Target assignment for robotic networks: asymptotic performance under limited communication

Stephen L. Smith   Francesco Bullo

***Abstract*— We are given an equal number of mobile robotic agents, and distinct target locations. Each agent has simple integrator dynamics, a limited communication range, and knowledge of the position of every target. We address the problem of designing a distributed algorithm that allows the group of agents to divide the targets among themselves and, simultaneously, leads each agent to reach its unique target. We do not require connectivity of the communication graph at any time. We introduce a novel assignment-based algorithm with the following features: initial assignments and robot motions follow a greedy rule, and distributed refinements of the assignment exploit an implicit circular ordering of the targets. We prove correctness of the algorithm, and give worst-case asymptotic bounds on the time to complete the assignment as the environment grows with the number of agents. We show that among a certain class of distributed algorithms, our algorithm is asymptotically optimal. The analysis utilizes results on the Euclidean traveling salesperson problem.**

I. INTRODUCTION

Consider a group of n mobile robotic agents and n target locations, all lying in $\mathbb{R}^d$, $d \geq 1$. Each agent has a limited communication range, and knows the location of some subset (possibly all) of the n targets through GPS coordinates or a map of the environment. The *target assignment problem* we consider is to design a distributed algorithm that allows the group of agents to efficiently divide the n targets among themselves and, simultaneously, that leads each agent to reach its unique target. Such a problem could arise in several applications. For example, one could think of the agents as UAV's on a surveillance mission, and the targets as the centers of their desired loitering patterns. Or in the context of formation control, the target positions could describe the desired formation for a group of robots.

The first question is; how would we divide the targets among the agents in a centralized fashion? A reasonable strategy would be to minimize the sum of the distances traveled by each agent to arrive at its target. The problem of optimally dividing n persons among n objects, subject to a linear cost function, is a problem in combinatorial optimization [1]. When the cost function is the sum, the problem is referred to as the *assignment problem*, or the *minimum weight perfect matching problem in bipartite graphs*. The assignment problem can be written as an integer linear program. Unlike some integer linear programs, such as the Euclidean traveling salesperson problem (ETSP), optimal solutions for the assignment problem can be computed in polynomial time. In 1955 Kuhn [2] developed the Hungarian method—the first polynomial time method for solving the assignment problem. Kuhn's method solves the problem in $O(n^3)$ computation time (see Section II for a definition of the O notation).

Another approach to the assignment problem is the *auction algorithm* [3], [4], [5], first proposed by Bertsekas. This method solves the problem in $O(n^3)$ computation time, but can be computed in a parallel fashion, with one processor for each person. Recently, Moore and Passino [6] modified the auction algorithm to assign mobile robots to spatially distributed tasks in the presence of communication delays. However, in order to exchange bids on a particular object (task), the auction algorithm, and thus the work in [6], requires that the communication graph between processors (robots) is complete.

In this paper we address the target assignment problem when each agent has knowledge of all target positions, and a limited communication range $r > 0$. We introduce a class of distributed algorithms, called *assignment-based motion*, which provide a natural approach to the problem. Following the recent interest in determining the time complexity of distributed algorithms for robotic networks (for example, see [7] and [8]) we study the worst-case asymptotic performance of the assignment-based motion class as the environment grows with n. We show that for a d-dimensional cube environment, $[0, \ell(n)]^d$, $d \geq 1$, if the side length $\ell(n)$ grows at a rate of at least $(1+\epsilon)rn^{1/d}$, where $\epsilon > 0$, then the completion time is in $\Omega(n^{(d-1)/d}\ell(n))$, for all algorithms in this class.

In Section V we introduce a novel control and communication algorithm, called ETSP ASSGMT, within the assignment-based motion class. In this algorithm, each agent computes an ETSP tour through the n targets, turning the cloud of target points into an ordered ring. Agents then move along the ring, looking for the next available target. When agents communicate, they exchange messages of $O(\log n)$ size, containing information on the location of the next available target along the ring. In Section V-A, we verify the correctness of this algorithm for any communication graph which contains, as a subgraph, the r-disk graph. In Section V-B, we show that when $\ell(n) \geq (1+\epsilon)rn^{1/d}$, for some $\epsilon > 0$, among all algorithms in the assignment-based motion class, the ETSP ASSGMT algorithm is asymptotically optimal (i.e., a constant factor approximation of the optimal). Finally, in Section V-D, we note that the ETSP ASSGMT algorithm solves the target assignment problem in the case when there are n agents and m targets, with $n \neq m$. Due to space constraints, all proofs have been omitted and can be found in [9].

II. Background

In this section we introduce notation and review some relevant results in combinatorial optimization.

A. Notation

We let $\mathbb{R}$ denote the set of real numbers, $\mathbb{R}_{>0}$ denote the set of positive real numbers, and $\mathbb{N}$ denote the set of positive integers. For a set finite A we let $|A|$ denote its cardinality. For two functions $f, g : \mathbb{N} \to \mathbb{R}_{>0}$, we write $f(n) \in O(g)$ (respectively, $f(n) \in \Omega(g)$) if there exist $N \in \mathbb{N}$ and $c \in \mathbb{R}_{>0}$ such that $f(n) \leq cg(n)$ for all $n \geq N$ (respectively, $f(n) \geq cg(n)$ for all $n \geq N$). If $f(n) \in O(g)$ and $f(n) \in \Omega(g)$ we say $f(n) \in \Theta(g)$. Finally, we use the notation $\pmod{n}$ to denote arithmetic performed modulo $n \in \mathbb{N}$. Thus, for an integer $n \in \mathbb{N}$ we have $n+1 = 1 \pmod{n}$ and $0 = n \pmod{n}$, and $\{n-1, n, n+1\} = \{n-1, n, 1\} \pmod{n}$.

