & 1 & -1 \end{pmatrix}, 0.3)^{-}$$

$$(\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, 0)^{open}$$

$$(\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, 0)^{open} \quad (\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix}, 0)^{open}$$

$$(\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 1 \end{pmatrix}, 0)^{open} \quad (\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & -1 \end{pmatrix}, 0)^{open}$$

Figure 2: Agent a_1 finds two agreements (top tree). From the first one, agent a_2 derives (bottom tree) two possible agreements that satisfy them both. The agent pair finds a consensus in which a_1 gets r_1 and a_2 either r_2 or r_3.

The trees thus constructed have the interesting property that the frugal reduction of the set of satisfying sub-allocations in a fuzzy sub-allocation F is included in the set of signatures of the frugal tree whose root is F.

Applying the frugal reduction operator after having collected a tree's signatures is advocated as it enables the agents to ignore any superfluous agreements. The reason why we do not loose any useful agreement by working only on the positive nodes signatures is slightly technical and justified by the following result, where the role of the signatures set is played by A and the satisfying set of allocation in the root is played by Σ:

if $F(\Sigma) \subseteq A \subseteq \Sigma$ then $F(A) = F(\Sigma)$ (namely, a frugal reduction of A is also one of Σ).

The order in which the agents join in the group is an order that coordinates group negotiations. We have noticed in [13] that *social orders*, i.e. orders derived from welfare metrics, can have a strong (positive) impact on the time complexity of the negotiations. In particular, it is important to order the agents in increasing level of initial welfare. When the unhappiest agents think first about the resources they need, the detection of impossibility to find a common consensus is made earlier thus saving negotiation time. Also, since unhappiest agents tend to consume more resources than the others, they leave the others with a more restricted choice, which simplifies their reasoning task. We refer to this heuristic as LW.

When an agent constructs a search tree, it is important to minimise the depth of the tree. A good heuristic for that consists in splitting open nodes by thinking about the most useful resource that remains available to the agent. Indeed, this increases the probability that the left sub-tree is simply a positive leaf. When this heuristic (that we refer to as MU) and the earlier LW heuristic are combined, the total negotiation time is reduced by a factor almost equal to 30 (cf figure 3) in comparison with negotiations where no heuristics are applied and the agents negotiate in a random order, do not prioritise resources, and have initial welfare and preferences uniformly distributed in the interval $[0, 1]$. A precise description of the settings used for the corresponding experiments can be found in [13].

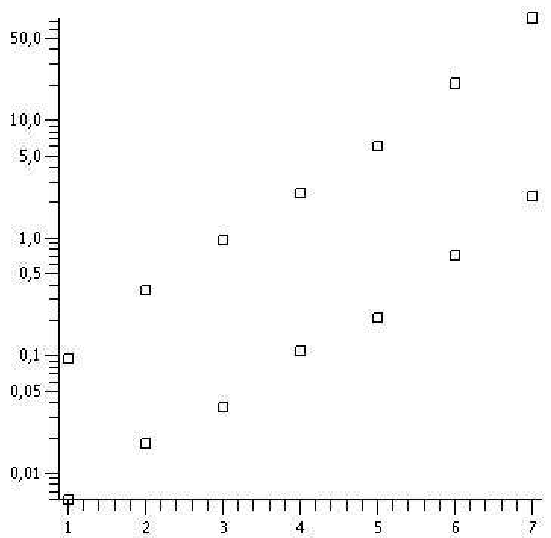

Figure 3: Negotiation time (in seconds) using the combined LW-MU heuristic (bottom) compared to a random strategy (up). The negotiation speed is multiplied by 30 when using this heuristic.

4 Protocol and Policy

The resource allocation problem can be solved distributedly by means of negotiation amongst the agents. The description of this process can be given by defining a public communication protocol, agreed by all agents, and private computational policies, held by the individual agents. The protocol defines what agents are allowed to say and how they should react (by means of their internal policy) to messages they receive. Giving a protocol is a necessary requirement for the definition of a suitable semantics of an agent communication language [16, 20]. In order to render the negotiation mechanism unambiguous, each policy needs to conform to the protocol.

