Auction-Based Multi-Robot Routing

Michail G. Lagoudakis*, Evangelos Markakis†, David Kempe‡ Pinar Keskinocak*,
Anton Kleywegt*, Sven Koenig‡, Craig Tovey*, Adam Meyerson§, and Sonal Jain‡

* School of Industrial and Systems Engineering
Georgia Institute of Technology, Atlanta, GA 30332–0205
Email: {mlagouda, pinar, anton, ctovey}@isye.gatech.edu

† College of Computing
Georgia Institute of Technology, Atlanta, GA 30332–0280
Email: vangelis@cc.gatech.edu

‡ Computer Science Department
University of Southern California, Los Angeles, CA 90089-0781
Email: {dkempe, skoenig, sonaljai}@usc.edu

§ Computer Science Department
University of California, Los Angeles, CA 90095-1596
Email: awm@cs.ucla.edu

***Abstract*— Recently, auction methods have been investigated as effective, decentralized methods for multi-robot coordination. Experimental research has shown great potential, but has not been complemented yet by theoretical analysis. In this paper we contribute a theoretical analysis of the performance of auction methods for multi-robot routing. We suggest a generic framework for auction-based multi-robot routing and analyze a variety of bidding rules for different team objectives. This is the first time that auction methods are shown to offer theoretical guarantees for such a variety of bidding rules and team objectives.**

I. Introduction

Robot teams are increasingly becoming a popular alternative to single robots for a variety of difficult robotic tasks, such as planetary exploration or planetary base assembly. Robot teams offer many advantages over single robots: robustness (due to redundancy), efficiency (due to parallelism), and flexibility (due to reconfigurability). However, an important factor for the success of a robot team is the ability to coordinate the team members in an effective way. Coordination involves the allocation and execution of individual tasks through an efficient (preferably decentralized) mechanism.

In this paper, we focus on multi-robot routing, a class of problems where a team of mobile robots must visit a set of locations for some purpose (e.g., delivery or acquisition) with routes that optimize certain criteria (e.g., minimization of consumed energy, completion time, or average latency). Examples include search-and-rescue in areas hit by disasters, surveillance of a facility, placement of sensors, delivery of parts, and localized measurements. Such routing problems, including Vehicle Routing Problems (VRPs) and several variants of the Traveling Salesman Problem (TSP), have been studied from a centralized point of view in operations research, and recently in robotics with a focus on decentralized approaches.

Even in decentralized multi-robot coordination, some information exchange is necessary; it is desirable to enable good decision making while communicating as little information as possible. One promising approach of this type is the use of market-based mechanisms, in particular, auction-based methods, where the communicated information consists of bids robots place on various tasks, and coordination is achieved by a process similar to winner determination in auctions.

The efficiency of auction-based methods has been demonstrated experimentally [1]–[7], but there has been little theoretical study [6]. In this paper, we make the following contributions: (1) we suggest a generic framework for auction-based multi-robot routing, and (2) we derive and analyze six bidding rules for three team objectives (minimizing total cost, maximum cost, or average service cost), specifically, we provide lower and upper bounds on their performance relative to optimal performance. This is the first time that auction methods are shown to offer theoretical guarantees for such a variety of bidding rules and team objectives.

II. Multi-Robot Routing

A *multi-robot routing* problem is specified by a set of robots, $R = \{r_1, r_2, \ldots, r_n\}$, a set of targets, $T = \{t_1, t_2, \ldots, t_m\}$, their locations, and a non-negative cost function $c(i, j)$, $i, j \in R \cup T$, which denotes the cost of moving between locations i and j. We assume that these costs are symmetric, $c(i, j) = c(j, i)$, are the same for all robots, and satisfy the triangle inequality. Travel distances and travel times between locations satisfy these assumptions in any typical environment. The objective of multi-robot routing is to find an allocation of targets to robots and a path for each robot that visits all targets allocated to it so that a team objective is optimized. In this paper, we study three intuitive team objectives:

MiniSum: Minimize the sum of the robot path costs over all robots.
MiniMax: Minimize the maximum robot path cost over all robots.
MiniAve: Minimize the average target path cost over all targets.

The *robot path cost* of a robot r is the sum of the costs along its entire path, from its initial location to the last target on its path. The *target path cost* of a target t is the total cost of the path traversed by robot r from its initial location up to target t, where r is the unique robot visiting t.

III. Optimal Solutions

Optimizing performance for any of the three team objectives is NP-hard, as shown by the following theorem.

Theorem 1: There is no polynomial time algorithm for solving multi-robot routing optimally with the MiniSum, the MiniMax, or the MiniAve objective, unless P = NP.

Proof: We show that a polynomial-time algorithm for multi-robot routing with any of the three objectives implies a polynomial time algorithm for Hamiltonian Path, a well known NP-complete problem. An instance of Hamiltonian Path consists of a graph $G = (V, E)$ and a vertex v, and we are asked to decide if there exists a path starting from v that visits all the vertices exactly once. We reduce it to an instance of multi-robot routing as follows. Let $G' = (V, c)$ be the complete weighted graph on V with weights $c(u, w) = 1$, if $(u, w) \in E$, and $c(u, w) = 2$, otherwise. One robot is placed at vertex v and the remaining $|V| - 1$ vertices are designated as targets. The costs (weights) in G' satisfy the triangle inequality.

We claim that G has a Hamiltonian path if and only if an optimal MINISUM solution in G' has a cost of $|V| - 1$. A Hamiltonian path in G is also an optimal MINISUM solution in G' with cost $|V| - 1$ (this is the least possible cost for visiting $|V| - 1$ targets). Conversely, if G does not have a Hamiltonian path, then any path in G' that starts from v and visits all the vertices exactly once has to use some edge of cost 2 in G'. Hence, the cost of an optimal solution will be at least $|V|$. Similarly, G has a Hamiltonian path if and only if an optimal MINIMAX solution in G' has a cost of $|V| - 1$. Finally, G has a Hamiltonian path if and only if an optimal MINIAVE solution in G' has a cost of $\big(1 + 2 + \ldots + (|V| - 2) + (|V| - 1)\big)/\big(|V| - 1\big) = |V|/2$. ■

Given this hardness result, we focus on efficient approximation algorithms for solving large-scale instances of multi-robot routing. However, optimal solutions for small instances can be obtained through a mixed integer programming formulation.