B. The assignment problem

Following [4], the classical assignment problem can be described as follows. Consider n persons who wish to divide themselves among n objects. For each person i, there is a nonempty set $\mathcal{Q}^{[i]}$ of objects that i can be assigned to, and cost $c_{ij} \geq 0$ associated to each object $j \in \mathcal{Q}^{[i]}$. An *assignment* S is a set of person-object pairs (i, j) such that $j \in \mathcal{Q}^{[i]}$ for all $(i, j) \in S$. For each person i (likewise, object j), there is at most one pair $(i, j) \in S$. We call the assignment *complete* if it contains n pairs. The goal is to find the complete assignment which minimizes $\sum_{(i,j)\in S} c_{ij}$.

Let x_{ij} be a set of variables for i and j in $\mathcal{I} := \{1, \dots, n\}$. For an assignment S, we write $x_{ij} = 1$ if $(i, j) \in S$, and $x_{ij} = 0$ otherwise. Thus, the problem of determining the optimal assignment can be written as a linear program:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n} \sum_{j \in \mathcal{Q}^{[i]}} c_{ij} x_{ij}, \\ \text{subject to} \quad & \sum_{j \in \mathcal{Q}^{[i]}} x_{ij} = 1 \quad \forall\, i \in \mathcal{I}, \\ & \sum_{\{i | j \in \mathcal{Q}^{[i]}\}} x_{ij} = 1 \quad \forall\, j \in \mathcal{I}, \\ & x_{ij} \geq 0. \end{aligned}$$

We cannot use linear inequalities to write the constraint that x_{ij}'s attain only the values zero and one. However, it turns out, [4], that there always exists an optimal solution in which the x_{ij}'s satisfy our integer assumption.

C. The Euclidean traveling salesperson problem

Let $\mathcal{Q}$ be a set of n points in a compact environment $\mathcal{E} \subset \mathbb{R}^d$, $d \geq 1$, and let $\mathcal{Q}_n$ be the set of all point sets $\mathcal{Q} \subset \mathcal{E}$ with $|\mathcal{Q}| = n$. Let $\mathrm{ETSP}(\mathcal{Q})$ denote the cost of the ETSP tour over the point set $\mathcal{Q}$, i.e., the length of the shortest closed path through all points in $\mathcal{Q}$. An important result, from [10], is that given a compact set $\mathcal{E}$, there exists a finite constant $\alpha(\mathcal{E})$ such that, for all $\mathcal{Q} \in \mathcal{Q}_n$,

$$\mathrm{ETSP}(\mathcal{Q}) \leq \alpha(\mathcal{E}) n^{(d-1)/d}. \tag{1}$$

In fact, we have that in the worst-case setting, the $\mathrm{ETSP}(\mathcal{Q})$ belongs to $\Theta(n^{(d-1)/d})$.

In our application of these results it will be useful to consider the case where the environment grows with the number of points. That is, we are interested in environments which are cubes, $[0, \ell(n)]^d$, $d \geq 1$, where $\ell(n)$ is the side length of the cube. Applying a simple scaling argument to the result in (1), we arrive at the following corollary.

Corollary 2.1 (ETSP tour length): Consider an environment $\mathcal{E} = [0, \ell(n)]^d$, where $d \geq 1$. For every point set $\mathcal{Q} \in \mathcal{Q}_n$, we have $\mathrm{ETSP}(\mathcal{Q}) \in O(n^{(d-1)/d} \ell(n))$.

The problem of computing an optimal tour is known to be NP-complete. However, there exist heuristics which can be computed efficiently and give a constant factor approximation to the optimal tour. The best known approximation algorithm is due to Christofides [11]. The *Christofides' algorithm* computes a tour that is no more than $3/2$ times longer than the optimal. It runs in time $O(n^3)$. Another method, known as the *double-tree algorithm*, produces tours that are no longer than twice the optimal, in run time $O(n^2)$.

III. Problem formulation

To describe the target assignment problem formally, consider n agents in an environment $\mathcal{E}(n) \subset \mathbb{R}^d$, $d \geq 1$. The environment $\mathcal{E}(n)$ is compact for each n but may grow with the number of agents. For ease of presentation let $\mathcal{E} := [0, \ell(n)]^d$, where $\ell(n) > 0$ (that is, $\mathcal{E}$ is a d-dimensional cube with side length $\ell(n)$). Each agent has a unique identifier (UID) taken from the set $I_{UID} \subseteq \mathbb{N}$. For simplicity, we assume that $I_{UID} := \mathcal{I} = \{1, \dots, n\}$. However, each agent does not know the set of UIDs being used (i.e., agent n does not know it has the largest UID). Agent $i \in \mathcal{I}$ has position $\mathbf{p}^{[i]} \in \mathcal{E}$. Two agents, i and k in $\mathcal{I}$, are able to communicate if and only if $\|\mathbf{p}^{[i]} - \mathbf{p}^{[k]}\| \leq r$, where $r > 0$ is called the communication range. We refer to the graph representing the communication links as the r-disk graph. Agent i's kinematic model is $\dot{\mathbf{p}}^{[i]} = \mathbf{u}^{[i]}$, where $\mathbf{u}^{[i]}$ is a velocity control input bounded by $v > 0$. We assume that the agents move in continuous time and communicate according to a discrete time communication schedule consisting of an increasing sequence of time instants with no accumulation points, $\{t_k\}_{k\in\mathbb{N}}$. We assume that $|t_{k+1} - t_k| \leq t_{max}$, for all $k \in \mathbb{N}$, where $t_{max} \in \mathbb{R}_{>0}$. At each communication round, agents can exchange messages of length $O(\log n)$. [1] We assume that communication round k occurs at time t_k, and that all messages are sent and received instantaneously at t_k. Motion then occurs from t_k until t_{k+1}. It should be noted that in this setup we are emphasizing the time complexity due to the motion of the agents.

