Improving Combinatorial Auctions for Multi-Robot Exploration

Rodolfo C. Cavalcante Thiago F. Noronha Luiz Chaimowicz

***Abstract*—The use of multiple robots in exploration missions has attracted much attention in recent years. Here we deal with the specific problem in which a team of robots have to visit a set of target points to perform some action. Robots have a map of the environment and should compute and execute paths in a distributed way, trying to minimize the total mission cost that is dependent on the quality of the target-to-robot allocation. In this paper, we focus on this target allocation problem and use combinatorial auctions to solve it. We propose novel approaches for improving combinatorial auction mechanisms in the target allocation problem and compare them with approaches based on single-item auctions, sequential auctions, and other combinatorial auction algorithms. Experimental results showed that the auction approaches for multi-robot target allocation proposed in this work achieved better results than other auction based mechanisms found in the literature.**

I. INTRODUCTION

The exploration problem is considered one of the fundamental problems in mobile robotics [1]. The term exploration is used in the literature to define both (i) the problem in which robots must build a map of an unknown environment [2], and (ii) the problem in which the robots have a model of the environment and must explore some areas of interest in that environment [3], [4]. We deal with the second problem, in which there is a team of robots and a fixed number of target points that must be visited by exactly one robot from the team. The objective is to provide a fully distributed approach to coordinate the robots in order to minimize the total mission cost of visiting all the targets. This study is motivated by applications such as surveillance and monitoring, search and rescue, and data collection in sensor networks, among others. In all these applications, the target-to-robot allocation is a critical step of the coordination approach.

More formally, the problem tackled in this paper can be defined as follows. Given a set of robots $R = \{r_1, r_2, \dots, r_n\}$, a set of targets $T = \{t_1, t_2, \dots, t_m\}$, whose coordinates are known beforehand, the exploration problem consists in (i) finding an assignment $S = \{S_1, S_2, \dots, S_n\}$ of targets to robots, where $S_i \subseteq T$ denotes the set of targets assigned to robot $r_i \in R$, with $S_i \cap S_j = \emptyset$ for all $i \neq j$, and $\bigcup_{i=1}^{m} S_i = T$, and then (ii) computing a route P_i for each robot r_i such that all targets in S_i are visited once and that the robot returns to its initial position. The objective is to minimize the sum of travel distances over all robots. This problem was proven to be NP-Hard in [5].

The authors are affiliated to the Computer Vision and Robotic Laboratory (VeRLab), Computer Science Department, Universidade Federal de Minas Gerais, Brazil. {rodolfocc,tfn,chaimo}@dcc.ufmg.br

The authors would like to thank CNPq, CAPES and FAPEMIG for their support.

978-1-4799-2722-7/13/$31.00 ©2013 IEEE

Single-Item Auctions [6] and Sequential Single-Item auctions [7] are distributed mechanisms that have been widely used to coordinate multiple robots in exploration missions because they are easy to design and implement, besides presenting low computational complexity compared with centralized solutions. But due to their limitations, these mechanisms may not be able to provide good allocations. Combinatorial Auctions [8], on the other hand, are able to provide better solutions since these mechanisms allow robots to consider the value of combinations or packages of targets. However, if robots are allowed to bid on all possible combinations of targets, the allocation problem becomes exponential.

In this paper, we propose and investigate different combinatorial auction strategies for limiting the possible combinations of targets the robots can bid on in order to make the target allocation a polynomial process. The proposed strategies were implemented and compared with other auction strategies found in the literature based on single-item auctions, sequential auctions, and other combinatorial auctions. Computational experiments showed that the auction strategies for multi-robot target allocation proposed in this work achieved better results than other strategies in the literature.

The remainder of this paper is organized as follows. In Section II, we review the main auction mechanisms used to coordinate multiple robots in exploration missions in the literature. In sections III and IV, we propose new combinatorial auction mechanisms to be used in robot coordination approaches. In Section V, these mechanisms are evaluated and compared against other auction based coordination approaches found in the literature. Concluding remarks are presented and opportunities for future works are discussed in the last section.

