Target assignment for robotic networks: Worst-case and stochastic performance in dense environments

Stephen L. Smith Francesco Bullo

***Abstract*— Consider an equal number of mobile robotic agents and distinct target locations dispersed in an environment. Each agent has a limited communication range and knowledge of every target's position. We study the following target assignment problem: design a distributed algorithm with which the agents divide the targets among themselves and, simultaneously, move to their unique target. This paper focuses on "dense" environments, where the sum of the communication footprints is larger than the area of the environment. We introduce the class of monotonic algorithms, whose worst-case completion time is lower bounded by the area of the environment. We propose a monotonic algorithm called GRID ASSGMT and characterize its asymptotic performance: First, the algorithm is an asymptotically optimal monotonic algorithm for worst-case initial conditions. Second, for uniformly randomly distributed agents and targets the completion time is upper bounded by the diameter of the environment with high probability. Third, if the number of agents exceeds the number of targets by a logarithmic factor, then the completion time is constant with high probability. Our algorithm also solves a sensor based target assignment problem where agents have no initial knowledge of target positions, but acquire them via limited range sensing.**

I. INTRODUCTION

Consider n mobile robotic agents, equipped with wireless transceivers for limited range communication, dispersed in an environment $\mathcal{E} \subset \mathbb{R}^2$. Suppose the environment also contains n target locations, and each agent is given a list of their positions (the positions may be GPS coordinates). The task is for the agents to divide the targets among themselves so that in minimum time, each target location is occupied by an agent. Since no *a priori* target-agent assignment is given, the agents must solve the problem through communication and motion. This *target assignment problem* could arise in several applications, such as UAV's on a surveillance mission, where the targets are the centers of their desired loitering patterns. Alternatively, one could imagine a mobile sensor network roughly deployed in a remote or hostile environment; the sensors must solve a target assignment problem to reach the desired (target) configuration.

a) Centralized and parallel assignment: There is a wealth of literature on solving centralized versions of the target assignment problem. The problem of assigning one agent to each target is known as the *maximum matching problem*. The problem of finding a complete assignment/matching which minimizes the sum of distances (respectively, the maximum distance) from each agent to its assigned target, is known as the *sum assignment* (respectively, *bottleneck assignment*) problem. There exist efficient polynomial time algorithms for the solution of all these problems [1], [2], [3]. Additionally, the sum assignment problem can be solved in a parallel fashion via the *auction algorithm* [4]. However, these solutions do not directly apply to our problem where, due to the agents' limited communication range, the communication topology is time-varying, and possibly disconnected.

b) Target/task assignment in robotic networks: There has been a significant amount of work on decentralized task assignment for UAVs (or UGVs), see for example [5], [6], [7], [8]. The goal is generally to assign vehicles to spatially distributed tasks while maximizing the "score" of the mission. These works develop advanced heuristic methods, and demonstrate their effectiveness through detailed simulation or real world implementation. In [9] the auction algorithm is adapted to solve a task allocation problem in the presence of communication delays. There has also been prior work on target assignment problems [10], [11]. In [10] the authors formulate a target assignment problem as a multi-player game and seek to optimize a global utility. In [11] an algorithm based on hybrid systems tools is developed and its performance is characterized by a bound on the number of switches of the hybrid system.

In [12] we began an investigation into the scalability properties of the minimum-time target assignment problem for agents with limited communication capabilities. We focused on characterizing the completion time as the number of agents, n, grows, and the environment $\mathcal{E}(n)$ grows to accommodate them. We introduced the ETSP ASSGMT algorithm with worst-case completion time in $O(\sqrt{|\mathcal{E}(n)|n})$.[1]

c) Contribution: In this paper we introduce a broad class algorithms called *monotonic algorithms*. We show that in "sparse" environments where communication is infrequent, i.e., when $|\mathcal{E}(n)|/n \to +\infty$, every monotonic algorithm has worst-case completion time in $\Omega(\sqrt{|\mathcal{E}(n)|n})$. In "dense" environments, i.e., when $|\mathcal{E}(n)|/n \to 0^+$, every monotonic algorithm has worst-case completion time in $\Omega(|\mathcal{E}(n)|)$. We then develop a novel distributed monotonic algorithm called GRID ASSGMT. In this algorithm, the agents partition the environment into cells, and determine local maximum assignments in the cell which they occupy. A leader is elected in each cell, and through communication between leaders of adjacent cells, local assignments are merged into a complete global assignment. We show that the

This material is based upon work supported in part by ARO MURI Award W911NF-05-1-0219 and NSF SENSORS Award IIS-0330008.

