Competitive Voters vs. Collaborative Bidders: Agreements in Dynamic Task Assignment*

Marin Lujak[1] and Marija Slavkovik[2]

[1] CETINIA, Universidad Rey Juan Carlos, Madrid, Spain
[2] Dept of Computer Science, University of Liverpool, United Kingdom

Abstract. This paper is concerned with the problem of dynamic task-assignment among multiple mobile agents. We study and experimentally compare two approaches for achieving an agreement on how the tasks are to be assigned: negotiation through auctions and aggregation of preferences through voting. For the second approach, we formalize the task assignment problem as a voting problem and present the corresponding algorithm for mobile agents. Our experimentation is performed in a 2D-environment applying both agreement methods and comparing them in terms of their social welfare values.

1 Introduction

Multiple mobile-agent task and resource assignment is an important part of decision-making processes in many areas, some of which are: industrial procurement [4], manufacturing [8, 10, 18], network routing [7], airport traffic management [11], crisis management [15], logistics [16], and public transport [6]. In all of these fields, it is common for the system to be made up of heterogeneous agents with different goals and preferences. Reaching a globally good solution for the system in a multiple mobile-agent task assignment problem under such circumstances can be a complex task. Since the problem of resource or task allocation requires for each item to be allocated to one agent, and each agent to be allocated to only one task (or resource), an auction is a straightforward method of resolving this problem in a distributed way for self-interested agents. However, in human societies voting is also used for the purpose of fair distribution of desired items [2].

The alignment of individual and global utilities is a big issue in heterogeneous agents. Even when the agents are cooperative, the utilities of the individual may not necessarily be aligned with the utilities of the group. Since the assignment cost and utility function of a self-interested agent may or may not coincide with the global system's cost and utility, different notions of optimality may arise. Optimality can be seen from the global and individual point of view. Looking from the global standpoint, globally optimal solution searches for the minimum cost and maximum utility of the group as a whole and it does not consider the individual dynamics of assignment. However, in the globally optimal solution,

* AT2012, 15-16 October 2012, Dubrovnik, Croatia.

individual assignments might be far from the optimal so that there might exist some agents who are unsatisfied with their individual assignment. The question is can we start from constructing a global solution from individual interactions looking only to minimize the damage of the worst-off agents and how it behaves in respect to the global solution.

Two allocations can be compared, in terms of individual utilities[3] of participating agents, by comparing their *collective utility values*. Assuming that any given allocation yields some utility for each agent, a group utility function assigns a single utility, *i.e.*, social welfare, to the agent group. Group utility values, are natively studied in the theory of public choice and welfare economics [13, Chapter 23]. The use of public choice methods and theories has recently also gained the interest of computer scientists [5].

Examples of group utility functions include *utilitarian* and *egalitarian collective utility*. The utilitarian group utility function is defined as the sum of all the individual utilities. A high utilitarian social welfare is indicative of a "globally" good allocation, but it does not reflect on how good, or fair, an allocation is "locally". For example, if one agent's utility were increased while decreasing the utility of another agent for the same amount, the utilitarian social welfare would not be changed. In contrast, the egalitarian social welfare reflects the fairness aspect of an allocation. The egalitarian collective utility function is the lowest individual utility in the allocation. Thus a high egalitarian social welfare is indicative of a "locally" fair allocation. Another common social welfare is the *elitist collective utility* which coincides with the maximal individual utility in the allocation. A high elitist social welfare is indicative of a society that maximizes the "profit" of an individual disrespecting fairness and group optimality.

In this paper we are interested in a decentralized multi-agent system in which mobile agents move in an environment to reach their assigned tasks and communicate within a connected communication graph while doing so. We explore two cases of self-interested agents: collaborative and competitive one. For each of these cases, we apply a different method for assigning tasks and compare the resulting solutions in terms of their collective utility values. For cooperative agents we use a Bertsekas [3] modified dynamic auction algorithm with mobility [12], while for competitive agents we use a voting-based algorithm, in particular using the Borda count method [14]. We compare the assignment solutions of those two methods in a two-dimensional grid world, in terms of utilitarian, egalitarian and elitist social welfare. Collaborative agents compete for tasks (resources) by giving a bid with some value for the best-off task, while competitive agents try to make other agents go to less desirable targets. As a corollary of the properties of the modified auction algorithm, its solutions have superior average utilitarian collective utility values. We observe that the collaborative agents using the auction algorithm also get better solutions on average in terms of egalitarian, utilitarian, and elitist social welfare than the voting competing agents. However,

[3] In the case of task assignment we work with costs, which can be taken to be negative utilities

somewhat surprisingly the difference in social welfare between solutions is not too high.

This paper is organized as follows. In Section 2 we formulate the task assignment problem. Section 3 describes briefly the auction method used by the collaborative agents. The voting method for the competitive agents is described in Section 4. Section 5 contains simulation results comparing these negotiation approaches. We discuss our results, draw conclusions and outline the directions for future work in Section 6.