IV. AUCTION FRAMEWORK

Our auction-based coordination system for multi-robot routing considers the robots as bidders and the targets as goods, and operates as follows. All targets are initially unallocated. During each round of bidding, all robots bid on all unallocated targets. The robot that places the overall lowest bid on any target wins and is allocated that particular target. A new round of bidding begins, and all robots bid again on all unallocated targets, and so on, until all targets have been allocated. Note that each robot needs to bid only on a single target in each round, namely on a target for which its bid is the lowest, since all other bids from the same robot have no chance of winning. Upon allocation of all targets, each robot computes a path for visiting the targets allocated to it and then moves along that path. Bid selection and path computation are the key factors that affect team performance.

The main advantage of this multi-round auction mechanism is its simplicity and the fact that it allows for a decentralized implementation on real robots. Initially, each robot needs to know its own location, the location of all targets, and the number of robots (the number of bids in each round), but not the locations of the other robots. In each round, each robot computes its single bid locally and in parallel with the other robots, broadcasts the bid to the other robots, receives the bids of the other robots, and then locally determines the winning bid. This procedure is repeated in every round of the auction. Broadcasting can be achieved by means of relaying messages from robot to robot. Clearly, there is no need for a central auctioneer, and therefore, there is no single point of global failure in the system. Notice also the low communication complexity; each robot needs to receive n numbers (bids) in each of the m rounds, therefore $O(nm)$ numbers need to be communicated over any single link.

V. PATHS VERSUS TREES

We explore two ways of obtaining approximate solutions to multi-robot routing within our auction framework: paths and trees. In particular, during the auction, one can consider constructing paths that collectively span all targets (one path for each robot), or constructing trees that span all targets (a forest with one tree rooted at each robot). Considering paths is a direct method, whereas considering trees is an indirect method, since the trees must be converted to paths to obtain a solution to the original problem. This extra step is relatively easy and does not significantly affect the quality of the solution.

The choice of paths versus trees depends on how efficiency and performance are affected. The rationale behind the idea of constructing trees rather than paths is that trees with certain properties might be readily computable compared to paths with similar properties. For example, given any weighted graph, a minimum-cost spanning tree can be obtained in polynomial time, whereas a finding a minimum-cost path through all nodes is an NP-hard problem. Therefore, instead of directly seeking paths that achieve a team objective, one may seek to find trees that achieve an analogous objective, and then convert the trees to paths that approximate the original team objective.

For MINISUM, the analogous objective is to find a minimum spanning forest (MSF), that is, a collection of trees rooted at the robots that span all targets with minimum total cost. Such a forest is computable in polynomial time by a variant of Prim's algorithm [8]. For MINIMAX, the analogous objective is to find a minimax spanning forest, which is a collection of trees rooted at the robots that span all targets such that the cost of the most expensive tree is minimized. Computing the minimax tree is an NP-hard problem [9]. Finally, for MINIAVE, the analogous objective is to find a minimum average-cost spanning forest which is a collection of trees rooted at the robots that span all targets such that the average root-target cost over all targets is minimized. Such a forest can be trivially computed by connecting each target to the closest root, and consists of stars.

A tree can be easily turned into a path using shortcutting [10], as commonly used in TSP algorithms. Shortcutting constructs a path from a tree by performing a depth-first search on the tree to derive the ordering of nodes in the path while skipping previously visited nodes. It is well-known that the total cost of the resulting path is no more than twice the cost of the tree [10]. Alternatively, one could use any sophisticated TSP algorithm on the nodes of each tree to obtain a good path for each robot. The specific method used does not affect the results in this paper as long as the total cost of each path is at most twice the cost of the corresponding tree, which can be guaranteed through shortcutting. Thus, in our auction framework and analysis, we assume that the final step of converting trees into paths incurs an approximation factor of at most 2 (in MINISUM cost).

VI. BIDDING RULES

In every round of the auction, the robots use a bidding rule to determine the appropriate (according to the team objective) bid for each target. We suggest a generic methodology for deriving such rules for any given team objective, and we derive six bidding rules for the three team objectives we consider. We divide the bidding rules into two classes depending on whether they aim to build paths or trees.

Suppose that the team objective is expressed as

$$\min_{\mathcal{A}} f\big(g(r_1, A_1), \ldots, g(r_n, A_n)\big)$$

where function g measures the performance of each robot, function f measures the performance of the team, and $\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$ is a partition of the set of targets, where targets in A_i are allocated to robot r_i. The three team objectives we consider fit this structure. Let $RPC(r_i, A_i)$ denote the minimum *robot path cost* for robot r_i to visit all targets in A_i from its current location. Similarly, let $CTPC(r_i, A_i)$ denote the minimum *cumulative target path cost* of all targets in A_i, again, if robot r_i visits all targets in A_i from its current location. Then, the three team objectives can be expressed as

$$\begin{aligned} \text{MINISUM} &: \min_{\mathcal{A}} \sum_j RPC(r_j, A_j), \\ \text{MINIMAX} &: \min_{\mathcal{A}} \max_j RPC(r_j, A_j), \\ \text{MINIAVE} &: \min_{\mathcal{A}} \frac{1}{m} \sum_j CTPC(r_j, A_j). \end{aligned}$$

Let $(S_1, S_2, \ldots, S_n)$ be the current partial allocation of targets to robots in some round of the auction, and let t be an unallocated target. We propose the following bidding rule, which is directly derived from the team objective.

> **Bidding Rule** Robot r bids on unallocated target t the difference in performance for the given team objective between the current allocation of targets to robots and the allocation that results from the current one if robot r is additionally allocated target t.

Consequently, robot r_i should bid on target t the difference

$$f\Big(g(r_1, S'_1), \ldots, g(r_n, S'_n)\Big) - f\Big(g(r_1, S_1), \ldots, g(r_n, S_n)\Big),$$

where $S'_i = S_i \cup \{t\}$ and $S'_j = S_j$ for $i \neq j$. This generic bidding rule thus performs some sort of hill climbing, aiming to find a good, but not necessarily optimal, allocation. Note that this generic bidding rule may require additional communication for computing the bids. However, for the objectives we consider, bid computation can be done locally.