S. L. Smith and F. Bullo are with the Center for Control, Dynamical Systems and Computation, University of California, Santa Barbara, CA 93106, USA, {stephen,bullo}@engineering.ucsb.edu

[1] $|\mathcal{E}(n)|$ denotes the area of $\mathcal{E}(n)$, and, $O(\cdot)$ and $\Omega(\cdot)$ are the asymptotic notations for upper and lower bounds, respectively (see Section II).

worst-case completion time of the GRID ASSGMT algorithm belongs to $O(|\mathcal{E}(n)|)$. Thus, in "dense" environments GRID ASSGMT is an asymptotically optimal monotonic algorithm for worst-case initial conditions. Hence, our two algorithms are complementary: ETSP ASSGMT has better performance in sparse environments, while GRID ASSGMT has better performance in dense environments. We also characterize the GRID ASSGMT algorithms' stochastic performance in "dense" environments. First, if the agents and targets are uniformly randomly distributed, then the completion time belongs to $O(\sqrt{|\mathcal{E}(n)|})$ with high probability. Second, if there are n agents and only $n/\log n$ targets, then the completion time belongs to $O(1)$ with high probability. Due to space constraints all proofs are omitted, and can be found in [13].

II. COMBINATORIC AND STOCHASTIC PRELIMINARIES

In this section we review a few useful results on the centralized matching problem, occupancy problems, and random geometric graphs. We let $\mathbb{R}$, $\mathbb{R}_{>0}$ and $\mathbb{N}$ denote the set of real numbers, the set of positive real numbers, and the set of positive integers, respectively. Given a finite set A, we let $|A|$ denote its cardinality, and given an infinite set $A \subset \mathbb{R}^2$ we let $|A|$ denote its area. 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 $f(n) \in \Omega(g)$, then we say $f(n) \in \Theta(g)$. We say that event $A(n)$ occurs *with high probability* (w.h.p.) if the probability of $A(n)$ occurring tends to one as $n \to +\infty$.

A. Centralized matching

Consider n persons and the problem of dividing them among n tasks. For each person i, there is a nonempty set $\mathcal{Q}^{[i]}$ of tasks to which i can be assigned. An *assignment* or *matching* M is a set of person-task pairs (i, j) such that $j \in \mathcal{Q}^{[i]}$ for all $(i, j) \in M$, and such that for each person i (likewise, task j) there is at most one pair $(i, j) \in M$. The matching M is a *maximum* matching if for every matching $\tilde{M}$, we have $|\tilde{M}| \leq |M|$. If $|M| = n$, then the matching is *complete*. The matching M is *maximal* if there does not exist a matching $\tilde{M}$, such that $\tilde{M}$ is a strict superset of M. Let us present a simple algorithm for computing a maximal matching. In this algorithm we choose the target-agent pair with lowest cost, add it to our matching, remove that target and agent from the problem, and repeat. If each person can be assigned to any of the n tasks, then this algorithm determines a complete, and thus maximum, matching.

MAXIMAL MATCH, outputs a maximal matching M

Initialize $M := \emptyset$, and $\mathcal{I}_i := \{1, \ldots, n\}$.
while *there exists an* $i \in \mathcal{I}_i$ *with* $|\mathcal{Q}^{[i]}| \neq 0$ **do**
  Compute the indices $(i^*, j^*) := \arg\min_{i \in \mathcal{I}_i, j \in \mathcal{Q}^{[i]}} c_{ij}$
  Set $M := M \cup (i^*, j^*)$, $\mathcal{I}_i := \mathcal{I}_i \setminus \{i^*\}$, and for each $i \in \mathcal{I}_i$, $\mathcal{Q}^{[i]} := \mathcal{Q}^{[i]} \setminus \{j^*\}$

B. Occupancy problems

Occupancy problems are concerned with randomly distributing m balls into n equally sized bins. The two results we present here will be useful in our analysis.

Theorem 2.1 (Occupancy properties, [14], [15]): Consider uniformly randomly distributing m balls into n bins and let γ be any function such that $\gamma(n) \to +\infty$ as $n \to +\infty$. The following statements hold:

(i) if $m = n$, then w.h.p. each bin contains $O\left(\frac{\log n}{\log \log n}\right)$ balls;
(ii) if $m = n \log n + \gamma(n)n$, then w.h.p. there exist no empty bins;
(iii) if $m = n \log n - \gamma(n)n$, then w.h.p. there exists an empty bin;
(iv) if $m = Kn \log n$, where $K > 1/\log(4/\mathrm{e})$, then w.h.p. every bin contains $\Theta(\log n)$ balls.

We will be interested in partitioning a square environment into equally sized and openly disjoint square bins such that the area of each bin is "small." To do this, we require the following simple fact.

Lemma 2.2 (Dividing the environment): Given $n \in \mathbb{N}$ and $r_{\text{comm}} > 0$, consider a square environment $\mathcal{E}(n)$. If $\mathcal{E}(n)$ is partitioned into b^2 equally sized and openly disjoint square bins, where $b := \lceil \sqrt{5|\mathcal{E}(n)|}/r_{\text{comm}} \rceil$, then the area of each bin is no more than $r_{\text{comm}}^2/5$. Moreover, if $x, y \in \mathcal{E}(n)$ are in the same bin or in adjacent bins, then $\|x - y\| \leq r_{\text{comm}}$.