2 Target-assignment problem

We consider a set $A = \{1, \ldots, n\}$ of n collaborative mobile (vehicle) agents. The agents are represented as points in a plane positioned in an environment $E \subset \mathbb{R}^2$. The position of an agent $a \in A$ at time $t = 1, \ldots, T$ is given by $p_a(t) \in E$. We also consider a set $\Theta = \{1, \ldots, n\}$ of n targets (tasks), with $q_\theta \in E$ being the static position of task $\theta \in \Theta$.

Each agent, $a \in A$, is described by a tuple

$$a = \{p_a(t),\ \rho,\ v^{[a]}_{max},\ dist_a(t), \}\ , \tag{1}$$

where $\rho \in \mathbb{R}_{>0}$ is a fixed transmitting (communication) range of agent a's wireless transceiver for limited range communication, $v^{[a]}_{max}$ is its maximum velocity (maximum movement distance in each time period t), and $dist_a(t)$ is the total distance passed until period t. At any time t, each agent a knows its $p_a(t)$ and the position q_θ of each target $\theta \in \Theta$. Let $c_{a\theta}(t)$ be the (Euclidean) distance between the position of agent a and target θ. We consider the problem of dynamic assignment of the set A of vehicle agents to the set Θ of target locations, under the requirement that each agent is assigned to at most one target. The total traveled distance by all agents moving towards their targets, calculated as

$$\sum_i c_{a\theta}(t)\ ,\ t = 1, \ldots, T\ , \tag{2}$$

should be minimized. T is the upper time bound in which all the agents reach their distinct assigned target locations.

In each period t, each agent a is able to communicate to a set of agents $C_a(t) \subseteq A$ (belonging to the same connected component) reachable in a multi-hop fashion within the communication graph; at any time t, agents i and j communicate if

$$\|p_i(t) - p_j(t)\|_2 \leq \rho\ . \tag{3}$$

In this way, agents which are not within the communication range of each other can, however, exchange information over other interconnected agents. We assume that no global assignment information is available *a priori* and that the agents only receive information through their local interaction with the environment and with the connected agents in the communication graph.

3 Agreements via auction-based negotiation

The auction algorithm of Bertsekas [3] is a well-known algorithm for distributed task assignment. It gives a globally optimal solution in the case of a complete communication network. The negotiation among the agents is performed in two steps: bidding and assignment. In the bidding step each agent bids for the target which gives him the highest net value, starting from the momentary task prices and the individual task values. For each task, in the assignment phase, there must be one auctioneer agent that collects all the bids and assigns the task to the highest bidding agent. This process is performed in iterations. In each iteration, the prices of the tasks increase. The algorithm stops when there has been at least one bid for each task and all the tasks are, therefore, assigned.

A modification of the Bertsekas algorithm for dynamic environments and moving agents with incomplete local information is presented in [12]. This is the algorithm we apply for cooperative agents in our experiments. The assignment is performed in three phases: bidding, assignment and movement towards the assigned targets. The details of the algorithm can be found in [12]. In this paper, we compare the performance of this algorithm with the voting mechanism since both can work in the environment with an incomplete communication network. In the next section we present the voting-based approach.

4 Agreements via voting

Voting is a general group option-choosing method for societies of self-interested agents [5]. Many voting rules have been proposed in the literature, *e.g.*, plurality (majority) rule and refinements, ranked pairs, and Borda count. For a general overview of voting theory, see for example [14].

Formally, a voting problem is specified by a non-empty set of social options O and a set $A = \{a_1, \ldots, a_n\}$ of at least two agents. Each agent $a_i \in A$ reports his/her preferences over elements in O, which are represented by a complete, transitive preference relation[4] $\succsim_i^O$. A profile $P = \{\succsim_i^O \mid i \in A\}$ is the set of the preference orders of the agents A.

Let $\mathcal{R}(O)$ denote the class of all preference orderings over O. The set of all preference profiles is then given by $\mathcal{R}(O)^n$. An $n-person$ *voting rule* is function $F : \mathcal{R}(O)^n \rightarrow 2^O \setminus \{\emptyset\}$, that assigns to each tuple of n preference orders from $\mathcal{R}(O)$ a non-empty sub-set of options from O.

Which voting rule is used for a particular problem depends on the nature of the problem. The goal of plurality (majority) rule is to ensure that the elected option has the support of a majority. Ranked pairs [17], on the other hand, aim at electing the candidate who would win each head-to-head option comparison (Condorcet winner), while the Borda count rule chooses a consensus option, looking at how strongly the agent dislike certain options as well. The Borda rule

[4] A preference relation is transitive if and only if $o_1 \succsim_i o_2$ and $o_2 \succsim_i o_3$ implies $o_1 \succsim_i o_3$. The preference relation is complete if for every $o_1, o_2 \in O$ either $o_1 \succ_i o_2$, $o_2 \succ_i o_1$ or $o_1 \sim_i o_2$.

is one of the voting rules based on the principle of dissatisfaction minimization, namely the winner is chosen to be such that the dissatisfaction of the individual voters with the election solution is minimal.

One desirable feature of the Borda count rule is the low computational complexity of the determining the winner. According to the Borda rule, each option is assigned a *score* and the option with the highest score is elected a winner. The computational complexity of determining the Borda score, and with that the election winner, is very low compared to some other rules based on scores such as, *e.g.*, Dodgson and Kemeny [1, 9].