Let $\mathcal{Q} := \{\mathbf{q}_1, \dots, \mathbf{q}_n\}$ be a set of distinct target locations, $\mathbf{q}_j \in \mathcal{E}$ for each $j \in \mathcal{I}$. Agent i is equipped with memory $M^{[i]}$, of size $|M^{[i]}|$. In this memory, agent i stores a set of target positions, $\mathcal{Q}^{[i]} \subseteq \mathcal{Q}$. These are the targets to which agent i can be assigned. We let $\mathcal{Q}^{[i]}(0)$ denote agent i's initial

[1] The number of bits required to represent an ID, unique among n agents, grows with the logarithm of n.

target set. In this paper we assume that each agent knows the position of every target. That is, $\mathcal{Q}^{[i]}(0) = \mathcal{Q}$ for each $i \in \mathcal{I}$. We refer to this as the *full knowledge* assumption. To store this amount of information we must assume that the size of each agents' memory, $|M^{[i]}|$, grows linearly with n. Our goal is to solve the *full knowledge target assignment problem*:

> Determine an algorithm for $n \in \mathbb{N}$ agents, with attributes as described above, satisfying the following requirement. There exists a time $T > 0$ such that for every agent $i \in \mathcal{I}$, there is a unique target $\mathbf{q}_{j_i} \in \mathcal{Q}^{[i]}(0)$ with $\mathbf{p}^{[i]}(t) = \mathbf{q}_{j_i}$ for all time $t \geq T$, where $j_i = j_k$ if and only if $i = k$.

In the remainder of the paper, we will refer to this as the *target assignment problem.*

Remark 3.1 (Consistent knowledge): A more general assumption on the initial target sets, $\mathcal{Q}^{[i]}(0)$, which still ensures the existence of a complete assignment, is the *consistent knowledge* assumption: For each $\mathcal{K} \subseteq \mathcal{I}$, $\left|\cup_{k\in\mathcal{K}} \mathcal{Q}^{[k]}(0)\right| \geq |\mathcal{K}|$. In fact, it was proved by Frobenius, 1917, and Hall, 1935 that this is the necessary and sufficient condition for the existence of a complete assignment [1]. •

In the full knowledge assumption, each agent knows the position of all targets in $\mathcal{Q}$. These positions will be stored in an array within each agents memory, rather than as an unordered set. To represent this, we replace the target set $\mathcal{Q}$ with the target n-tuple $\mathbf{q} := (\mathbf{q}_1, \ldots, \mathbf{q}_n)$, and the local target set $\mathcal{Q}^{[i]}$ with the n-tuple $\mathbf{q}^{[i]}$. Thus, in the full knowledge assumption, $\mathbf{q}^{[i]}(0) := \mathbf{q}$ for each $i \in \mathcal{I}$. (It is possible that the order of the targets in the local sets $\mathbf{q}^{[i]}$ may initially be different. However, given a set of distinct points in $\mathbb{R}^d$, it is always possible to create a unique ordering.)

IV. Assignment-based algorithms with lower bound analysis

In this section we introduce and analyze a class of deterministic algorithms for the target assignment problem.

A. The assignment-based motion class

The initialization, motion, and communication for each algorithm in the *assignment-based motion* class have the following attributes:

Initialization: In this class of algorithms agent i initially selects the closest target in $\mathbf{q}^{[i]}$, and sets the variable $\mathrm{curr}^{[i]}$ (agent i's current target), to the index of that target.

Motion: Agent i moves toward the target $\mathrm{curr}^{[i]}$ at constant speed $v > 0$:

$$\dot{\mathbf{p}}^{[i]} = \begin{cases} v\frac{\mathbf{q}^{[i]}_{\mathrm{curr}^{[i]}} - \mathbf{p}^{[i]}}{\|\mathbf{q}^{[i]}_{\mathrm{curr}^{[i]}} - \mathbf{p}^{[i]}\|}, & \text{if } \mathbf{q}^{[i]}_{\mathrm{curr}^{[i]}} \neq \mathbf{p}^{[i]}, \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

Communication: If agent i communicates with an agent k that is moving toward $\mathrm{curr}^{[k]} = \mathrm{curr}^{[i]}$, and if agent k is closer to $\mathrm{curr}^{[i]}$ than agent i, then agent i "removes" $\mathrm{curr}^{[i]}$ from $\mathbf{q}^{[i]}$ and selects a new target. The communication is described in more detail in the following.

Communication round for agent i.

