Competitive Coverage: (Full) Information as a Game Changer

Moshe N. Samson¹ and Noa Agmon²

Abstract—This paper introduces the competitive coverage problem, a new variant of the robotic coverage problem in which a robot \( R \) competes with another robot \( O \) in order to be the first to cover an area. In the variant discussed in this paper, the asymmetric competitive coverage, \( O \) is unaware of the existence of \( R \), which attempts to take that fact into consideration in order to succeed in being the first to cover as many parts of the environment as possible. We consider different information models of \( R \) that define how much it knows about the location of \( O \) and its planned coverage path. We present an optimal algorithm for \( R \) in the full-information case, and show that unless \( R \) has information about \( O \)'s initial location, it is as if it has no information at all. Lastly, we describe a correlation between the time it takes \( R \) to reach \( O \)'s initial location and the coverage paths quality, and present a heuristic algorithm for the case in which \( R \) has information only about \( O \)'s initial location, showing its superiority compared to other coverage algorithms in rigorous simulation experiments.

I. INTRODUCTION

The robotic coverage problem is one of the fundamental problems in robotic research, and as such has received considerable attention in the past two decades [10], [6]. The problem has its theoretical merits, but is of special interest due to its immediate applicability in real world settings, such as cleaning, coating, demining and search and rescue.

In the original problem of robotic coverage, a robot’s goal is to determine a path that will visit each point in a given area at least once, usually while minimizing the time for completion. In the multi-robot coverage problem, the coverage is a collaborative effort: each point in the area should be visited at least once by some robot from the team, and the common goal is to minimize the maximal working time of some robot from the team.

In this work we formally define a new variant of the coverage problem, competitive coverage, in which robots do not work collaboratively, but competitively. More formally, two robots, \( R \) and \( O \), are to cover a given area represented as a grid, and our goal is to maximize the number of cells \( R \) covers first, before they are covered by \( O \).

We examine in depth the asymmetric variant of the competitive coverage problem, in which \( O \) operates without the knowledge of \( R \)'s existence, and \( R \) knows it should compete with \( O \). The problem is modeled by the level of information \( R \) has on \( O \) (beside its existence): (i) \( R \) knows the initial location of \( O \) and its planned coverage path; (ii) \( R \) knows only the path, but does not know \( O \)'s initial location; (iii) \( R \) knows \( O \)'s initial location, but not its coverage path; (iv) \( R \) does not know either \( O \)'s path or its location.

We present an optimal algorithm for \( R \) in the full-information case. We also show that, surprisingly, having information only about the coverage path is equivalent to having no information at all about the opponent. Lastly, we present a heuristic algorithm for the case in which \( R \) has information only about the initial location of \( O \), and prove its superiority over other coverage algorithms based on rigorous empirical analysis using both a self-developed simulation, and the realistic ROS/Gazebo simulator.

¹This work was supported in part by ISF grant 2306/18
²Moshe N. Samson is with the Computer Science Department, Bar-Ilan University, Ramat Gan 5290002, Israel samson.moshe@gmail.com
Noa Agmon is with the Computer Science Department, Bar-Ilan University, Ramat Gan 5290002, Israel agmon@cs.biu.ac.il
II. BACKGROUND AND RELATED WORK

The problem of single-robot coverage has been extensively discussed in the literature. Refer to [10] for a recent survey of coverage path planning methods.

The coverage problem can be classified as either offline or online [6]. Online algorithms assume zero or partial knowledge regarding the world to be covered, and the coverage-path is generated while advancing in that world. Conversely, Offline algorithms rely on stationary, known beforehand map of the world, and thus create the full coverage-path before even starting to move through it. In this work, we focus on offline coverage.

The coverage problem has been reduced to the traveling salesman problem [4], and thus known to be \( NP \)-complete even on simple graphs such as grid graphs [14]. However, there are known solutions to the coverage problem that work even in linear time (e.g., Spanning tree coverage, STC, as presented in [9]). In our work, we consider an approximate cellular decomposition (as explained in [10]) into finite grid, and thus we know there exists an optimal coverage path that can be found in linear time.