For the MINISUM team objective, robot r_i bids on target t

$$\begin{aligned} &\sum_j RPC(r_j, S'_j) - \sum_j RPC(r_j, S_j) \\ &= RPC(r_i, S_i \cup \{t\}) - RPC(r_i, S_i). \end{aligned}$$

For the MINIMAX team objective, robot r_i bids on target t

$$\begin{aligned} &\max_j RPC(r_j, S'_j) - \max_j RPC(r_j, S_j) \\ &= RPC(r_i, S_i \cup \{t\}) - \max_j RPC(r_j, S_j). \end{aligned}$$

This derivation uses the fact that $\max_j RPC(r_j, S'_j) = RPC(r_i, S'_i)$; otherwise, target t would have already been allocated in a previous round of bidding. The term $\max_j RPC(r_j, S_j)$ can be dropped since the outcome of the auction remains unchanged if all bids change by a constant. Thus, robot r_i can just bid $RPC(r_i, S_i \cup \{t\})$ on target t. Last, for the MINIAVE team objective, robot r_i bids on target t

$$\begin{aligned} &\frac{1}{m}\sum_j CTPC(r_j, S'_j) - \frac{1}{m}\sum_j CTPC(r_j, S_j) \\ &= \frac{1}{m}\Big(CTPC(r_i, S_i \cup \{t\}) - CTPC(r_i, S_i)\Big). \end{aligned}$$

The factor $1/m$ can be dropped since the outcome of the auction remains unchanged if all bids are multiplied by a positive constant. Thus, robot r_i can bid just $CTPC(r_i, S_i \cup \{t\}) - CTPC(r_i, S_i)$ on target t.

Thus, the bidding rules for the three team objectives are

- BIDSUMPATH: $RPC(r_i, S_i \cup \{t\}) - RPC(r_i, S_i)$,
- BIDMAXPATH: $RPC(r_i, S_i \cup \{t\})$, and
- BIDAVEPATH: $CTPC(r_i, S_i \cup \{t\}) - CTPC(r_i, S_i)$.

The robots need to be able to calculate their bids efficiently, but computing RPC or $CTPC$ is NP-hard. Therefore, we assume that each robot r_i uses a heuristic method to approximate these functions. In particular, we make use of the insertion heuristic for TSP: given a path that visits the targets in S_i, evaluate all insertions of target t into all possible positions on the existing path, and choose the one that minimizes the cost of the new path. Our results are not affected if other methods are used, as long as the resulting bids are not worse than the bids computed using the insertion heuristic.

A similar analysis can be used to derive bidding rules for the case of constructing trees. For any robot r_i and any set of targets S_i, let $RTC(r_i, S_i)$ denote the minimum *robot tree cost*, that is, the cost of a minimum spanning tree over the nodes $\{r_i\} \cup S_i$. Similarly, let $CTTC(r_i, S_i)$ denote the minimum *cumulative target tree cost* which is the sum of root-target costs for all targets in S_i in a spanning tree over $\{r_i\} \cup S_i$ with root r_i. Without going through details, the bidding rules for the three team objectives in this case are

- BIDSUMTREE: $RTC(r_i, S_i \cup \{t\}) - RTC(r_i, S_i)$,
- BIDMAXTREE: $RTC(r_i, S_i \cup \{t\})$, and
- BIDAVETREE: $CTTC(r_i, S_i \cup \{t\}) - CTTC(r_i, S_i)$.

Given the sequential nature of allocation, the TREE bidding rules can be further simplified. In particular, a tree over S_i

remains unchanged within a tree over $S_i \cup \{t\}$ under any of the three objectives. This is true because target t was not allocated in earlier rounds (even though it was present and available), and hence it does not offer a better way to connect nodes in S_i. For the BIDSUMTREE and BIDMAXTREE rules, target t is connected to S_i through the cheapest edge, whereas for the BIDAVETREE rule, it is connected directly to the root r_i because of the triangle inequality assumption. In other words, $RTC(r_i, S_i \cup \{t\}) = RTC(r_i, S_i) + c(S_i \cup \{r_i\}, t)$, where $c(S_i \cup \{r_i\}, t)$ is the cost of the cheapest edge between any node in $S_i \cup \{r_i\}$ and t, and $CTTC(r_i, S_i \cup \{t\}) = CTTC(r_i, S_i) + c(r_i, t)$. Thus, the rules can be expressed as:

- BIDSUMTREE: $c(S_i \cup \{r_i\}, t)$,
- BIDMAXTREE: $RTC(r_i, S_i) + c(S_i \cup \{r_i\}, t)$, and
- BIDAVETREE: $c(r_i, t)$.

Bids for the TREE rules are computable in polynomial time.

VII. SUMMARY OF RESULTS

We assess the performance of each bidding rule theoretically in comparison to optimal performance and with respect to each of the three team objectives. This is done in terms of upper and lower bounds on the performance ratio (maximum ratio of rule performance over optimal performance).

If $I(n, m)$ is the class of all instances of multi-robot routing with n robots and m targets, an upper bound on the performance ratio for a rule R and an objective X is a function $\mathrm{UB}(n, m, R, X)$ such that for any n and m:

$$\max_{I \in I(n,m)} \frac{R(I, X)}{O(I, X)} \leq \mathrm{UB}(n, m, R, X) ,$$

where $R(I, X)$ is the cost of the solution under objective X for instance $I \in I(n, m)$ obtained using rule R and $O(I, X)$ is the optimal cost under objective X for instance I. The performance ratio is lower bounded by a function $\mathrm{LB}(n, m, R, X)$ if there exists some infinite family of instances $\mathcal{F}$ such that for each $I \in \mathcal{F}$:

$$\mathrm{LB}(n_I, m_I, R, X) \leq \frac{R(I, X)}{O(I, X)} ,$$

where n_I and m_I are the number of robots and targets in instance I. Therefore, the performance ratio cannot be less than $\mathrm{LB}(n, m, R, X)$. An upper bound provides a guarantee on the performance of the corresponding rule for the corresponding objective, whereas a lower bound usually represents pathological cases that demonstrate worst-case behavior.