Broadcast a message, $\mathrm{msg}^{[i]}$, based on $\mathbf{q}^{[i]}$ and containing $\mathrm{curr}^{[i]}$ and the UID i.
Receive $\mathrm{msg}^{[k]}$ from each agent k within communication range.
for all $\mathrm{msg}^{[k]}$ received **do**
    Based on $\mathrm{msg}^{[k]}$, (possibly) remove assigned targets from $\mathbf{q}^{[i]}$.
    if $\mathrm{curr}^{[i]} = \mathrm{curr}^{[k]}$ **then**
        If agent i is farther from $\mathrm{curr}^{[i]}$ than agent k, or if they are the same distance but $i < k$, remove $\mathrm{curr}^{[i]}$ from $\mathbf{q}^{[i]}$.
Set $\mathrm{curr}^{[i]}$ to a target in $\mathbf{q}^{[i]}$.

B. Lower bound on task complexity

In order to classify the time complexity (i.e., the completion time) of the assignment-based motion class of algorithms in solving the target assignment problem, we introduce a few useful definitions. We say that agent $i \in \mathcal{I}$ is *assigned* to target $\mathbf{q}^{[i]}_j$, $j \in \mathcal{I}$, when $\mathrm{curr}^{[i]} = j$. In this case, we also say target j is assigned to agent i. We say that agent $i \in \mathcal{I}$ *enters a conflict* over the target $\mathrm{curr}^{[i]}$, when agent i receives a message, $\mathrm{msg}^{[k]}$, with $\mathrm{curr}^{[i]} = \mathrm{curr}^{[k]}$. Agent i *loses the conflict* if agent i is farther from $\mathrm{curr}^{[i]}$ than agent k, and *wins the conflict* if agent i is closer to $\mathrm{curr}^{[i]}$ than agent k, where ties are broken by comparing UIDs.

Now we show that if agent i is assigned to the same target as another agent, it will enter a conflict in finite time.

Lemma 4.1 (Conflict in finite time): Consider any communication range $r > 0$, and any fixed number of agents $n \in \mathbb{N}$. If, for two agents i and k, $\mathrm{curr}^{[i]} = \mathrm{curr}^{[k]}$ at some time $t_1 \geq 0$, then agent i (and likewise, agent k) will enter a conflict over $\mathrm{curr}^{[i]}$ in finite time.

With these definitions we give a lower bound on the time complexity of the target assignment problem when the environment grows with the number of agents.

Theorem 4.2 (Time complexity of target assignment): Consider n agents, with communication range $r > 0$, in an environment $\mathcal{E} = [0, \ell(n)]^d$, $d \geq 1$. If $\ell(n) \geq (1+\epsilon) r n^{1/d}$, where $\epsilon \in \mathbb{R}_{>0}$, then for all algorithms in the assignment-based motion class, the time complexity of the target assignment problem is in $\Omega(n^{(d-1)/d}\ell(n))$.

Remark 4.3 ($\ell(n) \leq \ell_{crit}$): We have lower bounded the time complexity when $\ell(n)$ grows faster than some critical value, $\ell_{crit} = r n^{1/d}$. This same type of bound appears in percolation theory and the study of random geometric graphs, where it is referred to as the thermodynamic limit [12]. When $\ell(n) \leq \ell_{crit}$, congestion issues in both motion and communication become more prevalent, and a more complex communication and motion model would ideally be used. •

In the next section we introduce an asymptotically optimal algorithm in the assignment-based motion class.

V. The ETSP Assgmt Algorithm

In this section we introduce the ETSP Assgmt algorithm—an algorithm within the assignment-based motion class. We will show that when $\ell(n)$ grows more quickly than a critical value, this algorithm is asymptotically optimal. The algorithm can be described as follows.

For each $i \in \mathcal{I}$, agent i computes a constant factor approximation of the optimal ETSP tour of the n targets in $\mathbf{q}^{[i]}$, denoted $\mathrm{tour}(\mathbf{q}^{[i]})$. We can think of tour as a map which

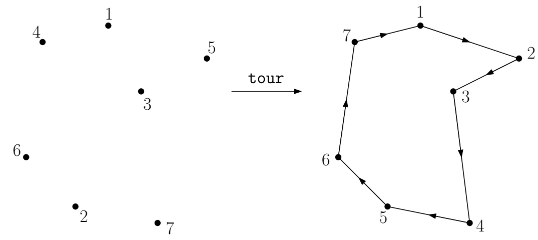


Fig. 1. The map tour, creating an ETSP tour of seven targets.

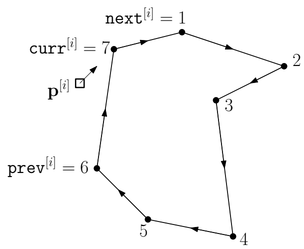
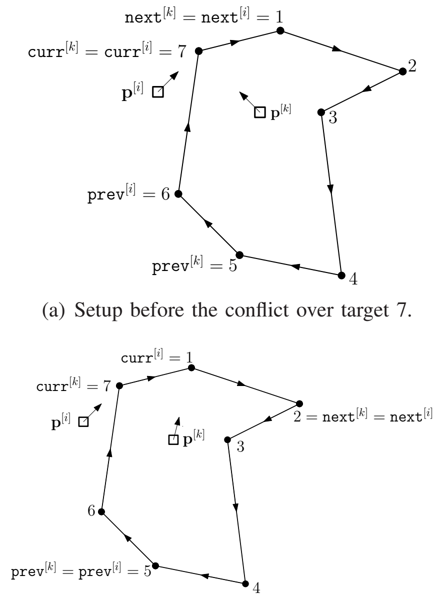


Fig. 2. The initialization for agent i.



(a) Setup before the conflict over target 7.

(b) Setup after resolution of the conflict.