Considerable attention has been given also to the multi-robot variant of the coverage problem, where multiple robots work in coordination in order to jointly cover an area. The robots can be with or without leader(s), relying on full or limited communication (e.g., [1]), in online or offline manner [1], [7]. In this work, we do consider multiple (exactly 2) robots, but working noncooperatively, one on each side.

Yehoshua et al. [20] recently introduced a new variant of the coverage problem, in which the covering robots operate in an adversarial environment, where threats exist and might stop the robot. Online algorithms for adversarial coverage were discussed in [18], and multi-robot algorithms for adversarial coverage were discussed in [19]. In this work, other robots considered as competitors, and are not being a threat to our robot.

Another worth-mentioning problem related to coverage is the patrolling task, in which the robot(s) are to repeatedly visit the area in order to monitor change in state. Examples to either partition-based or cyclic-based can be found in [11], [12], [13], [5]. Another one is adversarial patrolling ([2], [15], [3], where there is an adversary trying to penetrate through the patrol path, undetected. In this work, we consider competitors, where both are already in the area, and are trying to visit it as fast as possible.

Finally, the competitive problem is related to the foraging problem, which is searching and then transporting objects to one or more collection points. In [17] we find a fairly extensive survey of the subject. In our work, the robot does not need to find anything, therefore there is no notion of 'capacity' (that exists in foraging), and the choice to go back to certain points depends on the covering strategy assumptions (which, in our case, says that only position visited more than once is the initial position).

III. COMPETITIVE COVERAGE: DEFINITION

Let \( R \) and \( O \) be two robots operating in an obstacle-free grid \( W \) of size \( N = m \times n \). Both robots move in the four basic directions (North, South, East, West). Consider robot \( R \) to be our robot-of-interest, and robot \( O \) to be the opponent. The goal of each robot is to cover the area, that is, find a path (denoted as the coverage path) that visits each point in the area at least once. We define a coverage strategy of a robot as the coverage path, including the order of cells visited (specifically in a cyclic coverage path, the strategy indicates both the cells’ ordering, and the direction of movement—clockwise or counterclockwise). We denote \( R \)’s and \( O \)’s strategies by \( S_R, S_O \in S \), respectively, where \( S \) stands for the possible strategies space. In this paper we focus our attention to the offline version of the competitive coverage problem, in which \( S_R \) and \( S_O \) are deterministic, and computed in advance (before the execution), and thus consist of \( W \)’s cells permutation. For an online version, the strategy are random or deterministic, and may be adjusted during the execution based on the environment, the opponent’s behavior, random factors, and more. We leave the online version of the competitive coverage problem to future work.

Robot \( O \) is covering \( W \) using an optimal coverage
strategy, that is, it follows a path guaranteeing coverage in minimal time. Since solving the coverage problem is generally computationally hard [4], then for the sake of the analysis we focus on environments in which an optimal coverage path can be computed in polynomial time using the Spanning-Tree Coverage Algorithm [9] which generates cyclic coverage paths under some assumptions on the environment.

Robot R’s goal is to cover as many cells as possible before they are visited by R. Denote the number of cells in \( W \) first covered by robot \( R \) by \( FCC_R \). Therefore our goal is to find a coverage path for \( R \) that maximizes \( FCC_R \). When deciding between options with the same \( FCC \) value, \( R \) will choose the one that yields the fastest coverage time.