II. AUCTIONS IN MULTI-ROBOT EXPLORATION

Auction-based mechanisms have been used to coordinate multiple robots in exploration missions, due to their simplicity and flexibility [9], [10], [11], [12]. The targets that the robots must visit are the items being auctioned and the value of the bids are the expected utility cost for the robots to visit the corresponding targets. The robots compete for the targets trying to maximize their individual performances. Although the robots are selfish, the maximization of individual rewards results in the maximization of the whole team reward. Auctions are efficient both in communication cost, since the informations exchanged by the robots are just numerical bids, as in computation cost, since each robot compute its bids in parallel [13].

A. *Single-Item and Sequential Auctions*

Single-item auctions are those in which a robot bids on a target independently of the values he bid for the other targets it won in the past auction rounds [12], [14], [15]. They are distributed mechanisms that have been used to coordinate robots in exploration missions because they are easy to design and implement. Besides, they present low computational complexity compared with centralized solutions. However, due to their limitations, these mechanisms might not provide good allocations in applications where the utility cost of visiting a subset of targets with the same robot is smaller than that of visiting the same targets with different robots.

Sequential Single-Item auctions (SSI) are an extension of single-item auctions in which the value a robot bids for a target is calculated taking into consideration the targets the robot won in the past rounds of the auction [5], [7]. This mechanism may increase the performance of the single-item auctions since targets that are close to the other targets won by a robot in previous rounds have smaller utility costs than those that are not. Therefore, targets that are close to each other are more likely to be won by the same robot. However, as SSI auctions are greedy allocation mechanisms, the quality of the allocation they provide greatly depends on the order the targets are auctioned.

B. Combinatorial Auctions

Combinatorial auctions have been used in a wide range of applications, such as transportation logistics, scheduling of bus routes, allocation of airport arrival and departure slots, etc (see [8] for a full review). In this type of auction, agents can bid on a subset of items (also called *package of items*), instead of only on a single item [16]. This allows the bidder to bid on a subset of items that are only profitable if they are acquired together. In this case, either the bidder wins all items in the subset or it wins none of them. Basically, these auctions have two steps. In the *bid formulation step*, bidders decide and communicate to the auctioneer in which subsets of items they will bid, and in the *winner determination step*, the auctioneer decides which bids have higher profits and determine the winners. The main limitations of this auction mechanism are that (i) the number of subsets of items one can bid is exponentially large, and (ii) the winner determination step is NP-Hard in the general case [16]. Therefore, it is necessary to use strategies that limit the subset of items one can bid, such that the winner determination step can be done in polynomial time.

The *Package Tree Strategy* [17] consists in organizing the candidate packages of items in a tree structure, where each node is a package that robots can bid during the auction process. The child nodes contain disjoints subsets of the parent node items, and the union of all packages in the child nodes is equal to the package of the parent node. The root node is composed by all items to be auctioned, and the leafs are composed by a single item. Using this structure, the winner determination can be done recursively in polynomial time as described in [17].

The *Sorting Strategy* [17] is based on the fact that (i) if the items to be auctioned are sorted according to any criterion and (ii) the bids are restricted only to those packages made of consecutive items according to the sorting, then the winner determination can be done in polynomial time by a dynamic programming algorithm as described in [17].

As far as we know, the only works in the literature that apply combinatorial auctions to robot exploration are [18] and [19]. In [18], a package-tree based combinatorial auction mechanism is proposed for an area reconnaissance problem, which is different from the problem tackled here. On the other hand, Berhault et al. [19] tackles the same exploration problem studied in this work. They proposed four strategies to limit the possible combinations of targets the robots can bid on, described below.

The first and the simplest strategy consists of limiting the number of targets in each package. According to this strategy, robots can bid on any package with less targets than the established limit. In this case, the winner determination problem is solved by inspection.

The second strategy rely on pre-defined information about clusters of targets, with sizes varying from three to six, that were observed in the environment. According to this strategy, robots can bid on all packages containing one or two targets, as well as on packages composed by sets of pre-defined targets clustered in the environment. In this case, the winner determination problem is also solved by inspection.

The third combinatorial auction mechanisms of [19] is based on bidding on good sequences of targets. The sequences are recursively created by starting with just one target $t_i \in T$ and adding to this sequence the target $t_j \in T$ that is the closest to the last target in the sequence if the surplus of the new sequence (the reward minus the cost of visiting the targets) is greater than or equal to the surplus of the old sequence and t_j is the closest target to the last target in the old sequence.