It should be pointed out that each bidding rule essentially represents a family of rules. For the PATH rules, we do not specify a particular choice for the computation of the functions RPC and $CTPC$. This choice can be anything between computing them optimaly (NP-hard) and computing them approximately through the insertion heuristic (polynomial). However, we assume that whatever the choice, the approximation will not be worse than the insertion heuristic approximation. Similarly, for the TREE rules we do not specify a particular choice for the conversion of trees to paths, which can range from computing optimal paths (NP-hard) to using shortcutting (polynomial). Once again, we only assume that, whatever the choice, the cost of each path is at most twice the cost of the corresponding tree as guaranteed by shortcutting. Our bounds apply to the entire family of rules. In deriving the upper bounds, we do not assume any better heuristic than the insertion heuristic or shortcutting. In addition, the lower bounds hold even if RPC and $CTPC$, as well as the conversion of trees to paths, are computed optimally.

TABLE I

BOUNDS ON PERFORMANCE RATIO (RULE PERFORMANCE OVER OPTIMAL PERFORMANCE) WITH n ROBOTS AND m TARGETS.

Bidding Rule	**Team Objective**					
	MINISUM		MINIMAX		MINIAVE	
	Lower	Upper	Lower	Upper	Lower	Upper
BIDSUMPATH	1.5	2	n	$2n$	$\frac{m+1}{2}$	$2m$
BIDMAXPATH	n	$2n$	$\frac{n+1}{2}$	$2n$	$\Omega(m^{1/3})$	$2m$
BIDAVEPATH	m	$2m^2$	$\frac{n+1}{2}$	$2m^2n$	$\Omega(m^{1/3})$	$2m^2$
BIDSUMTREE	1.5	2	n	$2n$	$\frac{m+1}{2}$	$2m$
BIDMAXTREE	n	$2n$	$\frac{n+1}{2}$	$2n$	$\Omega(m^{1/3})$	$2m$
BIDAVETREE	m	$2m$	$\frac{n+1}{2}$	$2mn$	$\Omega(m^{1/3})$	$2m^2$

Table I summarizes our results. The PATH and the TREE rules offer almost identical guarantees, which implies that they are not fundamentally different from a theoretical point of view. In practice, the PATH rules yield somewhat better solutions, as they build paths directly, whereas the TREE rules are computationally more efficient, since bid computation for the TREE rules is much faster than for the PATH rules. Given that often $n \ll m$, it is clear that the best guarantees are offered for the MINISUM and the MINIMAX objectives, whereas there are only loose guarantees for the MINIAVE objective. Independently of the objective, the BIDSUMPATH and BIDSUMTREE rules provide uniformly the best guarantees. Overall, our results show that our auction-based methods constitute a principled, viable approach to multi-robot routing.

VIII. ANALYSIS

In this section we prove the bounds in Table I. We make the following notational conventions. The solution found by using any of the bidding rules is marked with the name of the rule, e.g. BIDSUMTREE. An optimal solution for each team objective is denoted by OPTSUM, OPTMAX, and OPTAVE, respectively, and the cost of a solution S according to each team objective by SUM(S), MAX(S), and AVE(S), respectively. With a slight abuse of notation, if F is a forest, we also use SUM(F) for the total cost of the forest, MAX(F) for the cost of the most expensive tree in the forest, and AVE(F) for the average of all root-target costs in the forest.

The following lemma on the relationship of the various objective functions is used repeatedly.

Lemma 1: Let F be a spanning forest rooted at the robots, that spans all targets in an instance of multi-robot routing with n robots and m targets. Then, it holds that

$$\text{AVE(F)} \leq \text{MAX(F)} \leq \text{SUM(F)} \leq n \ \text{MAX(F)},$$

$$\text{SUM(F)} \leq m \ \text{AVE(F)}.$$

Proof: The maximum root-target cost of any target can be at most equal to the cost of the most expensive tree in the forest. Therefore, the average of the root-target costs cannot be more than the cost of the most expensive tree in the forest. Furthermore, the cost of the most expensive tree in the forest cannot exceed the total cost of the forest. The total cost of the forest cannot exceed an n-multiple of the cost of the most expensive tree, since there are at most n trees in the forest. Finally, there are m targets in the forest and the contribution of each target to the total cost of the forest is no more than its root-target cost. Therefore, the total cost of the forest cannot exceed the sum of all root-target costs, which can be expressed as an m-multiple of the average root-target cost. ■

Note that Lemma 1 holds even if F is a collection of disjoint robot paths which span all targets.

A. Upper Bounds for BIDSUMPATH

Theorem 2: The performance ratio of the BIDSUMPATH bidding rule for the MINISUM team objective is at most 2.

Proof: Let $G = (R \cup T, c)$ be the weighted graph over all robot and target nodes. In each round k of the auction, $k = 0, \ldots, m-1$, let V_k be the set of robot nodes and allocated target nodes and $\bar{V}_k$ the set of unallocated target nodes. The sets V_k and $\bar{V}_k$ define a cut over G, and, obviously, $V_0 = R$, $\bar{V}_0 = T$, $V_m = R \cup T$, and $\bar{V}_m = \emptyset$. In each round k, BIDSUMPATH selects a target $t \in \bar{V}_k$ that can be added to one of the paths in V_k with the least additional cost. Let this cost be $b(V_k, \bar{V}_k)$, which is exactly the bid placed by the winning robot. Therefore, the SUM cost of the solution found by BIDSUMPATH at the end of the auction is:

$$\text{SUM}(\text{BIDSUMPATH}) = \sum_{k=0}^{m-1} b(V_k, \bar{V}_k)$$

Let $c(V_k, \bar{V}_k)$ be the cost of a cheapest edge across the cut $(V_k, \bar{V}_k)$. A target in $\bar{V}_k$ corresponding to a cheapest edge can be inserted into some path in V_k with a cost increase of at most $2c(V_k, \bar{V}_k)$ in SUM cost (because of the triangle inequality assumption). Since the BIDSUMPATH rule identifies an insertion with minimum increase in SUM cost, it must be the case that $b(V_k, \bar{V}_k) \leq 2c(V_k, \bar{V}_k)$. Hence,

