Robot Exploration with Combinatorial Auctions

M. Berhault(1) H. Huang(2) P. Keskinocak(2) S. Koenig(1) W. Elmaghraby(2) P. Griffin(2) A. Kleywegt(2)

(1)College of Computing
{marc.berhault,skoenig}@cc.gatech.edu
Georgia Institute of Technology
801 Atlantic Drive, Atlanta GA

(2)School of Industrial and Systems Engineering
{hhuang,pinar,welmagh,pgriffin,anton}@isye.gatech.edu
Georgia Institute of Technology
765 Ferst Drive, Atlanta GA

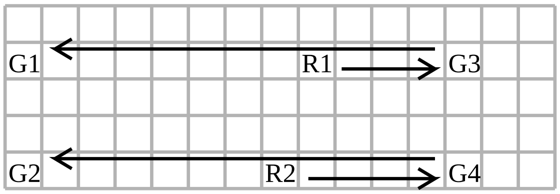
***Abstract*— We study how to coordinate a team of mobile robots to visit a number of given targets in partially unknown terrain. Robotics researchers have studied single-item auctions to perform this exploration task (where robots bid on single targets) but these do not take synergies between the targets into account. We therefore design combinatorial auctions (where robots bid on bundles of targets), propose different combinatorial bidding strategies and compare their performance with each other, as well as to single-item auctions and an optimal centralized mechanism. Our computational results in TeamBots, a multi-robot simulator, indicate that combinatorial auctions generally lead to significantly superior team performance compared to single-item auctions, and generate very good results compared to an optimal centralized mechanism.**

I. Introduction

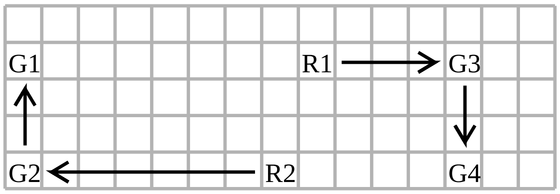
This joint work between robotics researchers and industrial engineers studies an exploration task where a team of mobile robots needs to visit a number of predetermined targets in a partially unknown terrain. Examples of situations where such exploration tasks occur include: 1) environmental or hazardous clean-up missions where the extent of damage is unknown, 2) space-exploration missions, and 3) search and rescue missions. An important characteristic of the exploration tasks is that the assignment of targets to robots can turn out to be suboptimal as the robots gain more information about the terrain, for example, when a robot suddenly discovers that it is separated by a wall from one of the targets assigned to it. We make the simplifying assumption that there is information symmetry among the robots, that is, when they visit targets or gain more information about the terrain during execution, they share this information with the other robots. How to assign and re-assign targets to robots for the exploration task is a difficult problem. Centralized control requires that the controller is always operational and in communication range. It is also inefficient in terms of both the required amount of computation and communication since the central controller is the bottleneck of the system. Decentralized control does not have these disadvantages but can result in suboptimal team performance. Researchers in artificial intelligence and robotics have recently investigated market-based approaches, in particular, auctions as a means for decentralized control. Auctions are efficient in terms of both the required amount of computation and communication since information is compressed into numeric bids that the robots can compute in parallel [1].

Auctions can be used to obtain feasible solutions for the exploration task as follows: Every robot bids on targets and then has to visit all targets that it wins. As the robots gain more information about the terrain during execution, the assignment of targets to robots can be changed by auctioning off the targets assigned to the robots. So far, researchers in robotics have studied single-item auctions in the context of the exploration tasks, where the targets are auctioned off one at a time. However, single-item auctions can result in highly suboptimal allocations if there are strong synergies between the items for the bidders. Two items are said to exhibit positive (negative) synergy for a bidder if their combined value for the bidder is larger (smaller) than the sum of their individual values. There are indeed strong synergies between the targets for the exploration task. To understand why this is so, consider the example gridworld from Figure 1(a) with two robots (R1 and R2) and four targets (G1, G2, G3, and G4). There is a strong positive synergy between targets G3 and G4 for robot R1 because the robot can reach the second target with a short travel distance after it has reached the first one. More generally, there is a strong positive synergy between nearby targets. On the other hand, there is a strong negative synergy between targets G1 and G3 for robot R1 because they are on opposite sides of the robot and the robot can therefore reach the second target only with a long travel distance after it has reached the first one.