According to Borda count rule, each element $o \in O$ is given a score based on its position in the individual preference orders in P. The scores for the elements $o \in O$ are defined as

$$s(o) = \sum_{i \in N} \#\{(o') | o' \in O \text{ and } o \succsim_i^O o'\}.$$

The nnumber $\#\{(o')|o' \in O \text{ and } o \succsim_i^O o'\}$ is effectively the position of the option a in the agent i's preference order. For example, in the order $o_1 \succ_i o_2 \succ_i o_3$, the top ranked option o_1 is assigned a value 3 because it at least as good as 3 other options including itself. The Borda count rule returns the option with the highest score as a winner of the election.

It is directly observable that the Borda score for each option can be calculated in linear time of the size of the profile of preferences and consequently the respective linear order over the options can be generated in time $\mathcal{O}(m^2 \times n)$, where m is the number of options and n is the number of agents. Other low complexity methods like the plurality rule are not adequate for the voting problem in task assignment. Namely, in elections for task assignment, the number of options (tasks) equals the number of voters (agents). The plurality rule returns as winner the option that is most preferred by the largest number of agents. However, when the number of options is greater or equal to the number of agents, it can frequently happen that no option is supported by more than a few agents and in this case no winner can be found.

The task-assignment problem can be formulated as a voting problem in the following manner. Let $A = \{a_1, \dots, a_n\}$ be a set of n agents and $\Theta = \{\theta_1, \dots \theta_n\}$ a set of n target locations. Each agent $i \in A$ submits his (cardinal) preference order $\succsim_i^O$ over a set of social options $O = \{(i, \theta) \mid \theta \in \Theta\}$. The treference order $\succsim_i^O$ is a complete strict order over O corresponding to the values $u_i(j, \theta) | j \in A$, where the utility $u_i(j, \theta)$ expresses the utility that agent i has if task θ is assigned to agent $j \in A$. The utility $u_i(j, \theta)$ in a task-assignment is connected with the minimization of the assignment cost $c_{i\theta}$. The individual preferences $u_i(i, \theta)$ are ordered in ascending way such that $u_i(i, \theta) = \min c_{i\theta}$ is ordered first; *i.e.*, each agent prefers to be assigned to a task closer to him. For $i \neq j$ there are different ways in which $u_i(j, \theta)$ can be ordered, the simplest being $u_i(j, \theta) = max(c_{i\theta})$ for all j expressing that an agent prefers his most distant targets to be assigned to other agents.

We present the detailed description of our task-assignment algorithm based on the Borda count voting rule. Let us assume that the set of unassigned agents is $a_u \in U$, and a set of unassigned targets $\theta_u \in T$. Then each unassigned agent $a_u \in U$ does the following steps:

- it keeps in its memory a set of unassigned agents $a_u \in U$ and a set of unassigned targets $\theta_u \in T$.
- after constructing n preference orders as described above, it keeps its preference order $u_i(i, \theta)$ and sends $u_i(j, \theta)|j \neq i$ to other agents $a_j \in U, j \neq u$.
- it receives preference orders from other communicating unassigned agents $j \in U, j \neq i$ and constructs a preference profile also considering his own preferences. For aggregation, the Borda count voting rule is applied.
- In the cases when the Borda voting rule produces a tie, *i.e.*, the voting points are the same for multiple remaining targets, the agent i selects the winning option to be the one with the lowest utility value $u_i(i, \theta)$ and informs the other communicating agents of the same.
- Once a target $\theta \in T$ is assigned to agent $j \in U$, both are removed from unassigned agents U and targets Θ respectively. An election is now held for the next agent in U. The agents who have already gotten a task do not vote.

Example 1. Consider a set of five agents $A = \{a, b, c, d, e\}$ and a set of five targets $\Theta = \{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5\}$ distributed (at moment t) like in Figure 1.

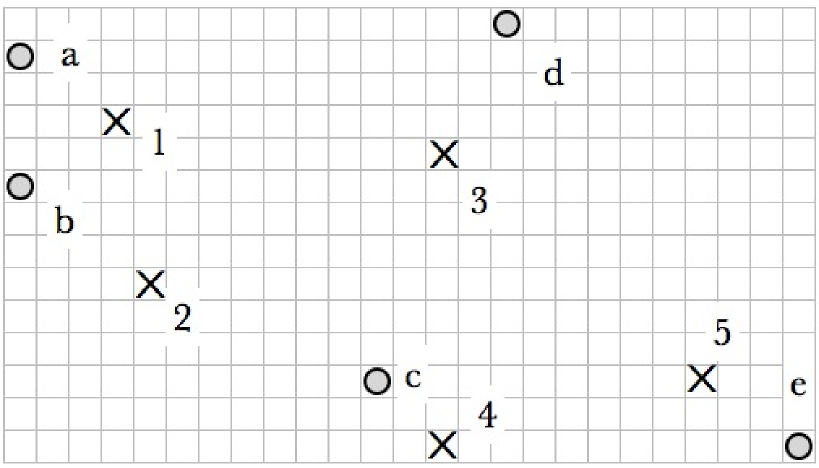

Fig. 1. An example distribution.