Fig. 3. The resolution of a conflict between agents i and k over target 7. Since agent k is closer to target 7 than agent i, agent k wins the conflict.

reorders the indices of $\mathbf{q}^{[i]}$; $\mathrm{tour}(\mathbf{q}^{[i]}) = (\mathbf{q}^{[i]}_{\sigma(1)}, \dots, \mathbf{q}^{[i]}_{\sigma(n)})$, where $\sigma : \mathcal{I} \to \mathcal{I}$ is a bijection. Notice that this map is independent of i since all agents use the same method. An example is shown in Fig. 1. Agent i then replaces its n-tuple $\mathbf{q}^{[i]}$ with $\mathrm{tour}(\mathbf{q}^{[i]})$. Next, agent i computes the index of the closest target in $\mathbf{q}^{[i]}$, and calls it $\mathrm{curr}^{[i]}$. Agent i also maintains the index of the next target in the tour which may be available, $\mathrm{next}^{[i]}$, and first target in the tour before $\mathrm{curr}^{[i]}$ which may be available, $\mathrm{prev}^{[i]}$. Thus, $\mathrm{next}^{[i]}$ is initialized to $\mathrm{curr}^{[i]} + 1 \pmod n$ and $\mathrm{prev}^{[i]}$ to $\mathrm{curr}^{[i]} - 1 \pmod n$. This is depicted in Fig. 2. In order to "remove" assigned targets from the tuple $\mathbf{q}^{[i]}$, agent i also maintains the n-tuple, $\mathrm{status}^{[i]}$. Letting $\mathrm{status}^{[i]}(j)$ denote the jth entry in the n-tuple, the entries are given by

$$\mathrm{status}^{[i]}(j) = \begin{cases} 0, & \text{if agent } i \text{ knows } \mathbf{q}^{[i]}_j \text{ is assigned} \\ & \text{to another agent,} \\ 1, & \text{otherwise.} \end{cases} \tag{3}$$

Thus, $\mathrm{status}^{[i]}$ is initialized as the n-tuple $(1, \dots, 1)$. The initialization is summarized in Table I. At each communication round agent i executes the algorithm COMM-RD displayed in Table II at the end of the paper. The following is an informal description.

TABLE I

THE INITIALIZATION PROCEDURE FOR AGENT i.

Initialization for agent i.

Assumes: $\mathbf{q}^{[i]} := \mathbf{q}$ for each $i \in \mathcal{I}$.

Compute a TSP tour of $\mathbf{q}^{[i]}$, $\mathrm{tour}(\mathbf{q}^{[i]})$, and set $\mathbf{q}^{[i]} := \mathrm{tour}(\mathbf{q}^{[i]})$.
Compute the closest target in $\mathbf{q}^{[i]}$, and set $\mathrm{curr}^{[i]}$ equal to its index: $\mathrm{curr}^{[i]} := \arg\min_{j \in \mathcal{I}} \{\|\mathbf{q}^{[i]}_j - \mathbf{p}^{[i]}\|\}$.
Set $\mathrm{next}^{[i]} := \mathrm{curr}^{[i]} + 1 \pmod n$.
Set $\mathrm{prev}^{[i]} := \mathrm{curr}^{[i]} - 1 \pmod n$.
Set $\mathrm{status}^{[i]} := \mathbf{1}_n$ (i.e., an n-tuple containing n ones).

Informal description of COMM-RD for agent i

Assumes: $\mathrm{status}^{[i]}(s) = 0$ for each $s \in \{\mathrm{prev}^{[i]} + 1, \mathrm{prev}^{[i]} + 2, \dots, \mathrm{next}^{[i]} - 1\} \setminus \{\mathrm{curr}^{[i]}\} \pmod n$.

Broadcast $\mathrm{msg}^{[i]}$, consisting of the target indices, $\mathrm{prev}^{[i]}$, $\mathrm{curr}^{[i]}$, and $\mathrm{next}^{[i]}$, the UID i, and the distance to the current target, $\mathrm{dist}^{[i]}$.
for all messages, $\mathrm{msg}^{[k]}$, received **do**
  Set $\mathrm{status}^{[i]}(j)$ to assigned ('0') for each target j from $\mathrm{prev}^{[k]} + 1 \pmod n$ to $\mathrm{next}^{[k]} - 1 \pmod n$ not equal to $\mathrm{curr}^{[i]}$.
  if $\mathrm{prev}^{[k]} = \mathrm{next}^{[k]} = \mathrm{curr}^{[k]} \neq \mathrm{curr}^{[i]}$ **then**
    Set the status of $\mathrm{curr}^{[k]}$ to 0 (because it was missed in the previous step).
  if $\mathrm{curr}^{[i]} = \mathrm{curr}^{[k]}$ but agent i is farther from $\mathrm{curr}^{[i]}$ than agent k (ties broken with UIDs) **then**
    Set the status of $\mathrm{curr}^{[i]}$ to assigned ('0').
  if $\mathrm{curr}^{[i]} = \mathrm{curr}^{[k]}$ and agent i is closer than agent k **then**
    Leave $\mathrm{curr}^{[i]}$ unchanged. However, agent k will set $\mathrm{curr}^{[k]}$ to a new target. This target will be at least as far along the tour as the farther of $\mathrm{next}^{[i]}$ and $\mathrm{next}^{[k]}$. So, set the status of $\mathrm{next}^{[i]}$ and $\mathrm{next}^{[k]}$ to assigned ('0').
if the status of every target is assigned ('0') **then**
  Exit ETSP ASSGMT and stop motion. (This can occur only if there are more agents than targets and every target is assigned.)
else
  Update $\mathrm{curr}^{[i]}$ to the next target in the tour with status available ('1'), $\mathrm{next}^{[i]}$ to the next available target in the tour after $\mathrm{curr}^{[i]}$, and $\mathrm{prev}^{[i]}$ to the first available target in the tour before $\mathrm{curr}^{[i]}$.