Combinatorial auctions attempt to remedy the disadvantages of single-item auctions by allowing bidders to bid on bundles of items. If a bidder wins a bundle, he wins all the items in that bundle and hence is able to incorporate his synergies into his bids. For example, if the targets are auctioned off in a single-item auction, then robots R1 and R2 first move to targets G3 and G4, respectively, and then



(a) Trajectories with Single-Item Auctions



(b) Trajectories with Combinatorial Auctions

Fig. 1. Motivating Example

they move to targets G1 and G2, respectively, for a total travel distance of 33 units. In contrast, if the targets are auctioned off in a combinatorial auction, then robot R2 wins bundle {G1, G2} and thus first moves to target G2 and then to target G1, and robot R1 wins bundle {G3, G4} and thus first moves to target G3 and then to target G4, for a total travel distance of 17.

The focus of this paper is on the design of combinatorial auctions for the exploration task and on the development of bidding strategies for the robots so that they achieve a high level of team performance. We propose different bidding strategies and compare the resulting team performance with that of the other bidding strategies, as well as that of single-item auctions and an optimal centralized mechanism. Our computational results in TeamBots [2], a multi-robot simulator, indicate that combinatorial auctions generally lead to significantly superior team performance compared to single-item auctions, and generate very good results compared to an optimal centralized mechanism. We also provide some interesting insights on the performance of alternative bidding strategies, and show that some of them are significantly more efficient than others with respect to multiple criteria such as the travel distance, the travel time, the robot utilization and the amount of communication.

II. Related Work

Multi-agent navigation and coordination is a fairly new area of research, and a wide variety of techniques are being looked into in order to coordinate teams of robots. An overview can be found in [3]. Although artificial intelligence has studied how to coordinate agents with market-based mechanisms for quite a while [4], robotics has only recently started to investigate how to use them to coordinate teams of robots, and only a limited amount of research exist on the topic. Auction-based systems for dynamic task allocation to robots have been applied to tasks like box pushing [5] and robosoccer [6] but we know of only two applications to exploration tasks, none of which use combinatorial auctions. Simmons et. al. designed an auction-based system for multi-robot exploration and mapping using single-item first price auctions [7]. The terrain is unknown at the beginning, and is divided into fine cells. The auctioneer integrates the sensor readings of the robot into a map and auctions off selected frontier cells (known cells bordering unknown cells) as targets for the robot. The bids of the robots for a target depend on both their distance to it and the information that they can gain upon reaching the cell, that is, how many unknown cells become known because they are within sensor range. The auctioneer repeatedly assigns the target with the highest bid to the winning robot, with the restriction that each robot is assigned no more than one target and each target is assigned to no more than one robot. Zlot et. al. designed a similar but decentralized system and studied the effects of various target selection techniques [8] and opportunistic optimization techniques with leaders [9] on the team performance. Our research extends this research by demonstrating that combinatorial auctions result in better team performance than the single-item auctions used so far for exploration tasks.

The body of literature on combinatorial auctions has been growing mainly in two directions [10]: auction design and winner determination (computing the optimal allocation of the items to the bidders). The winner determination problem has been shown to be NP-complete, that is, no algorithm is guaranteed to find the optimal solution in polynomial time [11]. Several researchers have designed search algorithms that find optimal solutions [12] [13] [14]. While finding optimal solutions would be ideal, near-optimal solutions that can be obtained quickly are acceptable for many applications, and hence approximate solution methods have been developed [14] [15]. An area that is largely unexplored in the combinatorial auction literature is bidding strategies. Submitting bids for bundles poses a challenging problem for bidders since the number of possible bundles is exponential in the number of items and it is therefore prohibitively time consuming to submit a bid for every bundle. Developing bundles and determining the appropriate bids remain open questions. By proposing and testing bidding strategies for our specific application, we contribute to the combinatorial auction literature in this direction.

III. Approach

We propose a combinatorial auction mechanism and bidding strategies for the robots to solve the exploration task. The degrees of freedom of the auction designer depend on the application areas. For example, the auctioneer of a transportation service procurement auction can determine the participants and the format of the auction but not the objectives or the valuations of the auction participants [16]. We, in contrast, have complete control over all of these factors. In the following, we first discuss some of the criteria used to measure team performance followed by our combinatorial auction mechanism and possible bidding strategies.

A. Market Design

We need to determine the various factors that define a market, such as the agents, the items, the auction format, and when and how transactions will occur. For the exploration task, robots are a natural choice for the agents, and targets are a natural choice for the items. The auctioneer is a virtual agent who has sole responsibility for holding auctions and determining their winners but has no other knowledge and cannot control the robots. Initially, no robot owns any targets. Whenever a robot visits a target or gains more information about the terrain, the auctioneer starts a new auction that contains all targets that have not yet been visited. Each robot, including the current owner of a target, then generates bids in light of the new information. The auctioneer closes the auction after a predetermined amount of time, determines the winning bids, and notifies the robots with winning bids. The robots with winning bids now own the corresponding targets and have the responsibility to visit them, whereas the robots that owned them previously are relieved from that responsibility. Notice that one could hold auctions less frequently or with fewer targets, but this would decrease the responsiveness of the robots to new information about the terrain.