Denote the initial location of \( R \) by \( i_0 \). Robot \( R \) can be given \( i_0 \), \( S_0 \), both or neither. These types of information are called Information Models, and are defined as follows:

**Definition III.1 (Information Model).** Information Model \( IM \in \{ \emptyset, \{ S, I \} \} \), represents the knowledge a robot has on its opponent. \( S \in S \cup \{ S_\emptyset \} \), where \( S_\emptyset \) stands for an unknown strategy, and \( I \in W \cup \{ \emptyset \} \), where \( \emptyset \) refers to an unknown initial point. If \( IM = \emptyset \) then the player of interest does not know its opponent exists. Let \( IM_R \) be the information model \( R \) is given about \( O \), and let \( IM_O \) be the information model \( O \) is given about \( R \).

We assume that \( IM_R \neq \emptyset \), that is, \( R \) knows \( O \) exists. However, \( IM_O = \emptyset \), that is, \( O \) does not know \( R \) exists. This is referred to as asymmetric competitive coverage (we leave the symmetric version, in which \( O \) is aware of the existence of \( R \), to future work). Therefore, considering all said above, the (asymmetric) Competitive Coverage Problem is formally defined as follows.

**Competitive Coverage Problem**

Let \( W \) be a finite, obstacles-free grid of size \( N \). Given \( IM_O = \{ S_0, i_0 \} \) find \( S^*_R \in S \) s.t.

\[
S^*_R = \arg \max_{S_R \in S} \{ FCC_R(IM_R, IM_O) \}
\]

We examine the competitive coverage problem with the following information models:

1. **Full Information** - \( IM_R = \{ S_0, i_0 \} \)
2. **Partial Information** - \( IM_R = \{ S_\emptyset, i_0 \} \)
3. **Partial Information** - \( IM_R = \{ S_0, \emptyset \} \)
4. **Zero Information** - \( IM_R = \{ \emptyset, \emptyset \} \)

**IV. Motivation**

Consider the following real-world scenario: Two robots that are looking for oil over international waters, where the first one to discover it gets the rights of mining it. In this case, even though each side wants to cover the whole area as fast as possible, it is way more important to discover first as much of the area as possible.

In general, any case where there are scattered goods over an area in unknown locations, where the objective is to discover first as many of the goods as possible, is relevant to our case. Theorem 1 connects the FCC measure to the expected number of collected goods in this scenario.

**Theorem 1.** In a world \( W \) with unknown number of scattered items in unknown locations, and the probability for an item to exist in a cell is uniform throughout \( W \). Given two robots \( R \) and \( O \) that are trying to collect the same items, then \( R \) maximizing its FCC is equivalent to maximizing the expected number of collected items.

**Proof of Theorem 1.** Let \( g \) be and number of scattered items in \( W \), where \( g_i \) is item number \( i \), and let \( c(g_i) \in W \) be the cell containing \( g_i \). The expected number of items collected by \( R \) (over the locations of the items), denoted by \( S_G_R \), is defined as:

\[
E[S_G_R] = E \left[ \sum_{i=1}^{g} \mathbb{1}[R \text{ visits } c(g_i) \text{ before } O] \right]
\]

\[
= \sum_{i=1}^{g} E[\mathbb{1}[R \text{ visits } c(g_i) \text{ before } O]]
\]

\[
= \sum_{i=1}^{g} P(R \text{ visits } c(g_i) \text{ before } O)
\]

where \( \mathbb{1} \) is the boolean version heaviside function [16]. Since the probability for an item to exist in a cell is uniform throughout \( W \), that is,

\[
E[c(g_i)] = E[c(g_j)] \forall i, j \in [1, g]
\]
we get that:
\[
\max \{E[S_R]\} = \max \left\{ \sum_{i=1}^{g} P(\text{R visits } c(g_i) \text{ before } O) \right\} = \max \left\{ \frac{1}{|W|} \sum_{j=1}^{[|W|]} \mathbb{I}[\text{R visits } c_j \text{ before } O] \right\} = \max \left\{ \sum_{j=1}^{[|W|]} \mathbb{I}[\text{R visits } c_j \text{ before } O] \right\} = \max \{\text{FCC}_R\}
\]
Which concludes our proof.