The Euclidean distances at t[5] are as follows:

$c_{a1} = 3.6$	$c_{a2} = 8.06$	$c_{a3} = 13.34$	$c_{a4} = 16.97$	$c_{a5} = 23.26$
$c_{b1} = 3.6$	$c_{b2} = 5$	$c_{b3} = 13.04$	$c_{b4} = 14.03$	$c_{b5} = 21.84$
$c_{c1} = 11.31$	$c_{c2} = 7.61$	$c_{c3} = 7.28$	$c_{c4} = 2.23$	$c_{c5} = 10$
$c_{d1} = 12.37$	$c_{d2} = 13.6$	$c_{d3} = 4.47$	$c_{d4} = 13.34$	$c_{d5} = 12.53$
$c_{e1} = 23.26$	$c_{e2} = 20.61$	$c_{e3} = 14.26$	$c_{e4} = 12$	$c_{e5} = 3.6$

Let us consider the following election sequence $N = (a, b, c, d, e)$.

Election 1 $A = \{a, b, c, d, e\}$, $O = \{(a, \theta_1), (a, \theta_2), (a, \theta_3), (a, \theta_4), (a, \theta_5)\}$. The profile of preferences regarding which target should be assigned to agent a is Profile 1 on Figure 2. The Borda scores are: $s(a, \theta_1) = 18$, $s(a, \theta_2) = 18$, $s(a, \theta_3) = 12$, $s(a, \theta_4) = 13$ and $s(a, \theta_5) = 14$. The winners are (a, θ_1) and (a, θ_2). Agent a is assigned to target a because $c_{a1} < c_{a2}$.

$Agent$	$Preferences$
a	$(a, \theta_1) \succ (a, \theta_2) \succ (a, \theta_3) \succ (a, \theta_4) \succ (a, \theta_5)$
b	$(a, \theta_5) \succ (a, \theta_4) \succ (a, \theta_3) \succ (a, \theta_2) \succ (a, \theta_1)$
c	$(a, \theta_1) \succ (a, \theta_5) \succ (a, \theta_2) \succ (a, \theta_3) \succ (a, \theta_4)$
d	$(a, \theta_2) \succ (a, \theta_4) \succ (a, \theta_5) \succ (a, \theta_1) \succ (a, \theta_3)$
e	$(a, \theta_1) \succ (a, \theta_2) \succ (a, \theta_3) \succ (a, \theta_4) \succ (a, \theta_5)$

Profile 1

$Agent$	$Preferences$
b	$(b, \theta_2) \succ (b, \theta_3) \succ (b, \theta_4) \succ (b, \theta_5)$
c	$(b, \theta_5) \succ (b, \theta_2) \succ (b, \theta_3) \succ (b, \theta_4)$
d	$(b, \theta_2) \succ (b, \theta_4) \succ (b, \theta_5) \succ (b, \theta_3)$
e	$(b, \theta_2) \succ (b, \theta_3) \succ (b, \theta_4) \succ (b, \theta_5)$

Profile 2

Fig. 2. Profiles from the first two elections.

Election 2 $A = \{b, c, d, e\}$, $O = \{(b, \theta_2), (b, \theta_3), (b, \theta_4), (b, \theta_5)\}$. The profile of preferences regarding which target should be assigned to agent b is Profile 2 on Figure 2. The Borda scores are: $s(b, \theta_2) = 16$, $s(b, \theta_3) = 9$, $s(b, \theta_4) = 8$ and $s(b, \theta_5) = 8$. The winner is (θ_2).

Election 3 $A = \{c, d, e\}$, $O = \{(c, \theta_3), (c, \theta_4), (c, \theta_5)\}$. The profile of preferences regarding which target should be assigned to agent c is Profile 3 on Figure 3. The Borda scores are: $s(c, \theta_3) = 6$, $s(c, \theta_4) = 8$ and $s(c, \theta_5) = 4$. The winner is (c, θ_4).

Election 4 $A = \{d, e\}$, $O = \{(d, \theta_4), (d, \theta_5)\}$. The profile of preferences regarding which target should be assigned to agent d is Profile 4 on Figure 3.

[5] we do not write the parameter t below to ease the readability

Agent	*Preferences*
c	$(c, \theta_4) \succ (c, \theta_3) \succ (c, \theta_5)$
d	$(c, \theta_4) \succ (c, \theta_5) \succ (c, \theta_3)$
e	$(c, \theta_3) \succ (c, \theta_4) \succ (c, \theta_5)$

Profile 3

Agent	*Preferences*
d	$(d, \theta_4) \succ (d, \theta_5)$
e	$(d, \theta_4) \succ (d, \theta_5)$

Profile 4

Fig. 3. Profiles from the last two elections.

The Borda scores are: $s(d, \theta_4) = 4$ and $s(d, \theta_5) = 2$. The winner is (d, θ_4), and the last assignment is (e, θ_5). The complete pairing of agents with targets is $\alpha = \{(a, \theta_1), (b, \theta_2), (c, \theta_4), (d, \theta_3), (e, \theta_5)\}$.