Our choice of the auction format determines several factors such as the richness of information to be communicated, the complexity of bid preparation, the complexity of choosing the final allocation, and the quality of the final outcome. We have already argued in favor of combinatorial auctions but still need to decide whether the auctions should consist of a single round or multiple rounds. Multi-round combinatorial auctions have some advantages over single-round combinatorial auctions. For example, 1) they save bidders from specifying their bids for a large number of bundles in advance, 2) they can be adapted to dynamic environments where bidders and items arrive and depart at different times, 3) bidders do not have to reveal all of their preferences to other bidders. However, the auctioneer has to determine winners in every round and communicate some information about the current bids to the bidders, which increases the amount of computation and communication, respectively. Therefore, we chose to use sealed-bid single-round combinatorial auctions. Each robot wins at most one bundle per auction because there can be negative synergies between items and thus also between bundles, and a robot might not want to win two bundles with negative synergies. Unfortunately, it is still NP-complete for the auctioneer to determine the winning bids, which are those that maximize the revenue of the auctioneer. Since the total time needed to solve the exploration task is a key measure of the team performance, we use an approximate solution method based on a primal-dual algorithm by Zurel and Nisan [17].

B. Bidding Strategies

We assume that the robot that visits target j for the first time receives reward R_j dollars. A robot has two types of costs: it has to pay the amount of its bids for the bundles that it wins and it has to pay one dollar for each unit of distance it travels. Thus, a robot needs to be able to determine distances quickly, for example, from its current location to a target or from one target to the next, even though it has only incomplete information about the terrain. Our robots use an optimistic distance estimate, namely the shortest distance under the assumption that unknown terrain is easily traversable. They compute these distances with D* Lite [18], an incremental version of A* [19], since their information about the terrain changes only slowly.

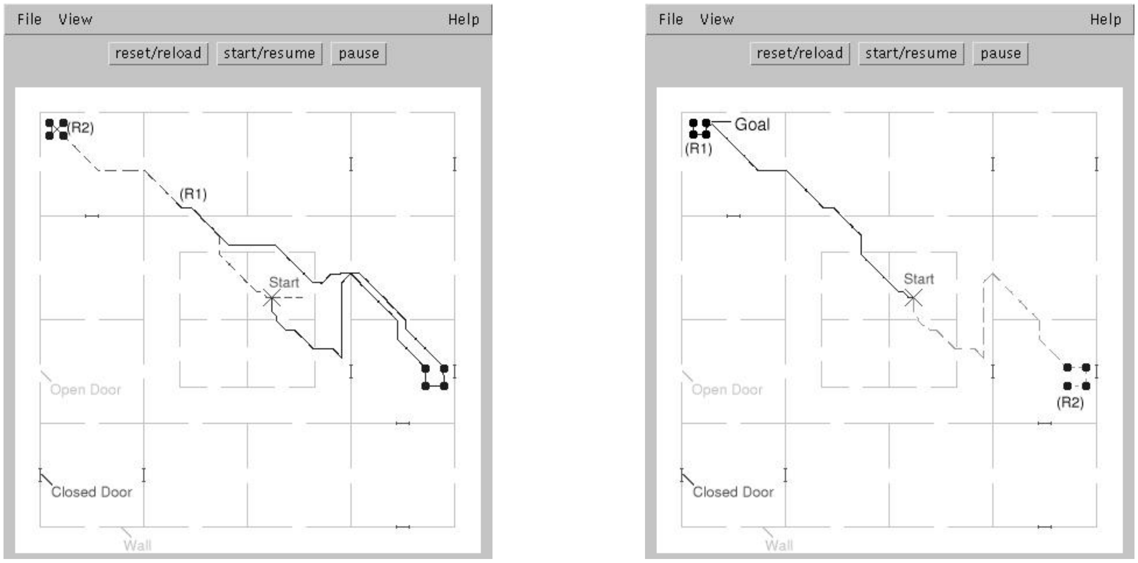
For single-item auctions, robotics researchers have used the following simple bidding strategy:

- ***Single*** : Each robot bids its surplus for a target, that is, the reward that it receives for the target minus the optimistic travel cost for visiting the target from its current location.