Fig. 3 gives an example of COMM-RD resolving a conflict between agents i and k, over $\mathrm{curr}^{[i]} = \mathrm{curr}^{[k]}$. In this figure, all other agents are omitted.

We are now ready to define the ETSP ASSGMT algorithm.

*Definition 5.1 (*ETSP ASSGMT*):* The ETSP ASSGMT algorithm is the triplet consisting of the initialization of each agent (see Table I), the motion law in (2), and COMM-RD (see Table II), which is executed at each communication round.

A. *Correctness of* ETSP ASSGMT

We will now prove the correctness of ETSP ASSGMT. It should be noted that this result is valid for any communication graph which contains the r-disk graph as a subgraph. In order to prove correctness, let us first present some properties of the algorithm.

*Lemma 5.2 (*ETSP ASSGMT *properties):* During an execution of ETSP ASSGMT the following statements hold:

(i) Once target $j \in \mathcal{I}$, is assigned to some agent, the assignment may change, but target j remains assigned for all time.

(ii) Agent i is assigned to the target $\text{curr}^{[i]}$ which satisfies $\text{status}^{[i]}(\text{curr}^{[i]}) = 1$.

(iii) For agent i, $\text{status}^{[i]}(j) = 0$, for each $j \in \{\text{prev}^{[i]} + 1, \text{prev}^{[i]} + 2, \ldots, \text{next}^{[i]} - 1\} \setminus \{\text{curr}^{[i]}\} \pmod n$.

(iv) For agent i, $\text{status}^{[i]}(j) = 0$ only if target j is assigned to some agent $k \neq i$.

(v) If, for agent i, $\text{status}^{[i]}(j) = 0$ at some time t_1, then $\text{status}^{[i]}(j) = 0$ for all $t \geq t_1$.

(vi) If agent i receives $\text{msg}^{[k]}$ during a communication round, agent i will set $\text{status}^{[i]}(j) = 0$ for each $j \in \{\text{prev}^{[k]} + 1, \ldots, \text{next}^{[k]} - 1\} \setminus \{\text{curr}^{[i]}\} \pmod n$.

With these properties we are now ready to present the main result of this section.

Theorem 5.3 (Correctness of ETSP ASSGMT*):* For any fixed $n \in \mathbb{N}$, ETSP ASSGMT solves the target assignment problem.

The following remark displays that the ETSP ASSGMT algorithm does not solve the target assignment under the consistent knowledge assumption.

Remark 5.4 (Consistent knowledge: cont'd): Consider as in Remark 3.1 the consistent knowledge assumption for each agent's target set. Specifically, consider two agents, 1 and 2, with initial target sets $\mathcal{Q}^{[1]}(0) = \{\mathbf{q}_2\}$, $\mathcal{Q}^{[2]}(0) = \{\mathbf{q}_1, \mathbf{q}_2\}$, and any initial positions such that $\mathbf{p}^{[1]}(0) = \mathbf{q}_2$, We will have $\text{curr}^{[i]} = \text{curr}^{[j]} = 2$. However, agent 2 will win the conflict over target 2. Thus, agent 1 will set $\text{status}^{[1]}(2) = 0$, and a complete assignment will not be possible. •

B. *Time complexity for* ETSP ASSGMT

In this section we will give an upper bound on the time complexity for ETSP ASSGMT. We will show that when $\ell(n) \geq (1+\epsilon)rn^{1/d}$, for some $\epsilon \in \mathbb{R}_{>0}$, ETSP ASSGMT is asymptotically optimal among algorithms in the assignment-based motion class. Before doing this, let us first comment on the lower bound when the environment grows at a slower rate.

In what follows we show that if an agent arrives and remains at its assigned target for sufficiently long time, then it stays there for all subsequent times.

Lemma 5.5: Consider n agents executing ETSP ASSGMT with communication range $r > 0$ and assume the time delay between communication rounds, t_{max}, satisfies $t_{max} < r/v$. If there exists a time t_1 and an agent i such that $\mathbf{p}^{[i]}(t) = \text{curr}^{[i]}$ for all $t \in [t_1, t_1 + t_{max}]$, then $\mathbf{p}^{[i]}(t) = \text{curr}^{[i]}$ for all $t > t_1 + t_{max}$.

With this lemma we are now able to provide an upper bound on the time complexity of our scheme.

Theorem 5.6 (Time complexity for ETSP ASSGMT*):* Consider n agents and n targets in $[0, \ell(n)]^d$, $d \geq 1$. If $t_{max} < r/v$, then ETSP ASSGMT solves the target assignment problem in $O(n^{(d-1)/d}\ell(n) + n)$ time. If, in addition, $\ell(n) \geq (1+\epsilon)rn^{1/d}$, where $\epsilon \in \mathbb{R}_{>0}$, then the time complexity is in $\Theta(n^{(d-1)/d}\ell(n))$, and ETSP ASSGMT is asymptotically optimal among algorithms in the assignment-based motion class.

Notice that when $\ell(n)$ satisfies the bound in Theorem 5.6, and $\ell(n) \in O(n^{1/d})$, the time complexity is in $O(n)$.