$$\text{SUM}(\text{BIDSUMPATH}) \leq 2 \sum_{k=0}^{m-1} c(V_k, \bar{V}_k)$$

Consider another graph G' which is identical to G except that exactly m edges have their costs lowered. In particular, for every cut $(V_k, \bar{V}_k)$ the cost of a cheapest edge connecting t (the target selected by BIDSUMPATH) to V_k is lowered to $c(V_k, \bar{V}_k)$ (the cost of a cheapest edge across the cut) in G'. Clearly, those m edges in G' form an MSF in G' (it is equivalent to running Prim's algorithm starting with the robot nodes connected to each other with zero cost). An MSF in G cannot have less SUM cost than any MSF in G', since we have only lowered costs while constructing G' from G. Therefore, we obtain:

$$\sum_{k=0}^{m-1} c(V_k, \bar{V}_k) = \text{SUM}\big(\text{MSF}(G')\big) \leq \text{SUM}\big(\text{MSF}(G)\big).$$

which implies that

$$\text{SUM}(\text{BIDSUMPATH}) \leq 2\ \text{SUM}\big(\text{MSF}(G)\big). \qquad (1)$$

An optimal solution OPTSUM for the MINISUM team objective is also a spanning forest in G, therefore it is true that

$$\text{SUM}\big(\text{MSF(G)}\big) \leq \text{SUM}(\text{OPTSUM}).$$

Thus, we conclude that

$$\text{SUM}(\text{BIDSUMPATH}) \leq 2\ \text{SUM}(\text{OPTSUM}).$$

■

Using Equation (1), Lemma 1, and the fact that both OPTMAX and OPTAVE are spanning forests, we also conclude:

Corollary 1: The performance ratio of the BIDSUMPATH bidding rule for the MINIMAX team objective is at most $2n$.

Corollary 2: The performance ratio of the BIDSUMPATH bidding rule for the MINIAVE team objective is at most $2m$.

B. Upper Bounds for BIDMAXPATH

Theorem 3: The performance ratio of the BIDMAXPATH bidding rule for the MINISUM team objective is at most $2n$.

Proof: As in Theorem 2, consider the cuts $(V_k, \bar{V}_k)$ in each round k of the auction. Let $c(V_k, \bar{V}_k)$ be the cost of a cheapest edge across the cut $(V_k, \bar{V}_k)$ and P_i^k, $i = 1, \ldots, n$, be the robot paths in V_k. We establish by induction that in any round k, the SUM cost of each path P_i^k is bounded by:

$$\text{SUM}(P_i^k) \leq 2 \sum_{j=0}^{k-1} c(V_j, \bar{V}_j)$$

The base case is certainly true as $P_i^0 = \{r_i\}$ (a single node), $\text{SUM}(P_i^0) = 0$. Assume that the assertion holds for k. In the next round, BIDMAXPATH allocates a target $t \in \bar{V}_k$ that minimizes the cost of the most expensive path in V_{k+1}. The path P_r^{k+1} where t was added must be the most expensive path in V_{k+1}, otherwise t would have been allocated in some previous round. Therefore, for any path P_i^{k+1} in V_{k+1} it is true that

$$\text{SUM}(P_i^{k+1}) \leq \text{SUM}(P_r^{k+1}).$$

Let (t'', t') be the cheapest edge across the cut $(V_k, \bar{V}_k)$, with $t' \in \bar{V}_k$. Also, let $P_{r'}^k \subseteq V_k$ be the path containing t''. Target t' can be inserted into path $P_{r'}^k$ with an increase of at most $2c(V_k, \bar{V}_k)$ in the SUM cost of $P_{r'}^k$ (because of the triangle inequality assumption). Since BIDMAXPATH chose to insert t in P_r^k in round k, it must be the case that

$$\text{SUM}(P_r^{k+1}) \leq \text{SUM}(P_{r'}^k) + 2\ c(V_k, \bar{V}_k)$$

Finally, by the inductive hypothesis we have

$$\text{SUM}(P_i^{k+1}) \leq 2 \sum_{j=0}^{k-1} c(V_j, \bar{V}_j) + 2\ c(V_k, \bar{V}_k) = 2 \sum_{j=0}^{k} c(V_j, \bar{V}_j).$$

Since the MAX cost of the BIDMAXPATH solution is the SUM cost of the most expensive path, we conclude that

$$\text{MAX}(\text{BIDMAXPATH}) \leq 2 \sum_{j=0}^{m-1} c(V_j, \bar{V}_j).$$

Using the construction for graph G' as in Theorem 2, we have

$$\textsc{Max}(\textsc{BidMaxPath}) \leq 2\ \textsc{Sum}(\text{MSF}). \tag{2}$$

An optimal solution OptSum for the MiniSum team objective is also a spanning forest, so by Lemma 1 we have

$$\textsc{Sum}(\textsc{BidMaxPath}) \leq 2n\ \textsc{Sum}(\textsc{OptSum}).$$

■

Using Equation (2), Lemma 1, and the fact that both OptMax and OptAve are spanning forests, we also conclude:

Corollary 3: The performance ratio of the BidMaxPath bidding rule for the MiniMax team objective is at most $2n$.

Corollary 4: The performance ratio of the BidMaxPath bidding rule for the MiniAve team objective is at most $2m$.

C. Upper Bounds for BidAvePath

Theorem 4: The performance ratio of the BidAvePath bidding rule for the MiniSum team objective is at most $2m^2$.