The collective utilitarian value of α can be calculated as $u(\alpha) = -(c_{a1} + c_{b2} + c_{c4} + c_{d3} + c_{e5}) = -18.9$. The collective egalitarian value of α can be calculated as $e(\alpha) = max(c_{a1}, c_{b2}, c_{c4}, c_{d3}, c_{e5}) = 5$. The collective elitist value of α can be calculated as $el(\alpha) = min(c_{a1}, c_{b2}, c_{c4}, c_{d3}, c_{e5}) = 2.23$.

As it can be observed directly in the example, the order in which the elections are held determines which target is assigned to which agent. If instead of the sequence $N = (a, b, c, d, e)$ we used the sequence $N = (b, a, c, d, e)$ the allocation reached would be $\alpha' = \{(b, 1), (a, 2), (c, 4), (d, 3), (e, 5)\}$ which is a worse allocation than α because $u(\alpha') = -21.96$ and $e(\alpha') = 8.06$.

Example 2. Consider a set of five agents $A = \{a, b, c, d, e\}$ and a set of five targets $\Theta = \{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5\}$ distributed (at moment t) like in Figure 2.

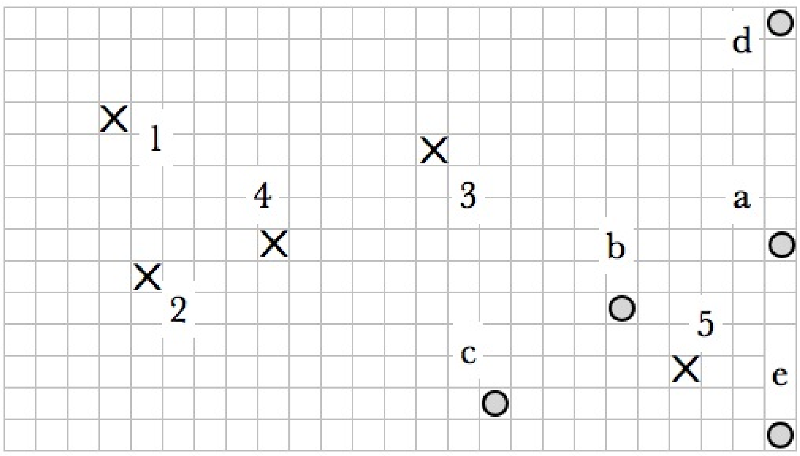

Fig. 4. An example distribution.

The Euclidean distances at t are as follows:

$c_{a1} = 21.38$	$c_{a2} = 20.02$	$c_{a3} = 11.4$	$c_{a4} = 16$	$c_{a5} = 5$
$c_{b1} = 17.08$	$c_{b2} = 15.03$	$c_{b3} = 7.81$	$c_{b4} = 11.18$	$c_{b5} = 2.83$
$c_{c1} = 14.31$	$c_{c2} = 11.7$	$c_{c3} = 8.24$	$c_{c4} = 8.6$	$c_{c5} = 6.08$
$c_{d1} = 21.21$	$c_{d2} = 22.45$	$c_{d3} = 11.7$	$c_{d4} = 17.46$	$c_{d5} = 11.04$
$c_{e1} = 23.26$	$c_{e2} = 7.07$	$c_{e3} = 14.21$	$c_{e4} = 17.09$	$c_{e5} = 3.16$

Let us consider the following election sequence $N = (a, b, c, d, e)$. We reach the following assignment $\alpha = \{(a, \theta_1), (b, \theta_4), (c, \theta_3), (d, \theta_5), (e, \theta_2)\}$. The utilitarian welfare is $u(\alpha) = -58.91$, the egalitarian welfare is $e(\alpha) = 21.38$, while the elitist welfare is $el(\alpha) = 7.07$.

5 Simulation setup and results

We simulate a multi-agent system with mobile agents applying the Bertsekas modified iterative auction algorithm and the voting-based algorithm in MatLab. The dynamic modified auction algorithm as well as the presented voting algorithm were experimented with complete assignment information exchange, *i.e.*, the communication graph among the agents is connected and every two agents in the group can communicate, if not directly, then at least in a multi-hop fashion through other connected agents.

Without loss of generality and for simplicity, we model the agents as points in a plane, which move on a straight line towards their assigned targets. The experiments were performed for up to 10 agents in environment $[0, 50]^2 \subset \mathbb{R}^2$ where the initial agent and target positions were generated uniformly randomly.

We considered 30 different instances for the problem with 10 agents and 10 targets. We measured the total distance traveled $\sum_{t=1}^{T} dist_{ai}(t)$ by agents $a_i \in A$ and calculated utilitarian, egalitarian, and elitist social welfare as a negative value of the passed distances of individual agents and the agent group as a whole. We chose this approach since every agent is better off if its assigned target is closer to it. Moreover, the cost of assignment is proportional to the traveled distance. Therefore, the elitist welfare is measured as the utility of the agent that is currently best off as negative distance cost $-\min_i \sum_{t=1}^{T} dist_{ai}(t)$. The utilitarian social welfare is the sum of individual utilities $-\sum_{i=1}^{n} \sum_{t=1}^{T} dist_{ai}(t)$, while the egalitarian social welfare is given by the utility of the agent that is currently worst off $-\max_i \sum_{t=1}^{T} dist_{ai}(t)$.