We use this strategy as a benchmark and generalize it to bundles of targets, where each robot continues to bid its surplus for a bundle, that is, the rewards that it receives minus the optimistic travel cost for visiting all targets contained in the bundle from its current location. We estimate the travel cost with a nearest neighbor heuristic, that is, the distance that results when one repeatedly moves to the closest unvisited target contained in the bundle.

The remaining issue is how the robots compute their bids, i.e., on which bundles they bid and how they set the bid prices for these bundles. We do not limit the number of bids that can be submitted by a robot. However, since the auction closes after a predetermined amount of time, the robots need to compute their bids and communicate this information to the auctioneer within the time limit, which effectively limits the number of bids they can submit. We therefore explored the following bidding strategies:

- ***3-max*** : Bid on all bundles with n or fewer targets. Notice that the number of bundles increases exponentially in n and this strategy quickly becomes infeasible for large n. We therefore use $n = 3$ in our experiments.
- ***Combination*** : Bid on all bundles of one or two items. Additionally, for all other bundles of six or less items,



(a) Robot Trajectories using ***Single***

(b) Robot Trajectories using ***Graph-Cut***

Fig. 2. Robot Trajectories (Screenshots of TeamBots)

bid on the top $6k$ bundles in each bundle size where k is the total number of bundles.

- ***Greedy*** : Define good sequences of targets recursively as follows: Each single target is a good sequence. A good sequence that ends in target s yields another good sequence when it is appended with target t if t is the closest target to s among all targets not contained in the old sequence and the surplus of the new sequence is no smaller than the surplus of the old sequence. Each sequence then corresponds to a bundle.
- ***Graph-Cut*** : Generate a complete undirected graph whose vertices correspond to the targets. The cost of an edge between two targets corresponds to the optimistic travel distance between them. Then use the maximum cut algorithm to split the graph into two parts by removing edges so that the sum of the costs of the removed edges is maximized. The targets in each of the two subgraphs form one bundle each. Then invoke the algorithm recursively for each of the two subgraphs to generate further bundles, until the subgraph contains only one target. Since it is NP-complete to compute the maximum cut, we used the "Computation Optimization Laboratory: Graph-Partition and Box-Constrained Quadratic Optimization" by Steven Benson, Yinyu Ye and Xiong Zhang to compute an approximation [20].

IV. Experiments

A. Setup

We tested our combinatorial auction mechanism with a team of three robots that navigate in a virtual building composed of rooms connected by doors that are closed with probability 0.2. The layout of the building that we used is shown in Figure 2(a), where the gray lines are walls, some of which contain small black lines which represent the closed doors. One of the most important factors that affect the performance of the team of robots is the (relative) locations of the targets. We therefore conducted experiments with different ways of distributing the targets in the building, resulting in four different environments:

- ***uniform*** : Eight targets are distributed randomly (that is, with uniform probability) in the building.
- ***c_1*** : One cluster of four targets each is placed randomly in the building (that is, the four targets are located near each other).
- ***c_2*** : Two clusters of four targets each are distributed randomly in the building.
- ***c_3*** : Three clusters of four targets each are distributed randomly in the building.
- ***c_4*** : Four clusters of four targets each are distributed randomly in the building.
- ***c_5*** : Five clusters of four targets each are distributed randomly in the building.

The simulations were run in Teambots [2], a multi-robot simulator. We ran ten different experiments for each environment, varying the locations of the targets or clusters in the building. We ran the simulations twice for different prior knowledge of the robots:

- ***Partially Known Environment:*** In this case, each robot knew the locations of the walls, doors and targets to be visited at the beginning of an experiment, but did not know which doors were closed. The state