The last and more sophisticated strategy proposed in [19] models the problem as a complete weighted graph in which the vertices correspond to targets and the edge weights are the travel costs between them. It builds a binary package-tree as follows. The root node consists of a package with all targets being auctioned. First, a max-cut problem [20] is solved in order to divide the vertices in this package into two disjoint packages. Next, both packages are placed as child nodes of the root node. Then, the process is recursively applied to every child node that has more than one target. Robots can bid on any package in this tree, and the winner determination problem is solved with the same algorithm used in [18]. Computational experiments showed that this is the best strategy among those studied in that work.

In this paper, we propose three new different strategies to improve combinatorial auctions in multi-robot exploration tasks. The C-REG and C-TSP are package-tree strategies and are presented in Section III. They use package-trees in a similar way as [19], but are based on heuristics for the Traveling Salesman Problem (TSP) [21] for constructing the package tree. The C-SORT is a sorting strategy and is presented in Section IV. As far as we know, this is the first work in the literature to propose an item sorting based combinatorial auction to perform the target-to-robot allocation.

III. PACKAGE TREES

The first strategy, called *C-REG*, is a top-down approach that builds a package tree from the root to the leafs. It is based on the region partition method proposed in [22] and used in [23] to heuristically solve the TSP. It builds a binary package-tree as follows. The root node has all the targets being auctioned. Next, the smallest rectangle that contains all the targets in this node is computed. Then, this rectangle is divided into two smaller rectangles with the same area. The items in one of these rectangles are placed in one of the child nodes of the current node, and those in the other rectangle are placed in the other child node. The process is recursively applied to the new sub-nodes that have more than one target. Figure 1 shows some of the steps of this algorithm.

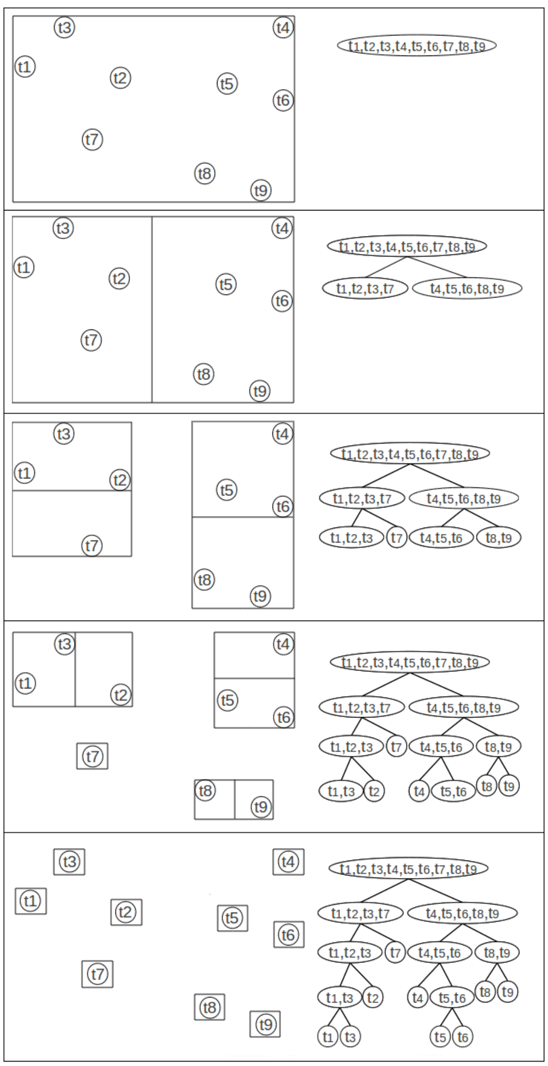

Fig. 1. Building a package tree using C-REG: as the region is being partitioned, new nodes are created.