C. Random geometric graphs

For $n \in \mathbb{N}$ and $r_{\text{comm}} \in \mathbb{R}_{>0}$, a planar *geometric graph* $G(n, r_{\text{comm}})$ consists of n vertices in $\mathbb{R}^2$, and undirected edges connecting all vertex pairs $\{x, y\}$ with $\|x - y\| \leq r_{\text{comm}}$. We also refer to this as the r_{comm}-geometric graph. If the vertices are randomly distributed in some subset of $\mathbb{R}^2$, then we call the graph a *random geometric graph*.

Theorem 2.3 (Connectivity of geometric graphs, [16]): Consider the random geometric graph $G(n, r_{\text{comm}})$ obtained by uniformly randomly distributing n points in the square environment $\mathcal{E}(n)$ with $\pi r_{\text{comm}}^2/|\mathcal{E}(n)| = (\log n + \gamma(n))/n$. Then $G(n, r_{\text{comm}})$ is connected w.h.p. if and only if $\gamma(n) \to +\infty$ as $n \to +\infty$.

This theorem will be important for understanding some of our results, as it provides a bound on the environment size necessary for the communication graph of n randomly deployed agents to be asymptotically connected.

III. NETWORK MODEL AND PROBLEM STATEMENT

In this section we formalize our agent and target models and define the sparse and dense environments.

A. Robotic network model

Consider n agents in an environment $\mathcal{E}(n) := [0, \ell(n)]^2 \subset \mathbb{R}^2$, where $\ell(n) > 0$ (that is, $\mathcal{E}(n)$ is a square with side length $\ell(n)$). The environment $\mathcal{E}(n)$ is compact for each n but its size depends on n. A robotic agent, $\mathcal{A}^{[i]}$, $i \in \mathcal{I} := \{1, \ldots, n\}$, is described by the tuple $\mathcal{A}^{[i]} := \{\text{UID}^{[i]}, \mathbf{p}^{[i]}, r_{\text{comm}}, \mathbf{u}^{[i]}, M^{[i]}\}$, where the quantities are as follows: Its unique identifier (UID) is $\text{UID}^{[i]}$, taken from the set $I_{\text{UID}} \subset \mathbb{N}$. Note that, each agent does not know the set of UIDs being used and thus does not initially know the magnitude of its UID relative to those of other

agents. Its position is $\mathbf{p}^{[i]} \in \mathcal{E}(n)$. Its communication range is $r_{\text{comm}} > 0$, i.e., two agents, $\mathcal{A}^{[i]}$ and $\mathcal{A}^{[k]}$, $i, k \in \mathcal{I}$, can communicate if and only if $\|\mathbf{p}^{[i]} - \mathbf{p}^{[k]}\| \leq r_{\text{comm}}$. Its continuous time velocity input is $\mathbf{u}^{[i]}$, corresponding to the kinematic model $\dot{\mathbf{p}}^{[i]} = \mathbf{u}^{[i]}$, where $\|\mathbf{u}^{[i]}\| \leq v_{\max}$ for some $v_{\max} > 0$. Finally, its memory is $M^{[i]}$ and is of cardinality (size) $|M^{[i]}|$. From now on, we refer to agent $\mathcal{A}^{[i]}$ as agent i.

The agents move in continuous time and communicate according to a synchronous discrete time schedule consisting of an increasing sequence $\{t_k\}_{k\in\mathbb{N}}$ of time instants with no accumulation points. We assume $|t_{k+1} - t_k| \leq t_{\max}$, for all $k \in \mathbb{N}$, where $t_{\max} \in \mathbb{R}_{>0}$. At each communication round, agents can exchange messages of length $O(\log n)$.[2] Communication round k occurs at time t_k, and all messages are sent and received instantaneously at t_k. Motion then occurs from t_k until t_{k+1}. In this setup we are emphasizing the time complexity due to the motion of the agents.

B. The target assignment problem

Let $\mathcal{Q} := \{\mathbf{q}_1, \ldots, \mathbf{q}_n\} \subset \mathcal{E}(n)$ be a set of distinct target locations. In this paper we assume that each agent knows the position of every target. Thus, agent i's memory, $M^{[i]}$, contains a copy of $\mathcal{Q}$, which we denote $\mathcal{Q}^{[i]}$. To store $\mathcal{Q}^{[i]}$ we must assume the size of each agents' memory, $|M^{[i]}|$, is in $\Omega(n)$. We refer to the assumption that each agent knows all target positions as the *full knowledge* assumption. 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 $T \geq 0$ such that for each target $\mathbf{q}_j \in \mathcal{Q}$, there is a unique agent $i \in \mathcal{I}$, with $\mathbf{p}^{[i]}(t) = \mathbf{q}_j$ for all $t \geq T$.

If the task begins at time $t = 0$, then the *completion time* T_c of target assignment is the minimum $T \geq 0$, such that for each target $\mathbf{q}_j \in \mathcal{Q}$, there is a unique agent $i \in \mathcal{I}$, with $\mathbf{p}^{[i]}(t) = \mathbf{q}_j$ for all $t \geq T$.