\section{V. Full Information}

In this case robot R has full information about robot O’s plans, that is, \(I_{MR} = \{S_O, i_O\}\). We show that if R simply travels as quickly as possible to the first location in O’s path and precede it, R maximizes its FCC. This behavior is depicted in Algorithm ITP (Algorithm 2).

\textbf{Definition V.1} (Interception-Point). The \textit{Interception-Point} between \(S_O\) and \(i_R\) is the first cell \(c_j \in S_O\) that the time it takes R to reach \(c_j\) is lower than the time it takes O to reach it. The method for finding Interception-Point is shown in Algorithm 1. Notice we used \text{Dijkstra}(i_R, c_j)[8] to compute the distance between cells in the graph, which means ITP should work with obstacles too.

\begin{algorithm}
\caption{Finding Interception-Point}
\begin{algorithmic}
\Require \(i_R\)
\Require \(S_O = \{c_1^O, c_2^O, \ldots, c_N^O\}\)
\For \(j \in [1, N]\)
\If {Dijkstra \((i_R, c_j) < j\)}
\Return \(j\)
\EndIf
\EndFor
\end{algorithmic}
\end{algorithm}

In Theorem 3 we prove the optimality of ITP in the full information model, and show its expected FCC. In order to prove the expected FCC, we first prove the following supporting lemma.

\begin{algorithm}
\caption{Intercept Then Precede (ITP)}
\begin{algorithmic}
\Require \(i_R\)
\Require \(S_O = \{c_1^O, c_2^O, \ldots, c_N^O\}\)
\State \(k \leftarrow \text{Algorithm 1}(i_R)\)
\State \GoTo \(c_k^O\)
\For \(j \in [k + 1, N]\)
\State \GoTo \(c_j^O\)
\EndFor
\end{algorithmic}
\end{algorithm}

\begin{lemma}
The expected distance between two cells selected uniformly at random on a rectangular grid of size \(m \times n\) is
\[
\frac{m^2 - 1}{3m} + \frac{n^2 - 1}{3n}
\]
Proof of Lemma 2. Let \(X_1, Y_1, X_2, Y_2\) be random variables, indicating the coordinates for cell \(C_1 = \{X_1, Y_1\}\) and cell \(C_2 = \{X_2, Y_2\}\). \(X_1, X_2\) can fall anywhere in the range \([1 \ldots m]\), where \(Y_1, Y_2\) can fall anywhere in the range \([1 \ldots n]\). The expected distance between two cell is:
\[
\mathbb{E}[||C_2 - C_1||] = \mathbb{E}[||X_2 - X_1||] + \mathbb{E}[||Y_2 - Y_1||]
\]

The expression \(\mathbb{E}[||X_1 - X_2||]\) is computed in Equation (1). The expression \(\mathbb{E}[||Y_1 - Y_2||]\) is computed similarly in the range \([1, n]\), thus adding the two expressions concludes the proof.

\[
\begin{align*}
\mathbb{E}[||X_1 - X_2||] &= \sum_{x_1=1}^{m} \sum_{x_2=1}^{m} \frac{|x_1 - x_2|}{m^2} = \\
&= \sum_{x_1=1}^{m} \sum_{x_2=1}^{m} \frac{x_1 - x_2}{m^2} + \sum_{x_1=1}^{m} \sum_{x_2=x_1+1}^{m} \frac{x_2 - x_1}{m^2} = \\
&= \sum_{x_1=1}^{m} \frac{x_1}{2} \left( \frac{1}{2} (m - x_1) (m + x_1 + 1) - (m - x_1) x_1 \right) = \\
&= \sum_{x_1=1}^{m} \frac{1}{2} (x_1^2 - (1 + m) x_1 + \frac{1}{2} m^2 + \frac{1}{2} m) = \\
&= \frac{m^2}{2} + \frac{1}{2} m + \frac{1}{2} m^2 = \frac{m^2 - 1}{3m} (1)
\end{align*}
\]
\end{lemma}
Theorem 3. In the full knowledge asymmetric competitive coverage problem on an obstacle-free grid, Algorithm ITP optimizes $E[FCC]$, and in a grid of size $m \times n$ yields