The second strategy, called *C-TSP*, is a bottom-up approach that builds a package tree from the leaf to the root node. It builds a binary package-tree as follows. Let $G = (V, E)$ be a complete weighted graph, where targets are represented by the vertices in V, and the edge weights are the distances between the targets. Initially, each target is a leaf node on the package tree. The algorithm iteratively tries to group targets in order to create larger packages in the tree. In each iteration, there is a threshold that limits which targets can be grouped according the TSP cost formed by these targets. This cost is computed using the *Farthest Insertion* (FI) heuristic [24]. If some targets, when grouped in a package, presents a TSP cost bellow the threshold, the method creates a new node in the package tree containing all clustered targets, and the targets become child nodes of the new one. When no package can be further grouped, the threshold is incremented and a new iteration begins. The algorithm tries to group smaller packages into larger packages according to the TSP. The process continues until the TSP threshold allows the clustering of all targets in the environment, and this group is the root of the package tree. Figure 2 shows the execution of this algorithm.

After the creation of the package tree using one of these two strategies, robots can bid on any package in the tree. The bid value for a package is the TSP cost to visit all targets in the package. This cost is computed with the FI Heuristic for the TSP. The winner determination is performed in two steps. In the first step, the auction process decides which are the winner bids for each package individually. In the second step, a recursive algorithm decides which packages are more profitable. The algorithm works as follows. The base case is a leaf package, which is considered to be the most profitable. To decide on a non-leaf package, the algorithm compares the bid value for the package and the sum of bids for its children. If the bids for children nodes are more profitable than the parent package, the children packages are marked as allocated. Otherwise, the parent package is marked as allocated. The procedure continues recursively until deciding between the root node and its children. At the end, the packages marked as allocated will be effectively allocated to the robots that bid on them.

IV. TARGET SORTING

The third strategy, called *C-SORT*, is a sorting strategy also based on the FI heuristic for TSP. It sorts the targets and builds a matrix of bids for each consecutive interval of items as follows. Let $G = (V, E)$ be a complete weighted graph, where targets are represented by the vertices in V, and the edge weights are the distances between the targets. First, C-SORT builds a TSP cycle with all the targets in V using FI. Next, it converts this cycle into an item permutation by removing one edge arbitrarily. Once the targets are sorted, robots create a *bid matrix*, formulating bids for each of the $n^2/2$ possible consecutive combination of targets that obey that order. Figure 3 shows an example of this matrix for a sequence of 8 targets $\{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8\}$. In the first line of the matrix there are bids for the intervals with just one target, such as $[t_1, t_1]$ = $\{t_1\}$, $[t_2, t_2]$ = $\{t_2\}$, $[t_3, t_3]$ = $\{t_3\}$, and so on. In the second line of the matrix there are bids for intervals with two targets, such as $[t_1, t_2]$ = $\{t_1, t_2\}$, $[t_2, t_3]$ = $\{t_2, t_3\}$. In the next line, there are bids for intervals

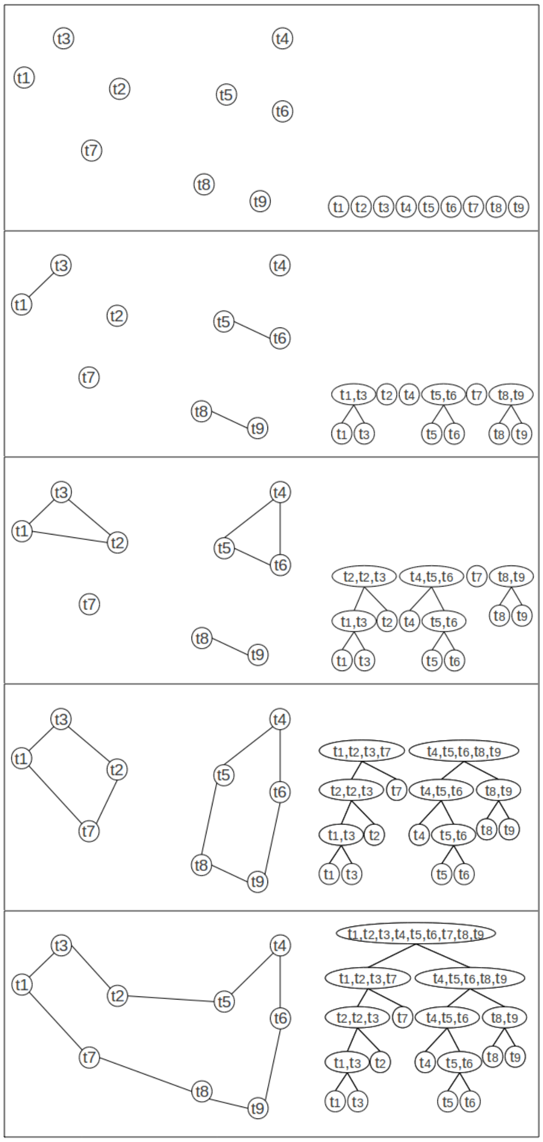

Fig. 2. Building a package tree using C-TSP: targets are incrementally clustered according to the cost of a TSP circuit.