We present here a public communication protocol derived from algorithms 1 and 2, and encapsulating our efficient methods for reasoning about agreements between groups of agents over resources allocations. The protocol is presented in the form of a deterministic finite state automaton (DFA) (see figure 4), in the flavour of [7]. The DFA consists of $n+1$ states: one state a_k per agent and a final state f. Each of the states a_k is characterised by the values of three variables: current lower bound L and upper bound U of the optimal egalitarian social welfare and the set $F_k(x)$ (for $x = (L+U)/2$) of agreements for the current group $E_k = \{a_1, ..., a_k\}$. The initial state is a_1 with variables assigned to the values $L_0, U_0, \emptyset$. The empty set means that the first agent does not need to take into account agreements reached by the other agents. The syntax [3] of our negotiation protocol is given by the language [18] $\mathcal{L}$ consisting of instances of the *tell* predicate which has the following five arguments: sender X, receiver Y, message M, lower bound L and upper bound U of the optimal egalitarian social welfare. A message M may take three forms, i) a non-empty set A of agreements, ii) failure due to the absence of possible consensus, iii) success when a consensus can be found, and iv) solution for publishing an egalitarian allocation A^*, solution of the problem. The language is then defined as

$$\mathcal{L} = \{\texttt{tell}(X, Y, M, L, U) | X \in S, Y \in S \texttt{ or } Y = S, (L, U) \in \mathrm{I\!R}^2, 0 \leq L \leq U\},$$

where $M \in \{\texttt{agreements}(A), \texttt{failure}, \texttt{success}, \texttt{solution}(A^*)\}$ and S stands for the socially ordered variant of the agent system $\{a_1, a_2, \ldots a_n\}$ such that $c_1 \leq c_2 \leq ... \leq c_n$.

The DFA's transition function maps pairs of states and elements of the input alphabet to states. In the context of communication protocols, elements of the input alphabet are dialogue moves and states are the possible stages of the interaction. The transition function consequently gives a clear semantics [3] to our protocol. We introduce the *next* function that transforms a_i into a_{i+1} for $i < n$ and a_n into a_1 so as to enable looping in the negotiations.

The transition function δ is then defined as the union of the following rules where i ranges from 1 to n:

- $\delta(a_i^{L,U,F_i}, \texttt{tell}(a_i, a_{next(i)}, \texttt{agreements}(F_i), L, U)) = a_{next(i)}^{L,U,F_{next(i)}}$
- $\delta(a_i^{L,U,\emptyset}, \texttt{tell}(a_i, a_1, \texttt{failure}, L, \frac{L+U}{2})) = a_1^{L,\frac{L+U}{2},F_1}$

- $\delta(a_n^{L,U,F_n}, \texttt{tell}(a_n, a_1, \texttt{success}, \frac{L+U}{2}, U)) = a_1^{\frac{L+U}{2}, U, F_1}$
- $\delta(a_n^{L,U,F_n}, \texttt{tell}(a_n, S, \texttt{solution}(A^*), \texttt{Round}(\frac{L+U}{2}, d), \texttt{Round}(\frac{L+U}{2}, d)) = f$

In each state a_k, agent a_k has to revise the set of agreements found by the prior agents $\{a_1, \ldots a_{k-1}\}$ and communicated by a_{k-1}. The graph loops back to the first agent either when no consensus can be found or when all the agents have found a consensus and in both cases the lower or upper bounds are updated accordingly to the dichotomous update, pessimistically in the first case and optimistically in the second one. The last agent a_n is responsible for detecting the final dichotomous step and does so by checking if $U - L < 10^{-d}$ holds. If it is the case, it chooses arbitrarily an egalitarian allocation and sends it to them. The negotiation stops and the agents can go and pick up their resources accordingly to the solution.

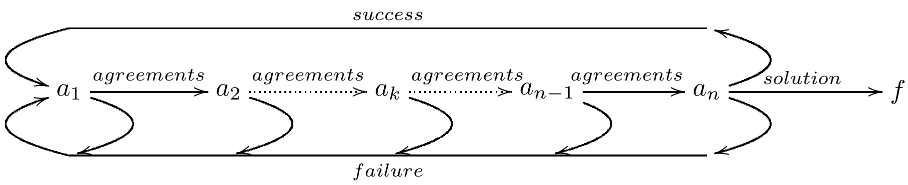

Figure 4: Public communication protocol.