$$E[FCC] = m \cdot n - \frac{m^2 - 1}{3m} - \frac{n^2 - 1}{3n}$$

Proof of Theorem 3. The FCC equals the number of cells robot $R$ visits before robot $O$. In a world of size $m \times n$ this equals the size of the world $(m \cdot n)$ minus the time it takes $R$ to reach the interception point of $O$’s coverage path. Therefore the expected FCC is $mn - E[|C_2 - C_1|]$. Following Lemma 2, this equals

$$E[FCC] = n \cdot m - \frac{m^2 - 1}{3m} - \frac{n^2 - 1}{3n}$$

We are now left to prove ITP’s optimality.

Remember that $O$ is said to be optimal, which means it does as less steps as it can, and in our case, it does exactly $|w|$ steps. In each step, $O$ is visiting a new cell. If it is the first to be there, it 'gains' the cell. Therefore, each step that $R$ is doing something else other than cover a new cell, $O$ is gaining a new cell. But, after interception, since $R$ is covering from that point using $S_0$, every step $R$ is taking is a guaranteed gain for $R$.

So, to minimize the amount of cells that $O$ is visiting before $R$ we intercept it (gaining steps along the way). From that point, by covering $w$ using $S_0$, $R$ maximizes its FCC. We just described ITP.

VI. Zero Information

In the zero-information case, $R$ knows neither $i_0$ nor $S_0$. In fact, in this information model, $R$ knows about $O$ only that it exists.

Let us introduce the Choose-Random-Strategy procedure (CRS), that chooses an optimal coverage path $S_R \in S$ at random. In Theorem 4 we prove the optimality of CRS, and its resulting $E[FCC]$. It follows that, in fact, the knowledge that an opponents exists in the world does not grant $R$ any advantage.

Theorem 4. In the zero-knowledge asymmetric competitive coverage problem on an obstacle-free grid, Algorithm CRS maximizes $E[FCC]$, and on a grid of size $m \times n$, CRS yields

$$E[FCC | S_R = CRS] = \frac{m \times n + 1}{2}$$

Proof of Theorem 4. Let us first introduce the notion of covering-time:

Definition VI.1 (Covering-Time $CT_R(c_i)$). The covering time of the cell $c_i$ by $R$ is the time it takes $R$ to reach cell $c_i$, for the first time. More formally, given $S_R = \{c_1^R, c_2^R, \ldots, c_N^R\}, CT_R(c_i)$ is the first index $j$ s.t. $c_j^R = c_i$.

Notice the following: a cell $c_i$ is 'gained by $R'$ if and only if $CT_R(c_i) < CT_O(c_i)$, which means that $R$ visits $c_i$ before $O$. Let $I[x]$ be the unity function, where $I[x] = 1$ if and only if $x$ is true, $I[x] = 0$ otherwise. One can re-write the expression for $R$’s gain using the $CT$ property:

$$E[FCC] = E \left[ \sum_{i=1}^{m \times n} I[CT_O(c_i) \geq CT_R(c_i)] \right]$$

To show that Equation (2) $= \frac{m \times n + 1}{2}$, we prove that $E[I[CT_O(c_i) \geq CT_R(c_i)]] = \frac{1}{2}$. Indeed:

$$E[I[CT_O(c_i) \geq CT_R(c_i)]] = P(CT_O(c_i) \geq CT_R(c_i)) = \frac{1}{2}$$

where the first equality can be easily proved, and the second is because when averaging over $i_0$ and $S_0$, $P(CT_O(c_i)) = P(CT_R(c_i)) = \frac{1}{m \times n}$, and since they are independent of each other, both can be considered as i.i.d variables, uniformly distributed over $[1, m \times n]$, and the probability that one is greater than the other (or, Equation (3)) is exactly $\frac{1}{2}$. Using Equation (2) and Equation (3) we get:

$$E[FCC] = \sum_{i=1}^{m \times n} E[I[CT_R(c_i) \geq i]] = \frac{m \times n + 1}{2}$$