Proof: As in Theorem 2, consider the cuts $(V_k, \bar{V}_k)$ in each round k of the auction. In each round k, BidAvePath selects a target $t \in \bar{V}_k$ that is added to one of the paths in V_k with the least increase in the Ave team objective. Let this increase be $b(V_k, \bar{V}_k)$ which corresponds to the bid placed by the winning robot. Therefore, the Ave cost of the solution found by BidAvePath at the end of the auction is:

$$\textsc{Ave}(\textsc{BidAvePath}) = \sum_{k=0}^{m-1} b(V_k, \bar{V}_k)$$

Let $c(R, \bar{V}_k)$ be the cost of a cheapest edge across the sets R and $\bar{V}_k$, that is, the cost of a cheapest edge between some unallocated target t' and some robot r'. Target t' can always be inserted as the first target in the path of r' in V_k. Such an insertion causes an increase of at most $2c(R, \bar{V}_k)$ in the robot-target cost of each target in the path because of the triangle inequality assumption, while the robot-target cost of t' is $c(R, \bar{V}_k)$. In the worst case, this insertion occurs at a path that contains all other targets. Since the increase of the robot-target cost for each of the m targets is at most $2c(R, \bar{V}_k)$, so is the increase in Ave cost. Since the BidAvePath rule identifies the insertion with the least increase in each round, it must be the case that $b(V_k, \bar{V}_k) \leq 2c(R, \bar{V}_k)$, and therefore

$$\textsc{Ave}(\textsc{BidAvePath}) \leq 2 \sum_{k=0}^{m-1} c(R, \bar{V}_k).$$

It holds that $c(R, \bar{V}_k) \leq \textsc{Sum}(\text{MSF})$, since in an MSF, no robot can reach a target with cost less than the cheapest direct edge from any robot to that target. Therefore,

$$\textsc{Ave}(\textsc{BidAvePath}) \leq 2m\ \textsc{Sum}(\text{MSF}). \tag{3}$$

An optimal solution OptSum for the MiniSum team objective is also a spanning forest, so by Lemma 1 we have that

$$\textsc{Sum}(\textsc{BidAvePath}) \leq 2m^2\ \textsc{Sum}(\textsc{OptSum}).$$

■

Using Equation (3), Lemma 1, and the fact that both OptMax and OptAve are spanning forests, we also conclude:

Corollary 5: The performance ratio of the BidAvePath bidding rule for the MiniMax team objective is at most $2m^2n$.

Corollary 6: The performance ratio of the BidAvePath bidding rule for the MiniAve team objective is at most $2m^2$.

D. Upper Bounds for BidSumTree

Theorem 5: The performance ratio of the BidSumTree bidding rule for the MiniSum team objective is at most 2 [6].

Proof: The bid placed by each robot in each round is equal to the cost of adding the closest unallocated target to its own subtree. Considering all robot nodes as connected with each other with zero cost, the auction with the BidSumTree rule is identical to Prim's algorithm for MST [8]. Therefore, the tree found by this rule is indeed an MSF. Converting the trees of an MSF to paths incurs a factor of 2, therefore:

$$\textsc{Sum}(\textsc{BidSumTree}) \leq 2\ \textsc{Sum}(\text{MSF}). \tag{4}$$

An optimal solution OptSum for the MiniSum team objective is also a spanning forest, therefore we conclude that

$$\textsc{Sum}(\textsc{BidSumTree}) \leq 2\ \textsc{Sum}(\textsc{OptSum}).$$

■

Using Equation (4), Lemma 1, and the fact that both OptMax and OptAve are spanning forests, we also conclude:

Corollary 7: The performance ratio of the BidSumTree bidding rule for the MiniMax team objective is at most $2n$.

Corollary 8: The performance ratio of the BidSumTree bidding rule for the MiniAve team objective is at most $2m$.

E. Upper Bounds for BidMaxTree

Theorem 6: The performance ratio of the BidMaxTree bidding rule for the MiniSum team objective is at most $2n$.

Proof: As in Theorem 2, consider the cuts $(V_k, \bar{V}_k)$ in each round k of the auction. Let $c(V_k, \bar{V}_k)$ be the cost of the cheapest edge across the cut $(V_k, \bar{V}_k)$. We establish by induction that in any round of the auction, the Sum cost of any tree T_i^k, $i = 1, \ldots, n$, in V_k is bounded as follows:

$$\textsc{Sum}(T_i^k) \leq \sum_{j=0}^{k-1} c(V_j, \bar{V}_j)$$

The base case is certainly true as $T_i^0 = \{r_i\}$ (a single node), $\textsc{Sum}(T_i^0) = 0$. Assume that the assertion holds for k. In the next round, BidMaxTree allocates a target $t \in \bar{V}_k$ that minimizes the cost of the most expensive tree in V_{k+1}. The tree T_r^{k+1} where t was added must be the most expensive tree in V_{k+1}; otherwise, t would have been allocated in some previous round. Therefore, for any tree T_i^{k+1} in V_{k+1} it is true that

$$\textsc{Sum}(T_i^{k+1}) \leq \textsc{Sum}(T_r^k) + c(T_r^k, t)$$

Let (t'', t') be the cheapest edge across the cut $(V_k, \bar{V}_k)$, with $t' \in \bar{V}_k$. Also, let $T_{r'}^k \subseteq V_k$ be the tree containing t''. Since

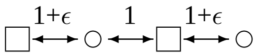

Fig. 1. A simple instance with 2 robots (squares) and two targets (circles).

BidMaxTree chose to attach t to T_r^k in round k, it must be the case that

$$\textsc{Sum}(T_r^k) + c(T_r^k, t) \le \textsc{Sum}(T_{r'}^k) + c(T_{r'}^k, t') \ ,$$

or, using the fact that $c(T_{r'}^k, t') = c(V_k, \bar{V}_k)$:

$$\textsc{Sum}(T_i^{k+1}) \le \textsc{Sum}(T_{r'}^k) + c(V_k, \bar{V}_k).$$

Finally, by the inductive hypothesis

$$\textsc{Sum}(T_i^{k+1}) \le \sum_{j=0}^{k-1} c(V_j, \bar{V}_j) + c(V_k, \bar{V}_k) = \sum_{j=0}^{k} c(V_j, \bar{V}_j).$$

Since the Max cost of the BidMaxTree solution is at most twice the Sum cost of the most expensive tree (taking into account the conversion of trees to paths), we conclude that

$$\textsc{Max}(\textsc{BidMaxTree}) \le 2 \sum_{j=0}^{m-1} c(V_j, \bar{V}_j).$$

Using the construction for graph G' as in Theorem 2, we have

$$\textsc{Max}(\textsc{BidMaxTree}) \le 2 \ \textsc{Sum}(\text{MSF}), \qquad (5)$$

An optimal solution OptSum for the MiniSum team objective is also a spanning forest, so by Lemma 1 we have

$$\textsc{Sum}(\textsc{BidMaxTree}) \le 2n \ \textsc{Sum}(\textsc{OptSum}).$$

■

Using Equation (5), Lemma 1, and the fact that both OptMax and OptAve are spanning forests, we also conclude:

Corollary 9: The performance ratio of the BidMaxTree bidding rule for the MiniMax team objective is at most $2n$.