From Figures 5 and 6, it can be seen that the egalitarian and the utilitarian welfare in the case of the iterative auction algorithm are better off in all the experimented instances in respect to the voting algorithm while the elitist welfare (Figure 7) is better in 25, equal in 4 and worse in 1 out of 30 instances. Performance of the iterative auction in respect to the voting method is evaluated measuring the gap (in percentage) $g = [(SW_{VM} - SW_{IA})/SW_{IA}] * 100$, between (elitist, egalitarian and utilitarian) social welfare obtained by voting mechanism SW_{VM} and iterative auction SW_{IA} respectively. The gap provides an estimation of the relative extra-cost incurred in the competitive scenario (voting mechanism) in respect to the collaborative one (iterative auction) for not having at disposal all the system information to optimally assign the available targets.

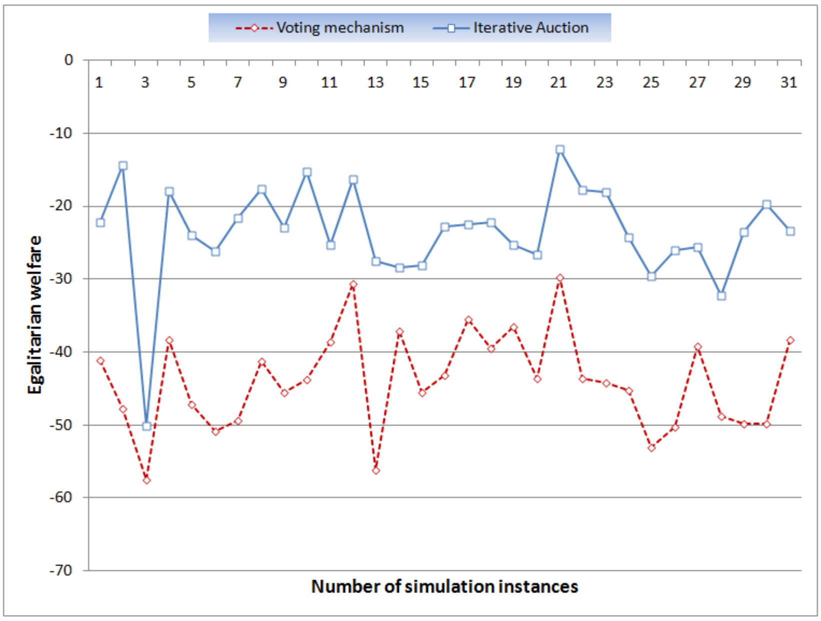

Fig. 5. Egalitarian welfare.

From Figure 8 and Table 1, it can be seen that the gap between those two scenarios for the egalitarian and utilitarian welfare is significantly lower and more uniformly distributed than the elitist welfare gap which in average is almost 3 times higher than the other two.

	Utilitarian welfare	Elitist welfare	Egalitarian welfare
(Auction-Voting)/Auction[%]	121,8	281,2	99,5

Table 1. Average gap values between voting mechanism and iterative auction for utilitarian, elitist, and egalitarian welfare [%]

The reason for the lower performance of the voting mechanism lies in the fact that although the communication graph among agents is connected, the voting mechanism obscures the information necessary for a good global solution. This can be explained by the strong influence of greedy policy in ordering preferences sent from one agent to others to direct them to less preferred targets, thus

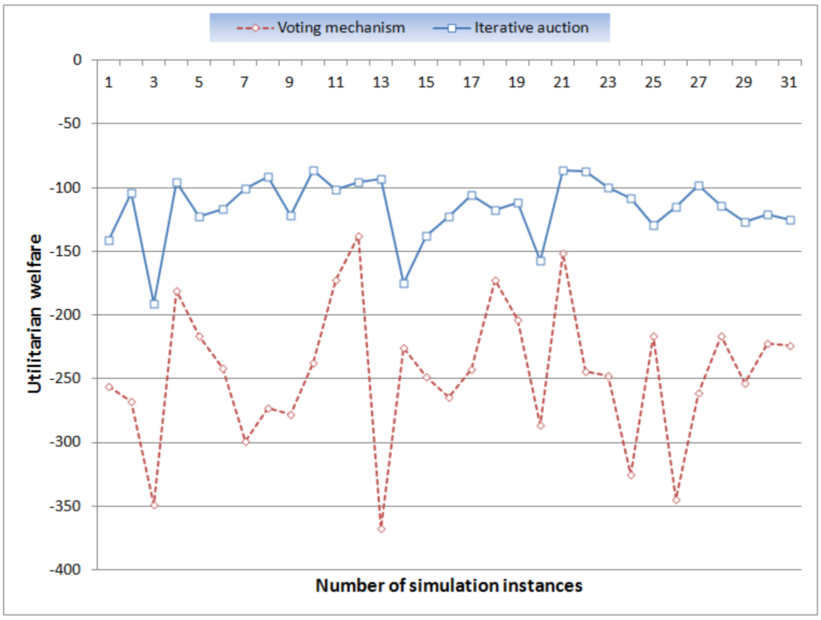

Fig. 6. Utilitarian welfare.