C. Sparse, dense, and critical environments

We wish to study the scalability of a particular approach to the target assignment problem; that is, how the completion time increases as we increase the number of agents, n. The velocity $v_{\max}$ and communication range r_{comm} of each agent are independent of n. However, we assume that the size of the environment increases with n in order to accommodate an increase in agents. Borrowing terms from the random geometric graph literature [16], we say that the environment is *sparse* if, as we increase the number of agents, the environment grows quickly enough that the density of agents (as measured by the sum of their communication footprints) decreases; we say the environment is *critical*, if the density is constant, and we say the environment is *dense* if the density increases. Formally, we have the following definition.

Definition 3.1 (Environment size): Environment $\mathcal{E}(n)$ is

(i) *sparse* if $|\mathcal{E}(n)|/n \to +\infty$ as $n \to +\infty$;
(ii) *critical* if $|\mathcal{E}(n)|/n \to \text{const} \in \mathbb{R}_{>0}$ as $n \to +\infty$;
(iii) *dense* if $|\mathcal{E}(n)|/n \to 0^+$, as $n \to +\infty$.

Note that a dense environment does not imply the communication graph between agents is dense; from Theorem 2.3 we see that the communication graph at random agent positions in a dense environment may not even be connected.

D. Monotonic algorithms and ETSP ASSGMT

We introduce a class of algorithms which provides an intuitive approach to target assignment.

Definition 3.2 (Monotonic algorithms): A deterministic algorithm for target assignment is *monotonic* if, for a subset of agents $\mathcal{J} \subset \mathcal{I}$, a target $\mathbf{q}_j$, and time $t_1 > 0$, we have $\mathbf{p}^{[i]}(t_1) = \mathbf{q}_j$ for each $i \in \mathcal{J}$, then there exists an agent $i \in \mathcal{J}$ such that $\mathbf{p}^{[i]}(t) = \mathbf{q}_j$ for all $t > t_1$.

We call these algorithms "monotonic" since occupied targets remain occupied for all time, and thus the number of occupied targets monotonically increases throughout the execution. We now lower bound the completion time of the target assignment problem for any monotonic algorithm.

Theorem 3.3 (Time complexity of target assignment): Consider n agents, with communication range $r_{\text{comm}} > 0$, and n targets in $\mathcal{E}(n)$. For all monotonic algorithms the worst-case completion time T_c of the target assignment problem is lower bounded as follows:

(i) if $\mathcal{E}(n)$ is sparse, then $T_c \in \Omega(\sqrt{n|\mathcal{E}(n)|})$;
(ii) if $\mathcal{E}(n)$ is critical, then $T_c \in \Omega(n)$;
(iii) if $\mathcal{E}(n)$ is dense, then $T_c \in \Omega(|\mathcal{E}(n)|)$.

The idea behind the proof is to place the targets in $\mathcal{E}(n)$ such that the r_{comm}-geometric graph generated by their positions has a maximum number of disconnected components.

In [12] we introduced the ETSP ASSGMT algorithm, which is a monotonic algorithm. In this algorithm, each agent computes a Euclidean traveling salesperson tour of the n targets, turning the cloud of target points into an ordered ring. Agents move along the ring looking for the next available target, and when they communicate, they exchange information on how far it is to the next available target along the ring. Combining Theorem 3.3 with the worst-case bound in [12] we obtain the following.

*Theorem 3.4 (*ETSP ASSGMT*):* For any initial positions of n agents and n targets in $\mathcal{E}(n)$, ETSP ASSGMT solves the target assignment problem in $O(\sqrt{n|\mathcal{E}(n)|})$ time. If $\mathcal{E}(n)$ is sparse or critical, then ETSP ASSGMT is an asymptotically optimal monotonic algorithm for worst-case initial positions.

IV. THE GRID ASSGMT ALGORITHM

In this section we introduce a monotonic algorithm called GRID ASSGMT. In this algorithm, the agents partition the environment into cells. Agents then determine local maximum assignments, and elect a leader in the cell which they occupy. Through communication between leaders of adjacent cells, each leader obtains estimates of the location of free targets, and uses this information to guide unassigned agents to free targets. We show that in critical or dense environments, GRID ASSGMT is an asymptotically optimal monotonic algorithm for worst-case initial conditions. In addition, by utilizing the results of Section II-B, we characterize the stochastic performance of GRID ASSGMT.

[2]The number of bits required to represent an ID, unique among n agents, is directly proportional to the logarithm of n.

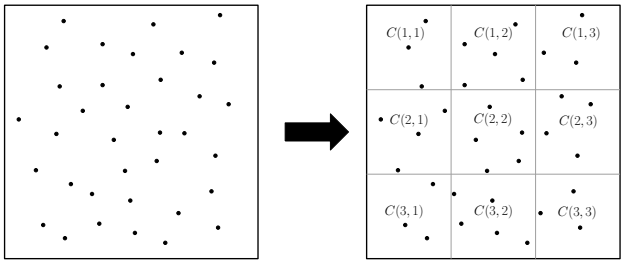


Fig. 1. Partitioning the environment $\mathcal{E}(n)$ into $b^2 = 9$ cells.