Order: $t_1,t_2,t_3,t_4,t_5,t_6,t_7,t_8$

$[t_1,t_1]$	$[t_2,t_2]$	$[t_3,t_3]$	$[t_4,t_4]$	$[t_5,t_5]$	$[t_6,t_6]$	$[t_7,t_7]$	$[t_8,t_8]$
$[t_1,t_2]$	$[t_2,t_3]$	$[t_3,t_4]$	$[t_4,t_5]$	$[t_5,t_6]$	$[t_6,t_7]$	$[t_7,t_8]$	
$[t_1,t_3]$	$[t_2,t_4]$	$[t_3,t_5]$	$[t_4,t_6]$	$[t_5,t_7]$	$[t_6,t_8]$		
$[t_1,t_4]$	$[t_2,t_5]$	$[t_3,t_6]$	$[t_4,t_7]$	$[t_5,t_8]$			
$[t_1,t_5]$	$[t_2,t_6]$	$[t_3,t_7]$	$[t_4,t_8]$				
$[t_1,t_6]$	$[t_2,t_7]$	$[t_3,t_8]$					
$[t_1,t_7]$	$[t_2,t_8]$						
$[t_1,t_8]$							

Fig. 3. Example of a bid matrix used in the C-Sort Algorithm.

with three targets, such as $[t_1, t_3]$ = $\{t_1, t_2, t_3\}$, and $[t_2, t_4]$ = $\{t_2, t_3, t_4\}$, for example. This process continues until the last line, in which there is a bid for all the targets, $[t_1, t_8]$ = $\{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8\}$.

Each robot in the team computes its bid matrix and broadcasts it to all the robots. The winner determination occurs in two phases. First, the auction decides which are the winner bids for each interval represented by a matrix cell. For each cell on the bid matrix, the auctioneer just compare the bid values on each corresponding matrix cell sent by robots. A new matrix is generated containing only the winner bid and the respective winner robot for each matrix cell. Then, the winner determination problem is to decide which packages corresponding to each matrix cell will be assigned to robots based on the generated matrix. This decision can be done by a dynamic programming algorithm proposed in [17].

V. EXPERIMENTAL RESULTS

The coordination approaches based on Single-Item Auction (SI) [13], Sequential Single-Item Auction (SSI) [7], the graph-cut combinatorial auction (C-MAX) [19], and those proposed in this paper: C-REG, C-TSP, and C-SORT were implemented in C++ and compiled with g++ version 4.4.5. The simulations were performed in the Player/Stage Simulator version 4.0.1 running on an Intel Core i5 machine with 4 GB of DDR3 RAM memory. The implemented auctions are fully distributed, in the sense that there is no central auctioneer responsible for receiving bids from each robot and compute the winners. Each robot acts as bidder and auctioneer, sending its bids to all other teammates and computing the winners individually and locally. For this, we consider a perfect communication scenario.

Due to the limited space, we only show here results for simulations performed on six different scenarios. In the first three scenarios, 40 targets are uniformly distributed in an environment with size of 282x282m, and the number of robots was 3, 5, and 9, respectively. In the other three scenarios, the 40 targets are non-uniformly distributed in the environment, and the number of robots was 3, 5, and 9, respectively. The non-uniform distribution works as follows. Firstly, the environment is divided into 16 squares of the same size. Next, five of these squares are selected at random. Then, one fifth of the total number of targets were uniformly distributed inside each of the five selected squares. For each scenario, 30 runs of each coordination approach were performed, varying the seed of the random number generator used to distribute the targets in the environment.