TABLE I

OVERVIEW

	Partially Known Environment						***Completely Known Environment***					
Type	***uniform***	***c_1***	***c_2***	***c_3***	***c_4***	***c_5***	***uniform***	***c_1***	***c_2***	***c_3***	***c_4***	***c_5***
single												
number of bids	102.3	28.5	112.8	243.3	412.2	632.1	101.7	29.1	111.0	238.8	392.1	635.1
travel distance	296.9	123.3	206.4	333.0	511.9	434.6	252.4	120.1	197.4	294.9	465.9	426.5
time spent	1554.1	495.0	807.9	1303.6	1979.0	1716.1	1237.8	481.0	771.9	1196.1	1798.1	1690.0
robot utilization	71.0	92.0	93.8	93.6	95.3	93.4	75.8	92.1	93.8	91.8	95.5	92.6
3-max												
number of bids	692.4	70.8	709.2	3090.0	9418.5	20488.5	692.4	72.9	709.2	3079.2	9333.6	20506.5
travel distance	298.8	91.4	149.1	160.5	245.1	263.2	230.4	82.1	123.0	142.4	224.6	247.9
time spent	1736.9	533.5	698.1	728.1	1086.4	1151.8	1166.8	478.8	558.4	625.9	1031.4	1055.7
robot utilization	65.5	63.3	80.0	83.1	84.6	85.1	74.4	63.4	81.9	84.9	82.5	87.5
combination												
number of bids	739.8	66.0	743.1	2392.8	5116.5	8620.2	745.5	68.1	739.2	2366.4	5086.8	8731.5
travel distance	303.8	91.3	121.0	158.7	264.9	246.8	246.8	82.0	97.2	152.9	206.5	241.7
time spent	1758.2	531.1	709.9	794.2	1243.1	1153.1	1265.7	480.3	574.9	733.5	969.5	1105.8
robot utilization	64.5	63.4	63.1	75.1	79.3	79.5	72.8	63.2	62.9	77.6	79.9	81.7
greedy												
number of bids	213.6	60.0	241.4	550.7	1075.4	1690.0	203.6	60.0	239.2	540.5	1005.4	1591.9
travel distance	291.1	50.0	92.5	123.5	228.6	261.2	247.5	45.2	79.8	116.1	178.0	231.8
time spent	1645.2	568.8	716.8	778.6	1378.4	1584.4	1247.8	514.9	569.1	685.0	951.6	1148.9
robot utilization	66.7	32.8	49.8	61.0	61.5	63.1	73.9	32.8	52.4	64.7	72.6	75.8
graph-cut												
number of bids	192.0	48.0	182.7	408.0	719.4	1095.6	183.9	48.0	182.7	403.2	717.0	1112.1
travel distance	292.0	49.9	84.4	129.7	185.0	210.0	244.4	45.1	73.1	111.1	170.8	184.1
time spent	2137.1	568.0	809.8	1116.4	1200.1	1470.7	1315.2	514.1	655.9	725.4	911.2	1076.2
robot utilization	53.2	32.8	39.3	49.0	58.6	53.2	69.5	32.8	42.6	60.7	71.2	63.9
Optimal												
travel distance							208.7	47.6	77.4	115.5	134.8/172.2*	151.4/195.9*

of a door is discovered when a robot reaches it and is broadcast to all the other robots.

- ***Completely Known Environment:*** In this case, each robot knew a complete map of the building in advance, including which doors were closed.

Figure 2 shows the different trajectories taken by a team of two robots in a partially known ***c_2*** environment, depending on whether they used ***Graph-Cut*** or ***Single*** as their bidding strategy. (We use three robots in the experiments but for better visibility we present a screenshot with two robots.)

B. Results

Table I reports the following four performance metrics for each bidding strategy in partially known and completely known environments, respectively:

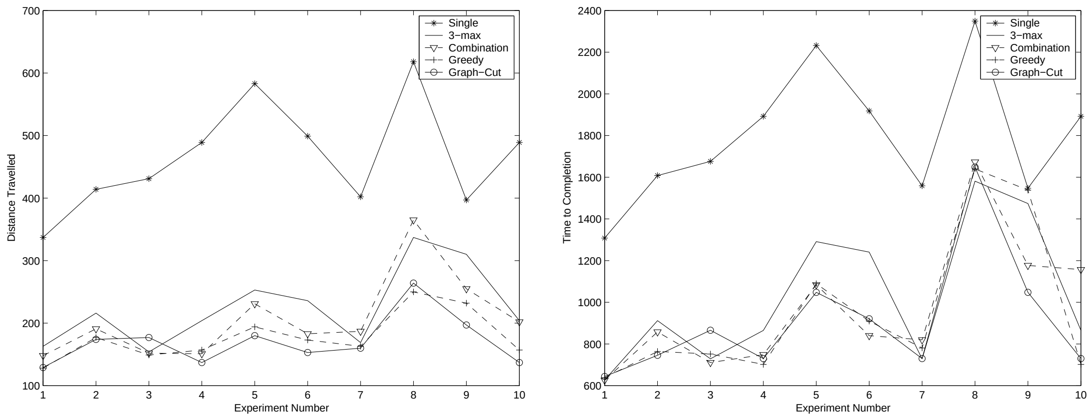
- "Number of bids" is the total number of bids submitted during an experiment.
- "Travel distance" is the total travel distance, that is, the sum of the travel distances of all robots, which roughly determines the amount of energy consumed.
- "Time Spent" is the total time it takes to solve the exploration task if communication and computation are instantaneous. (In the scheduling literature, this is known as the "makespan.")
- "Robot Utilization" reflects the percentage of time during which the robots were travelling.