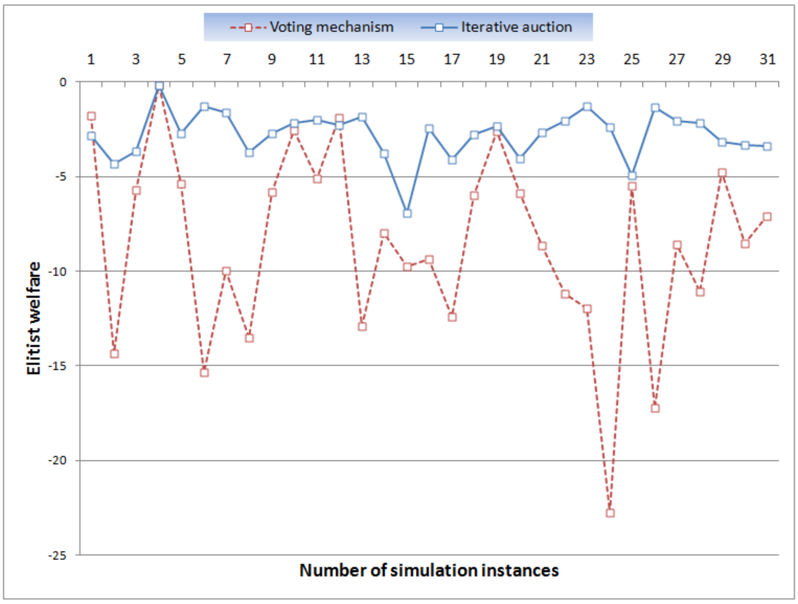

Fig. 7. Elitist welfare.

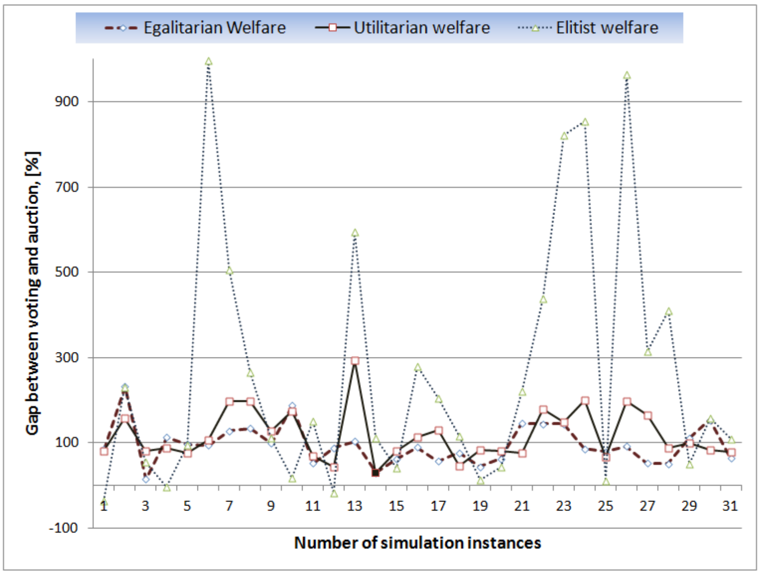

Fig. 8. Difference in egalitarian, utilitarian, and elitist welfare between negotiation through iterative auction and voting mechanism for instances with 10 agents and 10 targets.

minimizing individual cost and, on average, worsening the global and other individual solutions. The individual assignment process of the voting mechanism doesn't take into consideration individual agent bias and is performed taking into consideration all the information at disposal with the same weight. Furthermore, since a random factor is introduced through lexicographic ordering of equivalently valued targets, the distribution of the quality of the solution on the best-case and worst-case agent cannot be achieved.

6 Conclusions and future work

In this work we compared two approaches for solving the multiagent task allocation problem: negotiation through auctions and preference aggregation via a voting rule. The voting-based method is applicable both to competitive and to collaborative environments, though it is structured such that the assignment information exchanged is minimal and the communicated preferences are only ordinal. The auction-based method is a broadly used method for task assignment of collaborative agents that results in an optimal allocation solution. Our results show that cooperative agents that use the auction method to agree on how the tasks should be assigned on average reach solutions that are cost optimal for the system as a whole and sufficiently good for the agents individually. Moreover, the worst-off and best-off agent in the system have a significantly better local assignment solution in this case than when the voting approach is used. When we evaluated the deviation of the voting mechanism in respect to Bertsekas auction algorithm, the elitist welfare resulted in the lower, while the egalitarian welfare in the higher difference in respect to the (optimal) auction solution.

We formulated the problem as a dynamic task assignment problem to allow for the case when not all agents are connected in the communication graph and voting is done only among communicating agents. The voting algorithm presented in this paper assumes a connected communication graph which allows for a dynamic update of assignment information within all of the agents present in the system. We intend to consider the disconnected communication graph case in our future work.

In our experiments with the voting method, we modelled the suggested utility values of each agent for the assignment of other agents assuming a negative influence of similar assignments one to another. However, we can foresee scenarios in which the relations between the agents in the grid are more complex, *e.g.*, when the agents cooperate with some group of other agents but compete with others. It would be interesting to compare the results from auction-based methods and those from voting in such scenarios, as well.