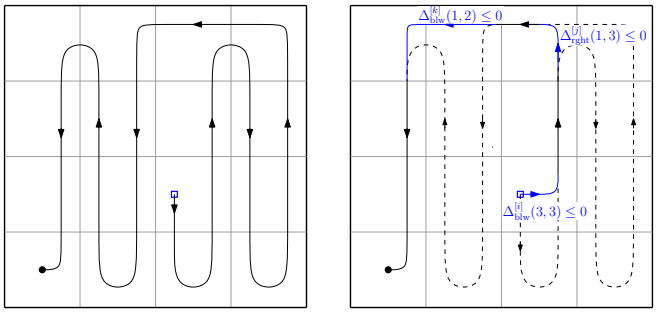


Fig. 2. Unassigned agent motion. Left: The nominal order in which an agent (blue square) searches the cells in the absence of communication. Right: The shortened path due to the non-positive estimates from leader i of $C(3,3)$, leader j of $C(1,3)$ and leader k of $C(1,2)$.

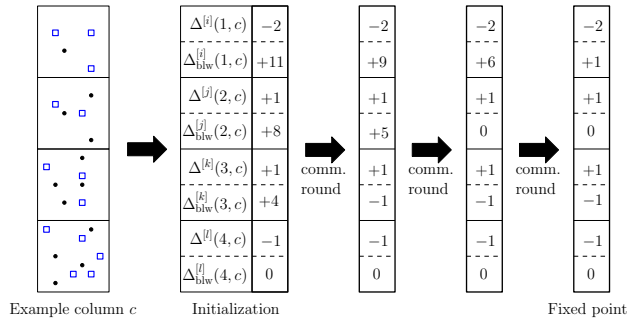


Fig. 3. Leader communication. Column c contains agents (blue squares) and targets (black disks). The figure shows leaders i of $C(1,c)$, j of $C(2,c)$, k of $C(3,c)$, and l of $C(4,c)$ initializing and updating their estimates of Δ and Δ_{blw}. Estimates converge to true values in three iterations.

A. Algorithm description

We assume that each agent knows the target positions $\mathcal{Q}$ (i.e., $\mathcal{Q}^{[i]} := \mathcal{Q}$) and the environment $\mathcal{E}(n)$. With this information, each agent partitions the environment into b^2 equally sized square cells, where $b \in \mathbb{N}$. It then labels the cells like entries in a matrix, so cell $C(r,c)$ resides in the rth row and cth column, as shown in Fig. 1. Since the agents start with the same information, they all create the same partition. The quantity b is chosen so that an agent in cell $C(r,c)$ is within communication range of any agent in cells $C(r,c)$, $C(r-1,c)$, $C(r+1,c)$, $C(r,c-1)$, and $C(r,c+1)$. In light of Lemma 2.2, we see that this is satisfied when $b = \lceil \sqrt{5|\mathcal{E}(n)|}/r_{\text{comm}} \rceil$. With this, we now outline the GRID ASSGMT algorithm. A complete description is given [13].

Outline of the GRID ASSGMT algorithm

***Initialization and role assignment*:** Each agent partitions $\mathcal{E}(n)$ as described above. In each cell, agents use MAXIMAL MATCH to find a maximum assignment between agents and targets occupying the cell, and assigned agents elect a leader among them. Accordingly, agents are labeled leader, unassigned, or assigned non-leader.

Assigned non-leader agents: Each assigned non-leader agent moves to its assigned target and goes silent.

Cell leaders: Each cell leader assigns free targets in its cell to unassigned agents that enter the cell. In addition, each cell leader estimates the number of available targets in all cells below it in its column (denoted $\Delta^{[i]}_{\text{blw}}(r,c)$ for leader i of cell $C(r,c)$). To maintain the estimates, cell leaders communicate to the cell leader in the cell directly above. Cell leaders in the top row communicate to the cell leader directly to the right to obtain an estimate of the number of available targets in all columns to the right (denoted $\Delta^{[j]}_{\text{rght}}(1,c)$ for leader j of cell $C(1,c)$).

Unassigned agents: Each unassigned agent seeks a free target by entering cells and querying their respective leaders. The motion of unassigned agents is illustrated in Fig. 2. Assuming no communication with the leaders, the nominal order in which an unassigned agent visits all cells of the grid is shown in the left-hand figure. The way in which this path is shortened as the unassigned agent receives available target estimates from cell leaders is shown on the right-hand figure.