We have given complexity bounds for the case when r and v are fixed constants, and $\ell(n)$ grows with n. We allow the environment $\mathcal{E}(n)$ to grow with n so that, as more agents are involved in the task, their workspace is larger. An equivalent setup would be to consider ℓ to be fixed, and allow r and v to vary inversely with the n. That is, we can introduce a set of parameters, $\tilde{\ell} = 1$, and $\tilde{r}(n)$ and $\tilde{v}(n)$ such that the time complexity will be the same as for the parameters r, v, $\ell(n)$.

Corollary 5.7 (Scaling radius and speed): Consider n agents in the environment $\mathcal{E} = [0,1]^d$, with speed $\tilde{v}(n) := v/\ell(n)$, and communication radius $\tilde{r}(n) := r/\ell(n)$, where $\ell(n) \geq (1+\epsilon)rn^{1/d}$, and $\epsilon \in \mathbb{R}_{>0}$. Then ETSP ASSGMT solves the target assignment problem with time complexity in $\Theta(n^{(d-1)/d}\ell(n))$.

Scaling the communication radius r inversely with the number of agents arises in the study of wireless networks [13]. In wireless applications there are interference and media access problems between agents in the network. Since the agents are in a compact environment, the only way to limit this interference is to scale the communication radius inversely with the number of agents. Scaling the agent speed inversely with n appears in the study of the vehicle routing problem in [7]. The inverse scaling is required to avoid collisions in the presence of traffic congestion.

C. *Simulations*

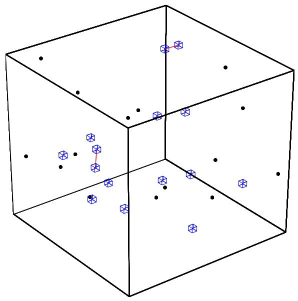
We have simulated ETSP ASSGMT in $\mathbb{R}^2$ and $\mathbb{R}^3$. To compute the ETSP tour we have used the `concorde` TSP solver.[2] A representative simulation for 15 agents in $[0, 100]^3 \subset \mathbb{R}^3$ with $r = 15$ and $v = 1$ is shown in Fig. 4. The initial configuration shown in Fig. 4(a) consists of uniformly randomly generated target and agent positions.

D. *The case of n agents and m targets*

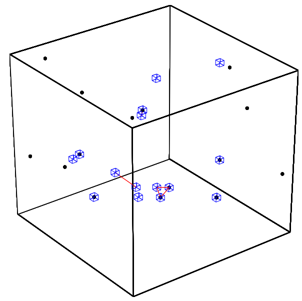
It should be noted that the ETSP ASSGMT algorithm works without any modification when there are n agents and m targets. If $m \geq n$, at completion, n targets are assigned and $m - n$ targets are not. When, $m < n$, at completion, all m targets are assigned, and the $n - m$ unassigned agents come to a stop after losing a conflict at each of the m targets. The complexity bounds are changed as follows.

The lower bound on the assignment-based motion class in Theorem 4.2, holds when $m \geq n$, and $\ell(n) \geq (1+\epsilon)rm^{1/d}$ (notice the m instead of n). The bound becomes $\Omega(\ell(n)m^{-1/d}n)$. If $m = Cn$ where $C \in \mathbb{R}_{\geq 1}$, (i.e., $m \geq n$ but they grow at the same rate), then the bound becomes $\Omega(\ell(n)n^{(d-1)/d})$. The upper bound on ETSP ASSGMT holds

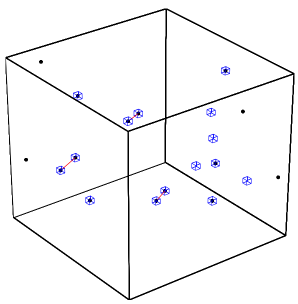
[2]The `concorde` TSP solver is available for research use at `http://www.tsp.gatech.edu/concorde/index.html`



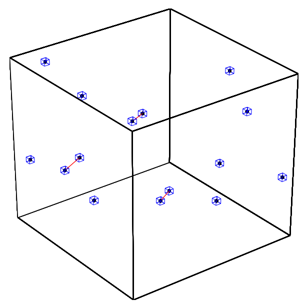
(a) Initial agent and target positions.



(b) Positions after 30 time units.



(c) Positions after 90 time units.



(d) Complete target assignment.

Fig. 4. Simulation for 15 agents, with $v = 1$ and $r = 15$. in an environment $[0, 100]^3$. The targets are spheres and the agents are cubes. An edge is drawn between two agents when they are within communication range.

for any n and m, and becomes $O(\ell(n)N^{(d-1)/d})$, where $N := \min\{n, m\}$. So our final result would be that if $m = Cn$ where $C \in \mathbb{R}_{\geq 1}$ and when $\ell(n) \geq (1 + \epsilon)rm^{1/d}$, then ETSP ASSGMT solves the target assignment problem in $\Theta(\ell(n)n^{(d-1)/d})$.

VI. CONCLUSIONS

We have developed the ETSP ASSGMT algorithm for solving the full knowledge target assignment problem. We derived worst-case asymptotic bounds on the time complexity, and we showed that among a certain class of algorithms, ETSP ASSGMT is asymptotically optimal. There are many possible extensions of this work. We have not given a lower bound on the time complexity of ETSP ASSGMT when $\ell(n) \leq \ell_{crit}$. Also, the problem is unsolved under the more general consistent knowledge assumption. We would like to extend the ETSP ASSGMT algorithm to agents with non-holonomic motion constraints. Also, it would be interesting the case where agents acquire target positions through local sensing. Finally, to derive asymptotic time bounds, we made some assumptions on the communication structure at each communication round. An interesting avenue for future study would be to more accurately address the communication issues in robotic networks.