A great disadvantage of our voting approach stems from its sequentially, namely the tasks are allocated to the agents following a sequence of one agent at a time. The solution directly depends on the sequence in which the agents are considered. In our experiments we use a lexicographic sequence, however there might exist a heuristic that helps us generate the sequences in such a manner that the desired social welfare of the assignment solution is increased. We intend to explore such heuristics in our future work.

7 Acknowledgements

The work of Marin Lujak was supported in part by the Spanish Ministry of Science and Innovation through the projects "AT" (Grant CONSOLIDER CSD2007-0022, INGENIO 2010) and "OVAMAH" (Grant TIN2009-13839-C03-02) co-funded by Plan E. The work of Marija Slavkovik was partially supported both by the Virtual Engineering Centre (funded by the NWDA and the ERDF) and EPSRC (funding via research project EP/F033567).

Bibliography

[1] J. Bartholdi, C. A. Tovey, and M. A. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6:157–165, 1989.

[2] Y. Barzel and T. Sass. The allocation of resources by voting. *Quarterly Journal of Economics*, 105(3):745–771, 1990.

[3] D.P. Bertsekas. The auction algorithm: A distributed relaxation method for the assignment problem. *Annals of Operations Research*, 14(1):105–123, 1988.

[4] M. Bichler, A. Davenport, G. Hohner, and J. Kalagnanam. Industrial procurement auctions. In P. Cramton, Y. Shoham, and R. Steinberg, editors, *Combinatorial Auctions*, pages 593–612. MIT Press, 2006.

[5] F. Brandt, V. Conitzer, and U. Endriss. Computational social choice. In G. Weiss, editor, *Multiagent Systems*. MIT Press, 2012 Forthcoming.

[6] E. Cantillon and M. Pesendorfer. Auctioning bus routes: The London experiance. In P. Cramton, Y. Shoham, and R. Steinberg, editors, *Combinatorial Auctions*, pages 573–592. MIT Press, 2006.

[7] R. Feldmann, M. Gairing, Thomas Lücking, Burkhard Monien, and Manuel Rode. Selfish routing in non-cooperative networks: A survey. In Branislav Rovan and Peter Vojtáš, editors, *Mathematical Foundations of Computer Science 2003*, volume 2747 of *Lecture Notes in Computer Science*, pages 21–45. Springer Berlin / Heidelberg, 2003.

[8] S. Giordani, M. Lujak, and F. Martinelli. A decentralized scheduling policy for a dynamically reconfigurable production system. In *Proceedings of the 4th International Conference on Industrial Applications of Holonic and Multi-Agent Systems: Holonic and Multi-Agent Systems for Manufacturing*, HoloMAS '09, pages 102–113, Berlin, Heidelberg, 2009. Springer-Verlag.

[9] E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Exact analysis of Dodgson elections: Lewis carroll's 1876 voting system is complete for parallel access to NP. *Journal of the Association for Computing Machinery*, 44(6):806–825, 1997.

[10] P. Y. Huang and C.S. Chen. Flexible manufacturing systems: An overview and bibliography. *Production and Inventory Management*, 27(3):80–90, 1986.

[11] G. Jonker, J.J. Meyer, and F. Dignum. Towards a market mechanism for airport traffic control. In *Proceedings of the 12th Portuguese conference on Progress in Artificial Intelligence*, EPIA'05, pages 500–511, Berlin, Heidelberg, 2005. Springer-Verlag.

[12] M. Lujak and S. Giordani. Value of incomplete information in mobile target allocation. In *MATES'11: Proc. of the 9th German conf. on Multiagent system technologies*, volume 5774 of *Lecture Notes in Computer Science*, pages 89–100. Springer-Verlag, 2011.

[13] D. C. Mueller. *Public choice III.* Cambridge University Press, Cambridge [England] New York, 2003.

[14] H. Nurmi. Voting theory. In D. Rios Insua and S. French, editors, *e-Democracy*, volume 5 of *Advances in Group Decision and Negotiation*, pages 101–123. Springer Netherlands, 2010.

[15] B.Silvia Suárez, Christian Quintero M., and Josep de la Rosa. A real time approach for task allocation in a disaster scenario. In Yves Demazeau, Frank Dignum, Juan Corchado, and Javier Pérez, editors, *Advances in Practical Applications of Agents and Multiagent Systems*, volume 70 of *Advances in Intelligent and Soft Computing*, pages 157–162. Springer Berlin / Heidelberg, 2010.

[16] P. J. 't Hoen and J. A. La Poutre. A decommitment strategy in a competitive multi-agent transportation setting. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, AAMAS '03, pages 1010–1011, New York, NY, USA, 2003. ACM.

[17] T. N. Tideman. Independence of clones as a criterion for voting rules. *Social Choice and Welfare*, 4:185–206, 1987.

[18] H. Van Dyke Parunak. Applications of distributed artificial intelligence in industry. In G. M. P. O'Hare and N. R. Jennings, editors, *Foundations of distributed artificial intelligence*, chapter Applications of distributed artificial intelligence in industry, pages 139–164. John Wiley & Sons, Inc., New York, NY, USA, 1996.