Figures 4 and 5 show the box plots that summarize the results of the simulations for the six coordination approaches on the first three scenarios (those with uniformly distributed targets) and the last three scenarios (those with non-uniformly distributed targets), respectively. Each box plot shows, from bottom to top, the smallest sum of traveled distances, the lower quartile, the median, the upper quartile, and the largest sum of traveled distances observed over the 30 runs. It can be seen that the best results were obtained with combinatorial auctions C-TSP and C-SORT, as they displayed the smallest medians and the smaller quartiles on all scenarios. C-TSP was slightly better than C-SORT in the scenarios with non-uniformly distributed targets, while the opposite happened in the other there scenarios. It can also be seen that C-MAX obtained similar (some times worse) results than SI and SSI auctions. Besides, C-TSP and C-SORT clearly outperformed C-MAX in all scenarios.

Figures 6 and 7 show the average computation time (in log scale) consumed by the six coordination approaches on the first three scenarios and the last three scenarios, respectively. It can be seen that single-item and sequential single-item auctions had the shortest execution times, while the combinatorial auction C-SORT had the longest computation times. This is due to the fact that robots need to compute bids for $O(n^2)$ packages in the bid matrix, while the other combinatorial auctions compute bids only for $O(n)$ packages in the package tree. Besides, the larger was the number of robots in the system, the larger was the running time, since there are more bids to process during the winner determination.

VI. CONCLUSIONS

In this work, we investigated the use of combinatorial auctions to target-to-robot allocation in exploration missions. We proposed new coordination approaches based on market-oriented combinatorial auctions. We proposed two package-tree based auction mechanisms, one top-down (C-REG) and the other bottom-up (C-TSP). Besides, we proposed an auction mechanism (C-SORT) based on sorted sequences of targets obtained with a TSP heuristic. As far as we know, this is the first work in the literature to propose an item sorting based combinatorial auction to perform the target-to-robot allocation.

Simulations were performed to compare these approaches with other approaches based on single-item auctions, sequential auctions, and other combinatorial auction mechanisms found in the literature. Experimental results showed that C-TSP achieved better results than the other approaches in the scenarios where targets are grouped in some regions of the environment. Besides, they showed that in the scenarios where targets are uniformly distributed, C-SORT outperformed C-TSP and the other approaches. However, the experiments also showed that C-SORT has the largest computation times among the coordination approaches studied in this work.

Future works can apply the combinatorial auctions proposed here for more complex exploration problems, or propose new combinatorial auction mechanisms that are more efficient or more flexible than those proposed in this paper.

REFERENCES

[1] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz, and B. Stewart, "Distributed multi-robot exploration and mapping," in *Proceedings of the IEEE*, 2006, pp. 1325–1339.

[2] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Monterey, CA, 1997, pp. 146–151.

[3] K. Zhang, E. G. Collins, and A. Barbu, "A novel Stochastic Clustering Auction for task allocation in multi-robot teams," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, 2010, pp. 3300–3307.

[4] X. Zheng and S. Koenig, "K-Swaps: Cooperative Negotiation for Solving Task-Allocation Problems," in *International Joint Conference on Artificial Intelligence*, Pasadena, USA, 2009, pp. 373–379.

[5] M. G. Lagoudakis, M. Berhault, S. Koenig, P. Keskinocak, and A. J. Kleywegt, "Simple auctions with performance guarantees for multi-robot task allocation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, 2004, pp. 698–705.

[6] M. Dias, R. Zlot, N. Kalra, and A. Stentz, "Market-Based Multirobot Coordination: A Survey and Analysis," *Proceedings of the IEEE*, vol. 94, pp. 1257–1270, 2006.

[7] S. Koenig, C. Tovey, M. G. Lagoudakis, E. Markakis, D. Kempe, P. Keskinocak, A. J. Kleywegt, A. Meyerson, and S. Jain, "The power of sequential single-item auctions for agent coordination," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Boston, USA, 2006, pp. 1625–1629.

[8] P. Cramton, Y. Shoham, and R. Steinberg, *Introduction to Combinatorial Auctions*. MIT Press, 2005.

[9] S. Koenig, P. Keskinocak, and C. Tovey, "Progress on Agent Coordination with Cooperative Auctions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Atlanta, USA, 2010, pp. 1713–1717.

[10] M. G. Lagoudakis, E. Markakis, D. Kempe, P. Keskinocak, A. Kleywegt, S. Koenig, C. Tovey, A. Meyerson, and S. Jain, "Auction-Based Multi-Robot Routing," *Proceedings of the International Conference on Robotics: Science and Systems*, pp. 343–350, 2005.