Remark 4.1 (Cell leader computations): If agent i is the leader of cell $C(r,c)$, it computes $\Delta^{[i]}(r,c)$, which is (# of targets) $-$ (# of agents) in $C(r,c)$. In addition, leader i maintains $\Delta^{[i]}_{\text{blw}}(r,c)$, which is an estimate of (# of targets) $-$ (# of agents) in cells $C(r+1,c)$ to $C(b,c)$. This quantity must be estimated because agent i does not initially know the number of agents in cells $C(r+1,c)$ to $C(b,c)$. The variable $\Delta^{[i]}_{\text{blw}}(r,c)$ is initialized to the number of targets in cells $C(r+1,c)$ to $C(b,c)$, which is necessarily no smaller than the actual value. Then, at each communication round agent i updates its estimate by communicating with the leaders in cells $C(r-1,c)$ and $C(r+1,c)$:

1 Send $\text{msg}^{[i]} := \Delta^{[i]}_{\text{blw}}(r,c) + \Delta^{[i]}(r,c)$ to leader in $C(r-1,c)$ and receive $\text{msg}^{[k]}$ from leader k of $C(r+1,c)$.

2 Set $\Delta^{[i]}_{\text{blw}}(r,c) := \text{msg}^{[k]} = \Delta^{[k]}_{\text{blw}}(r+1,c) + \Delta^{[k]}(r+1,c)$.

This update procedure is depicted in Fig. 3. A leader j of cell $C(1,c)$ in the top row uses a similar method to maintain the estimate $\Delta^{[j]}_{\text{rght}}(1,c)$. It should be noted that as unassigned agents enter and exit cells, the actual values of Δ_{blw} and Δ_{rght} change. Thus, there is a procedure (which is fully detailed in [13]) whereby agents send `enter` and `exit` messages to cell leaders, so that they can maintain their estimates. •

Remark 4.2 (Unassigned agent motion): The motion of unassigned agents can be described as follows. First, each unassigned agent seeks a free target in its column. It queries the leader of its current cell about free targets in its column, below its current cell. If the leader's estimate $\Delta^{[i]}_{\text{blw}}(r,c)$ is positive, then the agent moves down the column. Otherwise,

the agent moves up the column. While moving down, upon entering a new cell the agent first queries the cell leader on free targets in the cell, and then on free targets in cells below. If the agent starts moving up the column, then it only queries cell leaders on free targets in its current cell (since all targets below are taken).

Second, if the agent reaches the top cell of its column, then the column contains no free targets. To transfer to a new column, the agent queries the leader of the top cell about free targets in all columns to the right. If the leader's estimate $\Delta^{[j]}_{\text{rght}}(1,c)$ is positive, then the agent moves to the right; otherwise, the agent moves to the left. Upon reaching the cell to the left or right, the agent recommences the column procedure. •

Remark 4.3 (Comments on GRID ASSGMT*):* (1) Agents move at speed $v_{\max}$, and to transfer between cells agents move toward the center of the new cell. (2) If an agent or target lies on the boundary between cells, a simple tie breaking scheme is used assign it to a cell. (3) In our presentation, we implicitly assumed that every cell initially contains at least one agent and one target. If a cell has no targets, then any agents initially in the cell leave, and the empty cell is then ignored. If a cell initially contains targets but no agents, then the first agents to enter the cell run the MAXIMAL MATCH algorithm and a leader is elected. (4) In our description, agents use the top row to transfer to a new column. This choice of "transfer column" is arbitrary and the top row was chosen for simplicity of presentation. Intuitively, one could argue that the middle row is a more efficient choice. In the upcoming analysis we show that such a choice does not affect the algorithm's asymptotic performance. •

B. Correctness and time complexity of GRID ASSGMT

We now present our main results on GRID ASSGMT.

*Theorem 4.4 (*GRID ASSGMT *correctness and worst-case):* For any initial positions of n agents and n targets in $\mathcal{E}(n)$, GRID ASSGMT solves the target assignment problem in $O(|\mathcal{E}(n)|)$ time. In addition, if $\mathcal{E}(n)$ is dense or critical, then GRID ASSGMT is an asymptotically optimal monotonic algorithm for worst-case initial positions.

*Remark 4.5 (*GRID ASSGMT *vs.* ETSP ASSGMT*):* The worst-case bound for ETSP ASSGMT in Theorem 3.4 was $O(\sqrt{|\mathcal{E}(n)|n})$. Thus, in sparse environments ETSP ASSGMT performs better, where as in dense environments GRID ASSGMT performs better. In critical environments, the bounds are equal. In practice, a robot can determine which algorithm to run by comparing the area of the environment $|\mathcal{E}(n)|$ to the area of n disks of radius r_{comm}. That is, a robot could use a rule such as the following: if $|\mathcal{E}(n)| > \pi r^2_{\text{comm}} n$, then execute ETSP ASSGMT, else if $|\mathcal{E}(n)| < \pi r^2_{\text{comm}} n$, then execute GRID ASSGMT. •

In the following theorem we will see that for randomly placed targets and agents, the performance of GRID ASSGMT is considerably better than in the worst-case.

Theorem 4.6 (Stochastic time complexity): Consider n agents and n targets, uniformly randomly distributed in $\mathcal{E}(n)$. Then GRID ASSGMT solves the target assignment problem in $O(\sqrt{|\mathcal{E}(n)|})$ time with high probability if

$$|\mathcal{E}(n)| \leq \frac{r^2_{\text{comm}}}{5} \frac{n}{\log n + \gamma(n)},$$

where γ is any function such that $\gamma(n) \to +\infty$ as $n \to +\infty$.