Corollary 10: The performance ratio of the BidMaxTree bidding rule for the MiniAve team objective is at most $2m$.

F. Upper Bounds for BidAveTree

Theorem 7: The performance ratio of the BidAveTree bidding rule for the MiniSum team objective is at most $2m$.

Proof: Under the BidAveTree rule, the intermediate spanning forest at the end of the auction will consist of stars, one for each robot, where each target is connected directly to the closest (in terms of cost) robot. Because of the triangle inequality assumption, direct connections minimize the robot-target costs, and therefore the average.

Let c^* be the cost of the most expensive robot-target edge in the forest. It holds that $c^* \le \textsc{Sum}(\text{MSF})$, since in an MSF, no robot can reach a target with cost less than the cheapest direct edge from any robot to that target. Since there are m targets in total, the Sum cost of the forest is at most mc^*, and thus, the Sum cost of BidAveTree can be at most $2mc^*$,

$$\textsc{Sum}(\textsc{BidAveTree}) \le 2mc^* \le 2m \ \textsc{Sum}(\text{MSF}), \qquad (6)$$

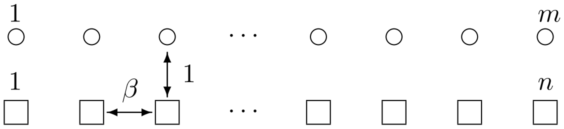

Fig. 2. Parallel lines construction: n robots and $m = n$ targets spaced evenly (distance of β) on two parallel lines (one for robots, one for targets). The distance between the two lines is 1, except for the left-most robot which is a little closer to its corresponding target.

where the factor of 2 comes from the conversion of stars to paths. Thus, given that any optimal solution OptSum is also a spanning forest, we conclude that

$$\textsc{Sum}(\textsc{BidAveTree}) \le 2m \ \textsc{Sum}(\textsc{OptSum}).$$

■

Using Equation (6), Lemma 1, and the fact that both OptMax and OptAve are spanning forests, we also conclude:

Corollary 11: The performance ratio of the BidAveTree bidding rule for the MiniMax team objective is at most $2mn$.

Corollary 12: The performance ratio of the BidAveTree bidding rule for the MiniAve team objective is at most $2m^2$.

G. Lower bounds

In all example instances, robots are shown as squares, single targets as open circles, clusters of targets as solid circles, and ϵ represents an arbitrarily small positive number. The examples that yield lower bounds for the Tree rules are identical to those for the Path rules, and therefore, they are omitted.

BidSumPath applied to the instance in Figure 1 allocates both targets to the robot on the right with a Sum cost of $3+\epsilon \approx 3$, as opposed to the OptSum cost of $2+2\epsilon \approx 2$ (one target to each robot). By replicating the construction, we obtain an infinite family of such instances, thus a lower bound for the performance ratio of BidSumPath for MiniSum is 1.5.

Applying the BidSumPath rule to the example in Figure 2 with $\beta = 1 - \epsilon$ yields a solution that allocates all targets to the left-most robot, and a path that runs through all targets left to right. Obviously, $\textsc{Max}(\textsc{BidSumPath}) \approx n$, whereas $\textsc{Max}(\textsc{OptMax}) = 1$ (each robot visits its corresponding target). Therefore, a lower bound for the performance ratio of BidSumPath for MiniMax is $n = m$. Similarly, $\textsc{Ave}(\textsc{BidSumPath}) \approx (1 + 2 + \ldots + n)/n = (n+1)/2$, whereas $\textsc{Ave}(\textsc{OptAve}) = 1$ (each robot visits its corresponding target). Therefore, a lower bound for the performance ratio of BidSumPath for MiniAve is $(n+1)/2 = (m+1)/2$.

Applying BidMaxPath to the instance in Figure 2 with $\beta = \epsilon$ yields a solution that allocates one target to each robot. Thus, $\textsc{Sum}(\textsc{BidMaxPath}) = n$, whereas $\textsc{Sum}(\textsc{OptSum}) = 1 + n\epsilon \ \approx \ 1$ (the left most robot takes all targets) for $\epsilon \ = \ o(1/n)$. Thus, a lower bound for the performance ratio of BidMaxPath for MiniSum is $n = m$. Similarly, $\textsc{Sum}(\textsc{BidAvePath}) = n$, whereas $\textsc{Sum}(\textsc{OptSum}) \approx 1$ (the left-most robot takes all targets). Thus, a lower bound for the performance ratio of BidAvePath for MiniSum is $n = m$.

Applying BidMaxPath (or BidAvePath) to the instance in Figure 3 yields a solution that allocates one target per cluster to each robot. The path of each robot traverses the grid

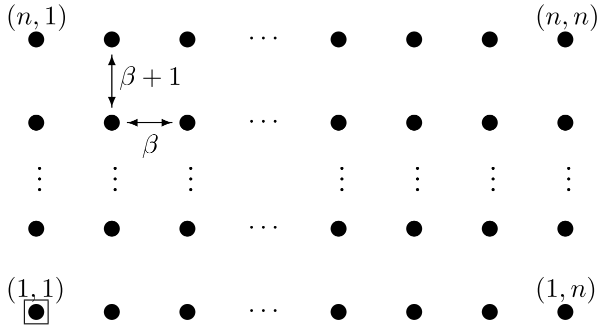

Fig. 3. Grid construction: n robots at $(1,1)$ and $m = n^3$ targets on a $(n \times n)$ rectangular grid; each of the n^2 gridpoints is a cluster of n targets. The intra-row distance is β, whereas the intra-column distance is $\beta+1$.

alternating left-to-right and right-to-left, and also from bottom to top. Figure 4 shows the necessary adjustments to make this possible. Thus, Max(BidMaxPath) $= n(n-1)\beta + (\beta+1)(n-1) = (n\beta+\beta+1)(n-1)$, whereas Max(OptMax) $\leq (n-1)\beta+(\beta+1)(n-1) = (2\beta+1)(n-1)$ (each robot takes one row of the grid). Thus, a lower bound for the performance ratio of BidMaxPath for MiniMax is $(n+1)/2 = (m^{1/3}+1)/2$ for large β. In addition, without going into a detailed analysis, it is obvious that Ave(BidMaxPath) $= \Omega(n^2)$, whereas Ave(OptAve) $= O(n)$ (each robot takes one row of the grid). Thus, a lower bound for the performance ratio of BidMaxPath for MiniAve is $\Omega(n)$ or $\Omega(m^{1/3})$ since $m = n^3$. The same bounds hold for BidAvePath.