□
VII. ONLY STRATEGY KNOWN

In this case, where \( R \) knows \( S_O \), but not \( i_O \), we examine whether \( R \) can achieve anything better than playing CRS, given that is is given more information: Unfortunately, as stated in Theorem 5, it cannot, and the best \( E[FCC] \) \( R \) can achieve is random-like. This result is surprising: the knowledge about \( S_O \) is irrelevant to \( R \), and it does not help achieving anything better than random-like results. That is, even though \( R \) has more information than in the zero-knowledge case, still no better results are achievable.

**Theorem 5.** When \( IM_R = \{S_O, i_O\} \), then

\[
\min_{S_R} \{E_{i_O}[FCC_R]\} = E_{i_O, S_R}[FCC_R] = \frac{N + 1}{2}
\]

**Proof of Theorem 5.** Since \( S_R \) and \( S_O \) are optimal-cyclic-coverage strategies, and since we assumed \( W \) is an obstacles-free rectangular grid, both \( S_R \) and \( S_O \) are actually Hamiltonian cycles, consisted of all the cells in \( W : c_0, ..., c_{N-1} \); The relative place a cell \( c_i \) is actually \( CT_R(c_i) \).

Note that each starting position \( i_r \) determines the covering time of all the cells \( c_0, ..., c_{N-1} \): Since we assume the strategy is known beforehand, then, for \( O \), the covering time is set after \( i_O \) is known, and changing it changes for all the cells their respective covering time. That is, \( CT(c_i) \) directly depends on \( i_O \) for all \( c_i \in W \), and \( CT_O(c_i) \in [0, N - 1] \).

Similar to Equation (2), one can write the FCC of a fixed problem (with all its variables known) \( FCC(W, S_R, S_O, i_r, i_O) \) as \#\{\( CT_R(c_i) \leq CT_O(c_i) \}\}. Let \( E_x(FCC) \) be the expected FCC where the randomness is taken over the variable \( x \). We therefore understand the following equation:

\[
E_{i_O}[FCC] = \frac{1}{N} \sum_{i_O \in W} \sum_{c_i \in W} 1 \{CT_R(c_i) \leq CT_O(c_i)\}
\]

If we change the order of summation, we can use what we know about ranging over the initial position and get:

\[
E_{i_O}[FCC] = \frac{1}{N} \sum_{i_O \in W} \sum_{c_i \in W} 1 \{CT_R(c_i) \leq CT_O(c_i)\}
\]

\[
= \frac{1}{N} \sum_{c_i \in W} \#\{CT_R(c_i) \leq CT_O(c_i)\}
\]

\[
= \frac{1}{N} \sum_{c_i \in W} N - CT_R(c_i) = \frac{N + 1}{2}
\]

where (*) is because \( CT_R(c_i) \) is not dependent on \( i_O \) (consider as constant), and the value of \( CT_O(c_i) \) ranges from 1 to \( N \). Combining the two, we get that there are exactly \( N - CT_R(c_i) \) different cases where \( CT_R(c_i) \leq CT_O(c_i) \).

\[ \square \]

VIII. ONLY INITIAL POSITION KNOWN

In this information model, where \( i_O \) is known but \( S_O \) is not, we present a heuristic algorithm for coverage, LTR (Longest To Reach), and demonstrate empirically its superiority over other algorithms in terms of maximizing FCC for \( R \). The optimality proof of LTR is left to future work.

**A. The LTR Algorithm**

The idea behind the LTR algorithm (Algorithm 3) is that for \( R \) to maximizes its expected FCC (over different \( O \)'s algorithms) it should cover areas with lower probability that \( O \) already visited, instead of areas with high such probability. Such covering algorithm \( S^* \) is the result of Equation (4).