Remark 4.7 (Generalization of Theorem 4.6): The bound in Theorem 4.6 holds not only for uniformly randomly distributed initial positions, but for any initial positions such that every cell contains at least one target and at least one agent.

Theorem 4.8 (Stochastic time complexity, cont'd): Consider n agents and $n/\log n$ targets, uniformly randomly distributed in $\mathcal{E}(n)$. Then GRID ASSGMT solves the target assignment problem in $O(1)$ time with high probability if there exists $K > 1/\log(4/\mathrm{e})$, such that

$$|\mathcal{E}(n)| \leq \frac{r^2_{\text{comm}}}{5} \frac{n}{K \log n}.$$

Remark 4.9 (Wireless congestion): Since wireless communication is a shared medium, simultaneous messages sent in close proximity will collide, resulting in dropped packets. Clear reception of a signal requires that no other signals are present at the same point in time and space. As the density of agents increases (as measured by their communication footprints), so does wireless communication congestion. In the design of GRID ASSGMT we have tried to limit the amount of simultaneous communication. To this end we introduced a leader in each cell, who sent messages (of size $O(\log n)$) only to its adjacent cells, and all other assigned agents were silent. However, to fully take wireless congestion into account, we would require a more sophisticated communication model than the r_{comm}-geometric graph. •

C. Ideas behind main GRID ASSGMT results

In this section we give some intuition into the proofs of the theorems in Section IV-B.

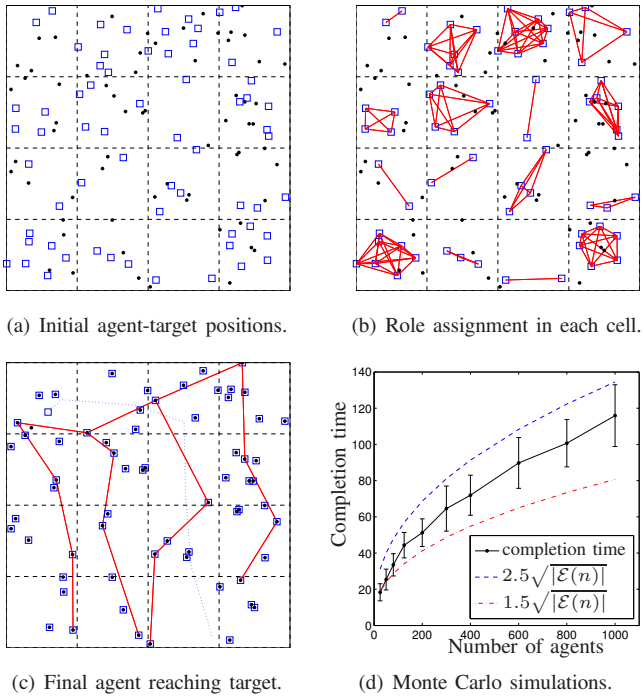
First, the proof of Theorem 4.4 utilizes the fact that in the worst-case an agent will have to visit every cell once. Second, the proof of Theorem 4.6 requires that the estimates maintained by the leaders of each cell converge to the true values. To discuss convergence, let $\Delta(r,c)(t)$ denote the difference between the number of targets in $C(r,c)$ and the number of agents in $C(r,c)$ at time $t > 0$ (recall that leader i's estimate of $\Delta(r,c)(t)$ is denoted $\Delta^{[i]}(r,c)$). In our model, communication round k occurs instantaneously at time t_k so t_k^- denotes start of the round, and t_k^+, its completion. With this notation, the convergence of the estimates can be stated as follows. If agent $i \in \mathcal{I}$ is the leader of cell $C(r,c)$, then for each communication time t_k, $k \in \mathbb{N}$:

(i) $\Delta^{[i]}(r,c)(t_k^+) = \Delta(r,c)(t_k)$;

(ii) $\Delta^{[i]}_{\text{blw}}(r,c)(t_k^+) \geq \sum_{r^*=r+1}^{b} \Delta(r^*,c)(t_k)$;

(iii) if $k > b$ and each cell in column c contains a leader, then $\Delta^{[i]}_{\text{blw}}(r,c)(t_k^+) = \sum_{r^*=r+1}^{b} \Delta(r^*,c)(t_k)$.

Thus, the Δ_{blw} estimates are never underestimates, and if there is a leader in every cell, the estimates converge after $b := \lceil \sqrt{5|\mathcal{E}(n)|}/r_{\text{comm}} \rceil$ communication rounds. We have an



(a) Initial agent-target positions. (b) Role assignment in each cell.

(c) Final agent reaching target. (d) Monte Carlo simulations.

Fig. 4. A simulation of 65 agents (blue squares) and targets (black disks) in a dense environment is shown in (a-c). Red lines are drawn when two agents are communicating. A Monte Carlo simulation is shown in (d).

analogous result for the convergence of $\Delta^{[i]}_{\text{rght}}(r,c)$. These two results, combined with Theorem 2.1(ii), are used to prove Theorem 4.6. Finally, Theorem 4.8 is proved using the occupancy result in Theorem 2.1(iv).