IX. Related Work

Multi-robot routing falls into the class of Location Routing problems [11]. There has been a tremendous amount of work on centralized algorithms for solving such problems optimally or approximately. The MiniSum objective has been studied in the context of k-TSP problems and can be approximated by a constant factor [12], [13]. The MiniMax objective appears also in the Nurse Location Problem for which there exists an 8-approximation [9]. This objective has also been studied in the context of job scheduling on unrelated parallel machines (makespan) [14]. Finally, the MiniAve objective (also known as the Traveling Repairman Problem) is approximable to a constant factor [15]. Robotics researchers have studied the MiniSum [2], [5]–[7] objective extensively, but only occasionally the MiniMax [1], [7] and MiniAve [7] objectives.

A variety of auction methods have been used for multi-robot routing. Berhault et. al. [5] have used combinatorial auctions; however, the complexity of these auctions makes them impractical for large problems. Dias and Stentz [2] have proposed a single-item auction similar to BidSumPath which has been implemented on real robots for exploration tasks. Our Path rules have been tested experimentally [7], and have been shown to perform best for their corresponding team objective. Also, their actual performance is well below the theoretical upper bounds.

X. Conclusion

We presented a theoretical analysis of auction-based methods for multi-robot routing and established for the first time performance guarantees. We are currently conducting extensive experimentation to assess the actual performance of our bidding rules and their effectiveness in solving large problems.

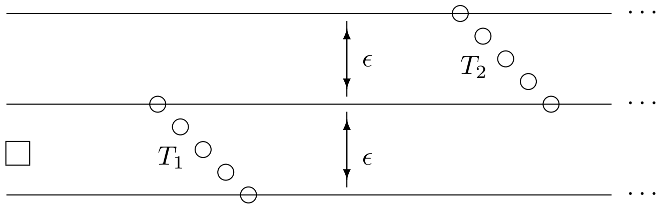

Fig. 4. Cluster arrangement in grid construction: example with 2 clusters of 5 targets each and 5 robots clustered on the left. Targets in cluster T_1 are arranged evenly on a line of slope -45° within a small cost from each other. Each of the 5 robots visits one target in T_1, since the straight line is the shortest path. Targets in T_2 have the same arrangement, but are shifted up by ϵ to ensure that the cost between corresponding targets in T_1 and T_2 is less than any other inter-cluster cost. The robot that visits the first target in T_1 will also visit the first target in T_2, and so on. This pattern repeats along the horizontal axis, but it can also be used for clusters arranged vertically.

Acknowledgment

This research was partially supported by NSF awards under contracts DMI-0113881, IIS-0413196, and IIS-0412912.

References

[1] G. Rabideau, T. Estlin, S. Chien, and A. Barrett, "A comparison of coordinated planning methods for cooperating rovers," in *Proceedings of the Intl Conference on Autonomous Agents*, 2000, pp. 100–101.

[2] M. Dias and A. Stentz, "A free market architecture for distributed control of a multirobot system," in *Proceedings of the International Conference on Intelligent Autonomous Systems*, 2000, pp. 115–122.

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

[4] R. Zlot, A. Stentz, M. Dias, and S. Thayer, "Multi-robot exploration controlled by a market economy," in *Proceedings of the International Conference on Robotics and Automation*, 2002, pp. 3016–3023.

[5] M. Berhault, H. Huang, P. Keskinocak, S. Koenig, W. Elmaghraby, P. Griffin, and A. Kleywegt, "Robot exploration with combinatorial auctions," in *Proceedings of the International Conference on Intelligent Robots and Systems*, October 2003, pp. 1957–1962.

[6] M. Lagoudakis, M. Berhault, P. Keskinocak, S. Koenig, and A. Kleywegt, "Simple auctions with performance guarantees for multi-robot task allocation," in *Proceedings of the International Conference on Intelligent Robots and Systems*, September 2004.

[7] C. Tovey, M. Lagoudakis, S. Jain, and S. Koenig, "The generation of bidding rules for auction-based robot coordination," in *Proceedings of the 3rd International Multi-Robot Systems Workshop*, March 2005.

[8] R. C. Prim, "Shortest connection networks and some generalizations." *Bell Systems Technical Journal*, pp. 1389–1401, 1957.

[9] G. Even, N. Garg, J. Könemann, R. Ravi, and A. Sinha, "Min-Max tree covers of graphs," *Operations Research Letters*, vol. 32, pp. 309–315, 2004.

[10] E. Lawler, J. Lenstra, A. Rinnooy Kan, and D. Shmoys, Eds., *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Wiley, 1985.

[11] P. Toth and D. Vigo, Eds., *The Vehicle Routing Problem*, ser. Monographs on Discrete Mathematics and Applications. SIAM, 2002.

[12] G. N. Frederickson, M. S. Hecht, and C. E. Kim, "Approximation algorithms for some routing problems," *SIAM Journal of Computing*, vol. 7, no. 2, pp. 178–193, 1978.

[13] J. O. Cerdeira, "The multi-depot traveling salesman problem," *Revista Investigao Operacional*, vol. 12, 1992.

[14] J. K. Lenstra, D. B. Shmoys, and E. Tardos, "Approximation algorithms for scheduling unrelated parallel machines," *Math. Program.*, vol. 46, no. 3, pp. 259–271, 1990.

[15] J. Fakcharoenphol, C. Harrelson, and S. Rao, "The k-traveling repairman problem," in *SODA '03: Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2003, pp. 655–664.