[11] A. R. Mosteo and L. Montano, "Comparative experiments on optimization criteria and algorithms for auction based multi-robot task allocation," in *IEEE International Conference on Robotics and Automation*, Roma, Italy, 2007, pp. 3345–3350.

[12] R. Zlot, A. Stentz, M. B. Dias, and S. Thayer, "Multi-robot exploration controlled by a market economy," in *IEEE International Conference on Robotics and Automation*, Washington, DC, USA, 2002, pp. 3016–3023.

[13] B. P. Gerkey and M. J. Matarić, "Sold!: auction methods for multirobot coordination," *Robotics and Automation, IEEE Transactions on*, vol. 18, no. 5, pp. 758–768, oct 2002.

[14] R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. M., S. Thrun, and H. Younes, "Coordination for Multi-Robot Exploration and Mapping," in *Proceedings of the AAAI National Conference on Artificial Intelligence*. Austin, TX: AAAI, 2000.

[15] M. Nanjanath and M. Gini, "Dynamic task allocation for robots via auctions," in *IEEE International Conference on Robotics and Automation*, Orlando, USA, may 2006, pp. 2781–2786.

[16] T. Sandholm, "Algorithm for optimal winner determination in combinatorial auctions," *Artificial Intelligence*, vol. 135, pp. 1–54, 2002.

[17] M. H. Rothkopf, A. Pekec, and R. M. Harstad, "Computationally Manageable Combinatorial Auctions," *Management Science*, vol. 44, 1998.

[18] R. Zlot and A. Stentz, "Complex Task Allocation For Multiple Robots," in *IEEE International Conference on Robotics and Automation*, Barcelona, Spain, april 2005, pp. 1515–1522.

[19] M. Berhault, H. Huang, P. Keskinocak, S. Koenig, W. Elmaghraby, P. Griffin, and A. Kleywegt, "Robot exploration with combinatorial auctions," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, Las Vegas, USA, oct. 2003, pp. 1957–1962.

[20] M. R. Garey and D. S. Johnson, *Computers and intractability: A Guide to the Theory of NP-Completeness*. Freeman New York, 1979, vol. 174.

[21] E. L. Lawler, J. K. Lenstra, A. R. Kan, and D. B. Shmoys, *The traveling salesman problem: a guided tour of combinatorial optimization*. Wiley New York, 1985, vol. 3.

[22] R. Karp, "Probabilistic analysis of partitioning algorithms for the traveling-salesman problem in the plane," *Mathematics of Operations Research*, pp. 209–224, 1977.

[23] D. S. Johnson and L. A. McGeoch, "The Traveling Salesman Problem: A Case Study in Local Optimization," in *Local Search in Combinatorial Optimization*, 1997, pp. 215–310.

[24] D. J. Rosenkrantz, R. E. Stearns, and I. P. M. Lewis, "An analysis of several heuristics for the traveling salesman problem," *SIAM journal on computing*, pp. 563–581, 1977.

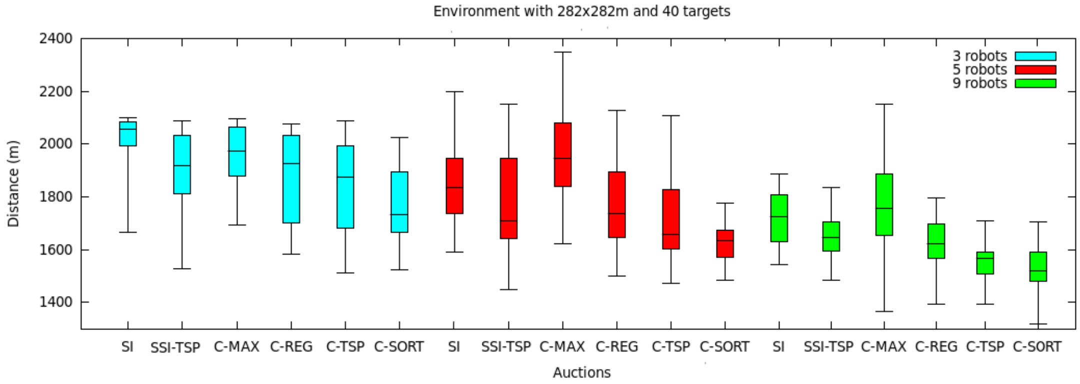

Fig. 4. Sum of traveled distances by robots in an environment with 40 targets uniformly distributed.