A policy that conforms to the protocol and encapsulates the computational techniques is now given. Following [18], policies are expressed as dialogue constraints of the form $p_i \wedge C \Rightarrow p_{i+1}$, where p_i and p_{i+1} are dialogue moves. The dialogue constraints are constructed so as to associate unambiguously to each agent and message received a (unique) dialogue move [2]. Those policies give a pragmatics [3] that is easy to implement and execute in a distributed architecture.

- $\texttt{tell}(X, Y, \texttt{agreements}(A), L, U) \wedge (F_Y(A) = \emptyset) \Rightarrow \texttt{tell}(Y, a_1, \texttt{failure}, L, (L+U)/2)$
- $\texttt{tell}(X, Y, \texttt{agreements}(A), L, U) \wedge \neg((Y = a_n) \wedge ((U-L) < 10^{-d})) \wedge (F_Y(A) \neq \emptyset) \Rightarrow \texttt{tell}(Y, \texttt{next}(Y), \texttt{agreements}(F_Y(A)), L, U)$
- $\texttt{tell}(X, Y, \texttt{agreements}(A), L, U) \wedge (Y = a_n) \wedge ((U-L) \geq 10^{-d})) \wedge (F_Y(A) \neq \emptyset) \Rightarrow \texttt{tell}(Y, a_1, \texttt{success}, (L+U)/2, U)$
- $\texttt{tell}(X, Y, \texttt{agreements}(A), L, U) \wedge (Y = a_n) \wedge ((U-L) < 10^{-d}) \wedge (F_Y(A) \neq \emptyset) \Rightarrow \texttt{tell}(Y, S, \texttt{solution}(\texttt{OneOf}(F_Y(A))), \texttt{Round}((L+U)/2, d), \texttt{Round}((L+U)/2, d)$
- $\texttt{tell}(X, Y, \texttt{failure}, L, U) \wedge (F_Y(\emptyset) \neq \emptyset) \Rightarrow \texttt{tell}(Y, \texttt{next}(Y), \texttt{agreements}(F_1()), L, U)$
- $\texttt{tell}(X, Y, \texttt{failure}, L, U) \wedge (F_1(\emptyset) = \emptyset) \Rightarrow \texttt{tell}(Y, Y, \texttt{failure}, L, (L+U)/2)$

[2] All variables in the given dialogue constraints are implicitly universally quantified from the outside.

- $\texttt{tell}(X, Y, \texttt{solution}(A^*), sw_e^*, sw_e^*) \Rightarrow$ COLLECT RESOURCES

We assume that each agent is equipped with this policy. By definition, a dialogue move p is legal with respect to a state s if and only if there exists a state s' such that $\delta(s, p) = s'$. In order to make sure that the policy conforms to the protocol and is well formed, the reader can check that:

- for any (legal) message Msg received by an agent Y, the agent can compute a unique state (L, U, F) (determined by the protocol) and that state satisfies the constraints of a unique policy rule amongst those whose p_i match Msg (policy rules exhaustivity and independence). Consequently:
- i) agents never utter any illegal move (weak protocol conformance)
- ii) agents utter at least one legal output move for any legal input they receive (exhaustive protocol conformance)

5 Conclusion

We presented a sound method that guarantees agents to find an allocation 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, which can be kept secret. 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 opportunistically any superfluous agreements they might come up with. The solutions are efficient as far as they never over-consume resources. We proved empirically that the agents reason collectively much faster when thinking in priority about the most useful resources and could efficiently coordinate the sequence of their negotiations by using the monotonic increasing social order. Finally, the negotiation mechanism has been distributed over the agents engaged in the allocation process using a protocol and a policy conforming to it which implements the dichotomous search and encapsulates the efficient consensus search algorithm here-presented. The overall mechanism has been implemented on a JADE platfrom [5]. Part of our future work will be dedicated to a theoretical and experimental study of the frugal reduction's efficiency. We will also propose other ways of modelling an agent's preferences that will enable to solve the allocation problem in polynomial time and show how the mechanism can be used for negotiating the allocation of markets supervised by fair trade organisations. Finally, we would like to make the mechanism strategy-proof.