REFERENCES

[1] B. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms*. No. 21 in Algorithmics and Combinatorics, New York: Springer Verlag, 3 ed., 2005.

TABLE II

COMMUNICATION ROUND (COMM-RD) FOR AGENT i.

Name:	COMM-RD
Goal:	Obtain information on assigned targets.
Assumes:	(i) Knowledge of the n-tuple $\mathbf{q}$, and a method for computing a constant factor TSP tour of the n targets, tour. (ii) A communication range $r > 0$.

Compute $\text{dist}^{[i]} := \|\mathbf{p}^{[i]} - \mathbf{q}^{[i]}_{\text{curr}^{[i]}}\|$.
Broadcast $\text{msg}^{[i]} := (\text{prev}^{[i]}, \text{curr}^{[i]}, \text{next}^{[i]}, i, \text{dist}^{[i]})$.
Receive $\text{msg}^{[k]}$, from each $k \neq i$ satisfying $\|\mathbf{p}^{[i]} - \mathbf{p}^{[k]}\| \leq r$.
for all $\text{msg}^{[k]}$ received **do**
  for $s = \text{prev}^{[k]} + 1$ to $\text{next}^{[k]} - 1 \pmod{n}$ **do**
    if $s \neq \text{curr}^{[i]}$ **then**
      Set $\text{status}^{[i]}(s) := 0$
  if $\text{prev}^{[k]} = \text{next}^{[k]} = \text{curr}^{[k]} \neq \text{curr}^{[i]}$ **then**
    Set $\text{status}^{[i]}(\text{curr}^{[k]}) := 0$
  if $\text{curr}^{[i]} = \text{curr}^{[k]}$ **then**
    if $(\text{dist}^{[i]} > \text{dist}^{[k]})$ OR $(\text{dist}^{[i]} = \text{dist}^{[k]}$ AND $i < k)$ **then**
      Set $\text{status}^{[i]}(\text{curr}^{[i]}) := 0$.
    else
      if $\text{next}^{[i]} \neq \text{curr}^{[i]}$ **then**
        Set $\text{status}^{[i]}(\text{next}^{[i]}) := 0$.
      if $\text{next}^{[k]} \neq \text{curr}^{[i]}$ **then**
        Set $\text{status}^{[i]}(\text{next}^{[k]}) := 0$.
if $\text{status}^{[i]}(j) = 0$ for every target j **then**
  Exit ETSP ASSGMT and stop motion.
else
  while $\text{status}^{[i]}(\text{curr}^{[i]})$=0 **do**
    $\text{curr}^{[i]} := \text{curr}^{[i]} + 1 \pmod{n}$.
  Set $\text{next}^{[i]} := \text{curr}^{[i]} + 1 \pmod{n}$.
  while $\text{status}^{[i]}(\text{next}^{[i]})$=0 **do**
    $\text{next}^{[i]} := \text{next}^{[i]} + 1 \pmod{n}$.
  while $\text{status}^{[i]}(\text{prev}^{[i]})$=0 **do**
    $\text{prev}^{[i]} := \text{prev}^{[i]} - 1 \pmod{n}$.

[2] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics*, vol. 2, pp. 83–97, 1955.
[3] D. P. Bertsekas and D. A. Castañón, "Parallel synchronous and asynchronous implementations of the auction algorithm," *Parallel Computing*, vol. 17, pp. 707–732, 1991.
[4] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Belmont, MA: Athena Scientific, 1997.
[5] D. A. Castañón and C. Wu, "Distributed algorithms for dynamic reassignment," in *IEEE Conf. on Decision and Control*, (Maui, HI), pp. 13–18, Dec. 2003.
[6] B. J. Moore and K. M. Passino, "Distributed task assignment for mobile agents," *IEEE Transactions on Automatic Control*, 2006. to appear.
[7] V. Sharma, M. Savchenko, E. Frazzoli, and P. Voulgaris, "Time complexity of sensor-based vehicle routing," in *Robotics: Science and Systems* (S. Thrun, G. Sukhatme, S. Schaal, and O. Brock, eds.), pp. 297–304, Cambridge, MA: MIT Press, 2005.
[8] S. Martínez, F. Bullo, J. Cortés, and E. Frazzoli, "On synchronous robotic networks – Part I: Models, tasks and complexity notions. & Part II: Time complexity of rendezvous and deployment algorithms," *IEEE Transactions on Automatic Control*, Apr. 2005. Submitted.
[9] S. L. Smith and F. Bullo, "Target assignment for robotic networks: asymptotic performance under limited communication." Available electronically at http://arxiv.org/abs/cs.RO/0703067, Mar. 2007.
[10] K. J. Supowit, E. M. Reingold, and D. A. Plaisted, "The traveling salesman problem and minimum mathcing in the unit square," *SIAM Journal on Computing*, vol. 12, pp. 144–156, 1983.
[11] N. Christofides, "Worst-case analysis of a new heuristic for the traveling salesman problem," Tech. Rep. 388, Carnegie-Mellon University, Pittsburgh, PA,, Apr. 1976.
[12] M. Penrose, *Random Geometric Graphs*. Oxford Studies in Probability, Oxford, UK: Oxford University Press, 2003.
[13] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, 2000.