Both the travel distance and the travel time are measures for how efficient the task allocation to robots is. The number of bids is a measure for the amount of communication and computation (for determining the bundles, the bids and the winners of the auctions). All numbers reported in the tables are the averages of ten experiments. As an example, Figure 3 shows the travel distances and travel times for each individual experiment in partially known environment ***c_3*** .

While we are mostly interested in the performance of different bidding strategies in partially known environments, we do not know how our combinatorial auction mechanism performs compared to the best possible solution since it is computationally intractable to compute the solution with minimal travel distance in partially known environments. However, we can determine the optimal solution in completely known environments by modeling the problem as a linear integer program (given in the appendix) and solving it with CPLEX, a commercial software package. The last row in Table I shows the results.

C. Discussion

We now discuss the results in Table I. In completely known environments, the travel distances and travel times tend to be smaller than in partially known environments, but our conclusions continue to hold, unless stated other-



(a) Travel Distance for the ***c_3*** Environment

(b) Travel Time for the ***c_3*** Environment

Fig. 3. Detailed Results for the Partially Known ***c_3*** Environment

wise. We first make the following key observations about the travel distances and travel times:

- *When the targets are distributed uniformly in partially known environments, the total travel distances of all bidding strategies are fairly close to each other.*
- *When the targets are clustered, the travel distances of combinatorial auctions are smaller than the ones of single auctions.* For example, the travel distance of ***Graph-Cut*** in environment ***c_4*** is about one-third of the travel distance of ***Single*** , as shown in Table I. In general, ***Single*** performs extremely poorly due to the fact that it sends all robots to the closest cluster, often taking them further away from the other clusters, as seen in Figure 2(a).
- *When the targets are clustered,* **Graph-Cut** *and* **Greedy** *result in smaller travel distances than the other two bidding strategies.* In fact, ***Graph-Cut*** and ***Greedy*** outperform the other two bidding strategies in almost every experiment performed in all clustered environments. Additionally, in every clustered environment, ***Graph-Cut*** perform better than ***Greedy*** , with the exception of ***c_3*** in the partially known environment, where the difference is due to only one run out of the total ten.
- *If the number of clusters is sufficiently large (that is, for the environments* **c_4** *and* **c_5** *in both the totally and partially known cases),* **Graph-Cut** *also results in smaller travel times than* **Greedy** . This is closely linked to the number of robots used. As ***Greedy*** allocates goals to all robots, it becomes slower than ***Graph-Cut*** as demonstrated by the lower robot utilization of ***Graph-Cut*** .

In completely known environments, the travel distance of ***Graph-Cut*** in environments with clustered targets is approximately equal to the optimal travel distance, and ***Graph-Cut*** remains tractable as the number of targets increases whereas the optimal algorithm becomes computationally infeasible. For example, we were not able to obtain the optimal solution for most problems in environments ***c_4*** and ***c_5*** after hours of running time. The results marked with an asterisk in the Tables I are the best solution so far but are not optimal. It should be noted that in all clustered environments, ***Graph-Cut*** has a smaller travel distance than ***Optimal*** . This is due to the fact that ***Graph-Cut*** navigates using DStar Lite which discretizes the environment, causing the robots to consider a target as being reached when it has entered the cell where said target is located, which might not be its exact location.

We make the following key observation about the number of bids:

- In terms of the number of bids, it holds that ***Single*** $\leq$ ***Graph-Cut*** $\leq$ ***Greedy*** $\leq$ ***3-max*** in every environment. Additionally, ***3-max*** and ***Combination*** fluctuate for environments ***c_1*** and ***c_2*** , but starting at ***c_3*** , ***Combination*** has a smaller number of bids.

In general, one would expect a trade-off between the travel distances (or travel times) and the number of bids because higher numbers of bids allow the bidders more flexibility in expressing synergies. However, it is undesirable to have large numbers of bids since they increase the communication and computation time. Consequently, it is important to develop bidding strategies that achieve small travel distances (or travel times) with a small number of bids. It is thus interesting to observe that ***Graph-Cut*** outperforms the other three combinatorial bidding strategies in environments with clustered targets not only with respect to travel distance (and, for a sufficiently large number of clusters, travel time) but also with respect to the number of bids. Thus, we can argue that ***Graph-Cut*** selects better bundles to bid on and, hence, is able to achieve better performance with a smaller number of bids. This result emphasizes the importance of the bidding strategies on performance. If the bids are formulated cleverly, one can achieve better results with fewer bids and performance thus does not have to come at the expense of increased communication and computation.