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] Kenneth J. Arrow. *Social Choice and Individual Values*. John Wiley and Sons, New York, 2 edition, 1963.

[2] Sylvain Bouveret, Michel Lemaître, Hélène Fargier, and Jérôme Lang. Allocation of indivisible goods: a general model and some complexity results. In Frank Dignum, Virginia Dignum, Sven Koenig, Sarit Kraus, Munindar P. Singh, and Michael Wooldridge, editors, *4rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), July 25-29, 2005, Utrecht, The Netherlands*, pages 1309–1310. ACM, 2005.

[3] Ronald J. Brachman and Hector J. Levesque. *Knowledge Representation and Reasoning*. Morgan Kaufmann Publishers, 2004.

[4] Marco Dall'Aglio and Fabio Maccheroni. Fair division without additivity. *AMM: The American Mathematical Monthly*, 112, 2005.

[5] Dionysis Dionysiou. *Egalitarian Resource Allocations in Multi-Agent Systems*. MSc Thesis, Imperial College London, 2006.

[6] Ulrich Endriss, Nicolas Maudet, Fariba Sadri, and Francesca Toni. Resource allocation in egalitarian agent societies, 2003.

[7] Ulrich Endriss, Nicolas Maudet, Fariba Sadri, and Francesca Toni. Logic-based agent communication protocols. 2004.

[8] Ulrich Endriss, Nicolas Maudet, Fariba Sadri, and Francesca Toni. Negotiating socially optimal allocations of resources. *Journal of Artificial Intelligence Research*, 25:315–348, 2006.

[9] Daniel Golovin. Max-min fair allocation of indivisible goods. *Technical Report CMU-CS-05-144, Carnegie Mellon University*, 2005.

[10] John C. Harsanyi. Can the maximin principle serve as a basis for morality? *American Political Science Review*, 86(2):269–357, 1996.

[11] Richard Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. In *CECOMM: ACM Conference on Electronic Commerce*, 2004.

[12] Hanan Luss. On equitable resource allocation problems: A lexicographic minimax approach. *Operations Research*, 47(3):361–378, 1999.

[13] Paul-Amaury Matt and Francesca Toni. Egalitarian allocations of indivisible resources: Theory and computation. In *Cooperative Information Agents (CIA'06)*. Lecture Notes in Computer Science, Springer Verlag, 2006.

[14] Hervé Moulin. *Axioms of Cooperative Decision Making*. Cambridge University Press, 1988.

[15] Arnold Neumaier. Complete search in continuous global optimization and constraint satisfaction. *Acta Numerica*, 13:271–369, 2004.

[16] Jeremy Pitt and Abe Mamdani. A protocol-based semantics for an agent communication language. In Dean Thomas, editor, *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99-Vol1)*, pages 486–491, S.F., July 31–August 6 1999. Morgan Kaufmann Publishers.

[17] John Rawls. *A Theory of Justice*. Harvard University Press, 1971.

[18] Fariba Sadri, Francesca Toni, and Paolo Torroni. An abductive logic programming architecture for negotiating agents. *Lecture Notes in Computer Science*, 2424, 2002.

[19] Amartya K. Sen. *Collective Choice and Social Welfare*. Holden Day, 1970.

[20] Munindar P. Singh. Agent communication languages: Rethinking the principles. *IEEE Computer*, 31(12):40–47, 1998.

[21] Gang Yu. On the max-min 0-1 knapsack problem with robust optimization applications. *Operations Research*, 44:407–415, 1996.

Paul-Amaury Matt, Francesca Toni and Dionysis Dionysiou
Department of Computing, Imperial College London,
South Kensington Campus, Huxley Building,
London SW7 2AZ, United Kingdom
Email: {`pmatt,ft,dd205`}`@doc.ic.ac.uk`