To reach all the cells, we run BFS from \( i_O \), giving each cell a LEVEL value of how much recursive calls were created to reach that cell. After all the cells are set with LEVEL value, \( R \) tries to cover groups of cells, from high to low.

\[
S^* = \arg \min_{S = \{c_1, c_2, ..., c_k\}} \sum_{i=1}^{k} P[CT_O(c_i) < CT_R(c_i)]
\]

B. Simulations and Results

We have tested LTR in different settings. First, we ran simplified simulations of LTR and other strategies using python code. The world is a grid of
**Algorithm 3 Longest To Reach (LTR)**

**Require:** \( i_O \)
- set \( \text{STATUS}=\text{READY} \) for each cell in \( W \)
- set \( i_O.\text{STATUS}=\text{WAITING} \)
- set \( i_O.\text{LEVEL}=1 \)
- Enqueue \( i_O \)

while Queue not empty do
  \( c \leftarrow \text{Dequeue cell} \)
  set \( c.\text{STATUS}=\text{PROCESSED} \)
  let \( \text{maxLevel} := 1 \)
  for all \( c' \in c.\text{NEIGHBORS} \) do
    if \( c'.\text{STATUS}=\text{READY} \) then
      set \( c'.\text{STATUS}=\text{WAITING} \)
      set \( c'.\text{LEVEL}=c.\text{LEVEL}+1 \)
      \( \text{maxLevel} := c'.\text{LEVEL} \)
      Enqueue \( c' \)
    end if
  end for
end while

\( v = \text{maxLevel} \)

while \( \exists c \in W \text{ s.t. } c.\text{LEVEL}=v \) do
  cover all cells with \( \text{LEVEL}=v \)
  \( v = v - 1 \)
end while

size \( 32 \times 32 \), and it takes 1 step to travel between adjacent cells (north/south/east/west). They are called 'simplified' because no physical constraints were taken into consideration (turning time, collisions). Each algorithm was averaged over 100 different \( S_O \) random MST (Minimum Spanning Tree), and we checked four different cases for \( i_O \) (\( i_R \) is always as \( (0,0) \)). We compared our algorithm with five different coverage algorithms, all of them are optimal (take exactly \( |W| \) steps to cover the world), as follows. LCP algorithm is the opposite of LTR: it covers the world starting from \( i_O \) and advance on increasing \( \text{LEVEL} \) values. MST is a simple random MST coverage path, averaged over 30 randomly chosen MST paths. CircVert is circular covering path that prefers vertical movements. CircHorz is defined similarly for horizontal movements. Non-CircVert is as CircVert but without the circularity constraint (that is, the path is not circular). The results are shown in Figure 2. As one can see, in three out of four times, LTR yields the best results compared to the other algorithms (statistically significant, using Student t-test with p-value < 0.0005). Even in the one case where LTR is only second to optimal, one should look at the huge error margin for the winning algorithm and the much lower error margin for LTR (error bars are standard deviation over the samples).

We have also examined LTR in a realistic simulations using ROS-GAZEBO. We used standard turtlebots with radius of 0.35 meters, and the world is of size \( 11.2 \times 11.2 \) meters, and can be thought of as \( 32 \times 32 \) grid with cells the size of one turtlebot. The results are shown in Figure 3. The results clearly support the ones we got from the simplified simulations.
IX. CONCLUSIONS AND FUTURE WORK

In this paper we presented the competitive coverage problem, in which two robots exist in an environment and compete to be the first to cover cells. We have examined in depth the asymmetric case, in which only one robot is aware it is in a competition with the other, and suggested solutions based on different information models the robot holds on its opponent. We have shown that only having full knowledge on the opponent’s strategy has a significant impact on the possibility of winning. There are still many directions to pursue in the future, among those examining the symmetric case, proving optimality of the LTR algorithm, and examining an online version of the problem.

REFERENCES