We make the following key observation about the utilization:

- In terms of the utilization, it holds that ***Graph-Cut*** $\leq$ ***Greedy*** $\leq$ ***Combination*** $\leq$ ***3-max*** $\leq$ ***Single*** in every environment except for ***uniform*** in the partially known environment and ***c_1*** where ***3-max*** has a smaller utilization than ***Combination*** .

In general, one would expect a trade-off between the travel time and the utilization because higher number of robots allow for more parallelism. Generally, a higher utilization implies a greater travel distance and lower travel time. This is true notably in environments with a small amount of clusters such as ***c_1*** or ***c_2*** . It is also clearly noticeable in ***c_5*** in both the known and unknown environments where ***Graph-Cut*** has a smaller utilization and travel distance than ***3-max*** but a higher travel time. It is thus interesting to observe that ***Graph-Cut*** outperforms the other three bidding strategies in environments with clustered targets with respect to travel distance (and, for a sufficiently large number of clusters, travel time) even though it never uses more robots. Thus, we can argue that ***Graph-Cut*** uses the robots in a smarter way and, hence, is able to achieve better performance with a smaller number of robots. If robots are costly or if it is difficult to operate a high number of robots for other reasons (for example, because one human operator has to control all robots at the same time), it might be desirable to complete the tasks utilizing fewer robots, in which case ***Graph-Cut*** is a clear winner. However, if reliability is important, then having a large number of robots active at all times might be desirable as an insurance against possible failures. If this is the case, then it is possible to modify the auction mechanism to increase robot utilization. For example, one can easily impose upper bounds on bundle sizes to ensure a more balanced allocation of targets to the robots. This partially explains why ***3-max*** and ***Combination*** result in higher utilization compared to ***Greedy*** and ***Graph-Cut*** . The maximum bundle size in ***3-max*** is three and six for ***Combination*** , whereas there is no upper bound on bundle sizes in the other two strategies. Therefore, when there are fewer clusters of targets, bids with larger bundles tend to win in ***Greedy*** and ***Graph-Cut*** resulting in fewer number of winners and hence, fewer number of active robots.

V. Conclusion

In this paper, we studied the problem of coordinating a team of mobile robots to visit a set of given targets in partially known terrain. We investigated the team performance when robots are coordinated using an auction mechanism. Our experimental results show a substantial advantage of combinatorial auctions over single-item auctions. They also show the large influence of the bidding strategy on the team performance, where our ***Graph-Cut*** strategy clearly outperformed three other strategies. Our future work will concentrate on developing more sophisticated bidding strategies and porting our code to ATRV Minis.

Acknowledgements

This research is partly supported by an NSF award under contract ITR/AP-0113881. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the sponsoring organizations, agencies, companies or the U.S. government.

Appendix

We model the exploration task in known environments as a linear integer program. Our goal is to coordinate a team of mobile robots to visit a set of predetermined targets with the objective of minimizing the travel distance. Suppose there are m targets and n robots.

- $T = \{1, \ldots, m\}$: the set of targets.
- $R = \{1, \ldots, n\}$: the set of robots.
- $L = \{0, 1, \ldots, m+n\}$: the set off all locations where $1 \leq j \leq m$ correspond to target locations and $n+1 \leq j \leq m+n$ correspond to the initial locations of the robots. 0 denotes a dummy location that represents the beginning and end of each path traversed by the robots.
- c_{ij}: the length of the shortest path from location i to location j, $\forall i, j \in L$. $c_{0j} = c_{j0} = 0$, $\forall j$.
- $x^r_{ij} = \begin{cases} 1 & \text{if robot } r \text{ visits} \\ & \text{location } j \text{ right after } i \quad \forall i, j \in L \\ 0 & \text{otherwise} \end{cases}$
- u^r_i: auxiliary continuous decision variables, $r \in R$, $i \in L - \{0\}$. If robot r visits location j u^r_j denotes the distance traveled by the robot until reaching location j. We use variables u^r_i to eliminate subtours, such that for each robot the solution to this linear integer model defines a list of targets to visit, and the sequence of visiting them.