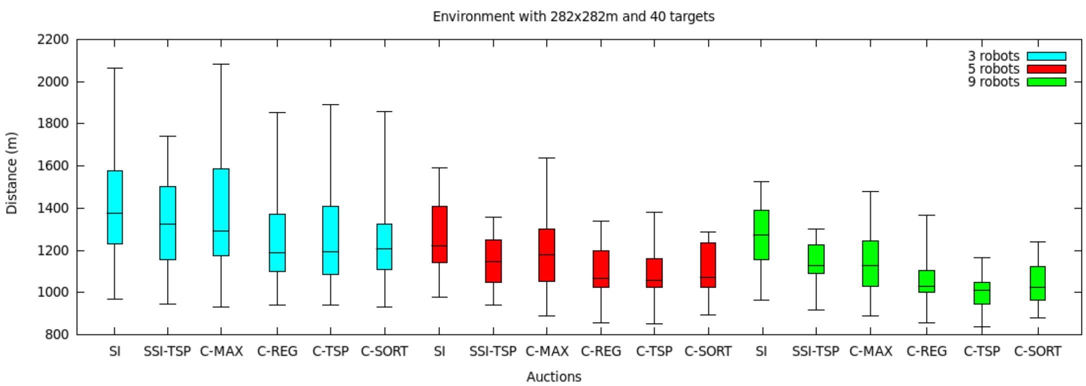

Fig. 5. Sum of traveled distances by robots in an environment with 40 targets non-uniformly distributed.

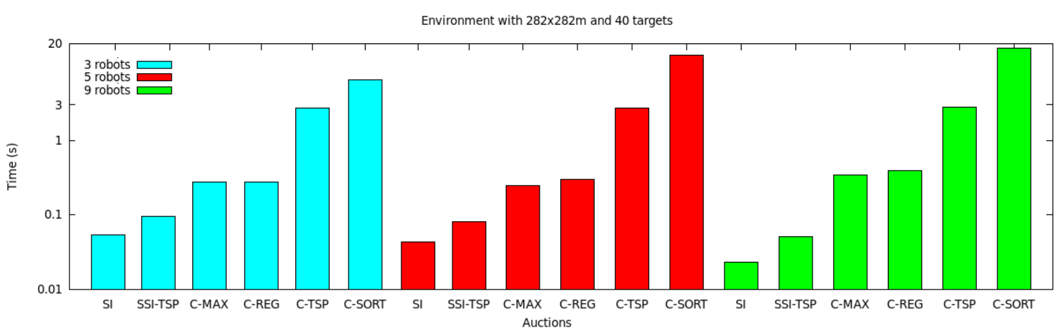

Fig. 6. Time consumed by the auctions to compute the allocations in scenarios with 40 targets uniformly distributed.

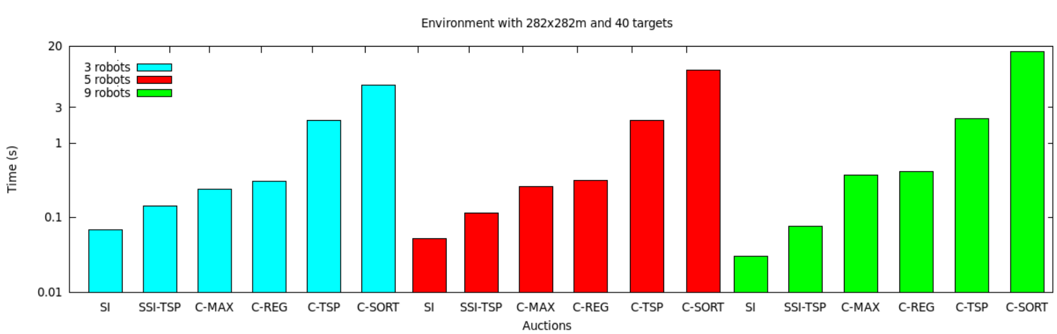

Fig. 7. Time consumed by the auctions to compute the allocations in scenarios with 40 targets nun-uniformly distributed.