V. Simulations

Fig. 4 contains a representative simulation of the Grid Assgmt algorithm for 65 agents and targets uniformly randomly distributed in a dense environment. Fig. 4(c) shows the communication between the leaders of each cell (solid red lines), and the trajectory of the final agent (dashed blue line) as it approaches its target in cell $C(1,1)$. Fig. 4(d) contains the numerical outcomes of Monte Carlo simulations. In the simulations $r_{\text{comm}} = 10$, $v_{\max} = 1$, $|\mathcal{E}(n)| = r^2_{\text{comm}} n/(6\log n)$, and each data point is the mean completion time of 30 trials, where each trial was performed at randomly generated agent and target positions. For simplicity of implementation we discard trials in which there exist a cell without targets. This is justified by the fact that w.h.p. every cell contains at least one target, and thus the number of discarded trails tends to zero as n increases. Error bars show plus/minus one standard deviation. The simulation suggests that asymptotically, the expected completion time is bounded below by $1.5\sqrt{|\mathcal{E}(n)|}$ and above by $2.5\sqrt{|\mathcal{E}(n)|}$.

VI. Discussion and Extensions

A. A sensor based version

In describing the Grid Assgmt algorithm, we assumed that each agent knows the position of all targets. The algorithm also works when each agent does not know the position of any targets, but has a sensing range r_{sense}, with which it can sense the positions of targets in range. If each agent can partition the environment as in Fig. 1, and if $r_{sense} \geq \sqrt{2/5}r$ so that each agent can sense the position of all targets in its current cell, then Grid Assgmt (with minor modifications) solves the target assignment problem, and the completion time results still hold.

B. Conclusions

We have developed the Grid Assgmt algorithm for solving the target assignment problem in dense environments. We showed that in the worst-case the time complexity is proportional to the area of the environment, and for uniformly randomly distributed agents and targets the complexity is proportional to diameter of the environment. There are many future research directions such as extensions to nonholonomic vehicles, to the case when targets are dynamically appearing and disappearing, or to dealing with collisions between agents. Another area of future research is to develop a communication framework which adequately models congestion and media access problems that are inherently present in wireless communications.

References

[1] J. E. Hopcroft and R. M. Karp, "An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs," *SIAM Journal on Computing*, vol. 2, no. 4, pp. 225–231, 1973.

[2] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics*, vol. 2, pp. 83–97, 1955.

[3] R. Burkard, "Selected topics on assignment problems," *Discrete Applied Mathematics*, vol. 123, pp. 257–302, 2002.

[4] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Belmont, MA: Athena Scientific, 1997.

[5] B. P. Gerkey and M. J. Mataric, "A formal analysis and taxonomy of task allocation in multi-robot systems," *International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.

[6] M. F. Godwin, S. Spry, and J. K. Hedrick, "Distributed collaboration with limited communication using mission state estimates," in *American Control Conference*, Minneapolis, MN, June 2006, pp. 2040–2046.

[7] M. Alighanbari and J. P. How, "Robust decentralized task assignment for cooperative UAVs," in *AIAA Conf. on Guidance, Navigation and Control*, Keystone, CO, Aug. 2006.

[8] C. Schumacher, P. R. Chandler, S. J. Rasmussen, and D. Walker, "Task allocation for wide area search munitions with variable path length," in *American Control Conference*, Denver, CO, 2003, pp. 3472–3477.

[9] B. J. Moore and K. M. Passino, "Distributed task assignment for mobile agents," *IEEE Transactions on Automatic Control*, vol. 52, no. 4, pp. 749–753, 2007.

[10] G. Arslan, J. R. Marden, and J. S. Shamma, "Autonomous vehicle-target assignment: A game theoretic formulation," *ASME Journal on Dynamic Systems, Measurement, and Control*, 2007, to appear.

[11] M. Zavlanos and G. Pappas, "Dynamic assignment in distributed motion planning with local information," in *American Control Conference*, New York, July 2007, pp. 1173–1178.

[12] S. L. Smith and F. Bullo, "Target assignment for robotic networks: Asymptotic performance under limited communication," in *American Control Conference*, New York, July 2007, pp. 1155–1160.

[13] ——, "Monotonic target assignment for robotic networks," Center for Control, Dynamical Systems and Computation. University of California at Santa Barbara, Tech. Rep. CCDC-07-0817, 2007, available electronically at `http://ccdc.mee.ucsb.edu`.

[14] R. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge, UK: Cambridge University Press, 1995.

[15] F. Xue and P. R. Kumar, "The number of neighbors needed for connectivity of wireless networks," *Wireless Networks*, vol. 10, no. 2, pp. 169–181, 2004.

[16] M. Penrose, *Random Geometric Graphs*, ser. Oxford Studies in Probability. Oxford, UK: Oxford University Press, 2003.