The linear integer program is as follows:

$$min \sum_{r \in R} \sum_{i,j \in L} c_{ij} x^r_{ij}$$

subject to the following constraints:

$$x^r_{0,m+r} = 1, \quad r \in R \tag{1}$$

$$\sum_{r=1}^{n} \sum_{i=1}^{m+n} x^r_{i,j} = 1, \quad j \in T \tag{2}$$

$$\sum_{i=0}^{m+n} x_{ij}^r = \sum_{q=0}^{m+n} x_{jq}^r, \quad r \in R, \ j \in L \tag{3}$$

$$\sum_{j=0}^{m+n} x_{j,0}^r = 1, \quad r \in R \tag{4}$$

$$u_i^r - u_j^r + (|T|+2)x_{ij}^r \leq |T|+1, \quad \forall i,j \in T, r \in R \tag{5}$$

$$\sum_{i=0}^{m+n} x_{ij}^r \leq 1, \quad \forall \quad r \in R, \ j \in L \tag{6}$$

$$x_{ij}^r \in \{0,1\}, \quad \forall i,j \in L, \ r \in R \tag{7}$$

$$u_i^r \geq 0, \quad \forall r \in R, \ i \in L - \{0\} \tag{8}$$

The objective is to minimize the travel distance. Constraints (1) ensure that the first location of each robot is its current location. Constraints (2) ensure that each target is visited by exactly one robot. Constraints (3) ensure that a robot leaves a location after a visit. Constraints (4) ensure that the last location visited by each robot is the dummy location 0. Constraints (5) ensure that the solution of the linear integer program defines a sequence of targets. Constraints (6) ensure that a robot visits a target no more than once. Constraints (7) and (8) ensure that the variables x_{ij}^r and u_i^r are binary and nonnegative, respectively.

Bibliography

[1] A. Stentz and M. Diaz, "A free market architecture for coordinating multiple robots," Tech. Rep. CMU-RI-TR-99-42, Robotics Institute, Carnegie Mellon University, Pittsburgh (Pennsylvania), 1999.

[2] T. Balch and A. Ram, "Integrating robotics research with JavaBots," in *Proceedings of the AAAI Spring Symposium*, 1998.

[3] P. Stone and M. Veloso, "Multiagent systems: A survey from a machine learning perspective," *Autonomous Robots*, vol. 8, no. 3, 2000.

[4] M. Wellman, "The economic approach to artificial intelligence," *ACM Computing Surveys*, vol. 7, no. 3, pp. 360–362, 1995.

[5] B. Gerkey and M. Mataric, "Sold! Auction methods for multi-robot coordination," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 758–768, 2002.

[6] R. Nair, T. Ito, M. Tambe, and S. Marsella, "Task allocation in the rescue simulation domain: A short note," in *RoboCup-2001: Robot Soccer World Cup V*, A. Birk and S. Coradeschi, Eds., Lecture Notes in Computer Science. Springer, 2002.

[7] R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. Younes, "Coordination for multi-robot exploration and mapping," in *Proceedings of the National Conference on Artificial Intelligence*, 2000, pp. 852–858.

[8] R. Zlot, A. Stentz, M. Dias, and S. Thayer, "Coordination for multi-robot exploration and mapping," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2002.

[9] M. Diaz and A. Stentz, "Opportunistic optimization for market-based multirobot control," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2002.

[10] S. de Vries and R. Vohra, "Combinatorial auctions: A survey," *INFORMS Journal of Computing*, (forthcoming).

[11] N. Nisan, "Bidding and allocation in combinatorial auctions," in *Proceedings of the ACM Conference on Electronic Commerce*, 2000, pp. 1–12.

[12] T. Sandholm, "Algorithm for optimal winner determination in combinatorial auctions," *Artificial Intelligence*, vol. 135, no. 1-2, pp. 1–54, 2002.

[13] T. Sandholm and S. Suri, "Improved algorithms for optimal winner determination in combinatorial auctions and generalizations," in *Proceedings of the National Conference on Artificial Intelligence*, 2000, pp. 90–97.

[14] Y. Fujishima, K. Leyton-Brown, and Y. Shoham, "Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 1999, pp. 1–12.

[15] D. Lehmann, L. O'Callaghan, and Y. Shoham, "Truth revelation in approximately efficient combinatorial auctions," in *Proceedings of the ACM Conference on Electronic Commerce*, 1999, pp. 96–102.

[16] W. Elmaghraby and P. Keskinocak, "Combinatorial auctions in procurement," in *to appear in The Practice of Supply Chain Management*, C. Billington, T. Harrison, H. Lee, and J. Neale, Eds. Kluwer, 2002.

[17] E. Zurel and N. Nisan, "An efficient approximate allocation algorithm for combinatorial auctions," in *Proceedings of the ACM Conference on Electronic Commerce*, 2001.

[18] S. Koenig and M. Likhachev, "Improved fast replanning for robot navigation in unknown terrain," in *Proceedings of the International Conference on Robotics and Automation*, 2002.

[19] N. Nilsson, *Problem-Solving Methods in Artificial Intelligence*, McGraw-Hill, 1971.

[20] Y. Ye, "Software: COPL_DSDP," in *ftp://dollar.biz.uiowa.edu/pub/yyye/copldsdp.zip*.