MRSAM: A Quadratically Competitive Multi Robots Navigation Algorithm

Shahar Sarid and Amir Shapiro
Department of Mechanical Engineering
Ben Gurion University of the Negev, Israel
{sarids,ashapiro}@bgu.ac.il

Yoav Gabriely
Department of Mechanical Engineering
Technion, Israel Institute of Technology
meeryg@tx.technion.ac.il

Technical Report

Abstract

We explore an on-line problem where a group of robots has to find a target whose position is unknown in an unknown planar environment whose geometry is acquired by the robots during task execution. The critical parameter in such a problem is the physical motion time, which, under the assumption of uniform velocity of all the robots, corresponds to length or cost of the path traveled by the robot which finds the target. The *Competitiveness* of an online algorithm measures its performance relative to the optimal off-line solution to the problem. While competitiveness usually means constant relative performance, this paper uses generalized competitiveness, i.e. any functional relationship between online performance and optimal off-line solution. Given an online task, its *Competitive Complexity Class* is a pair of lower and upper bounds on the competitive performance of all online algorithms for the task, such that the two bounds satisfy the same functional relationship. We classify a common online motion planning problem into competitive class. In particular, it is shown that navigation to a target whose position is recognized only upon arrival belongs to a quadratic competitive class. This paper describes a new on-line navigation algorithm, called *MRSAM* , which requires linear memory and has a quadratic competitive performance. Moreover, it is shown that in general any on-line navigation algorithm must have at least a quadratic competitive performance. The *MRSAM* algorithm achieves the quadratic lower bound and thus has optimal competitiveness. The algorithm is improved with some practical speedups and its performance is illustrated in office-like environments.

I. Introduction

The Problem of finding a target whose position is unknown in an unknown planar environment is very important in many practical and academic research fields, the most significant are humanitarian robotics, industry robotics and military robotics. Area coverage is a corresponding task, since the searching unit will cover a certain area before finding the target[1], and when the number of targets in a specific area is unknown, the whole area must be covered. Examples for the problems above are demining, search/rescue missions, cleaning supermarkets and train stations, detecting contaminated or radioactive substances in factories, nuclear reactors or in the open field, planetary exploration and sample acquisition. This paper is concerned with the aforementioned problem solved by multiple mobile robots.

The most critical parameter in mobile robot motion tasks is the physical travel time rather than on-board computation time. Under a uniform velocity assumption, travel time corresponds to path length, and under the assumption of uniform velocity among all the robots, travel time corresponds to the path length of the robot that found the target, or of any other robot that terminated at the same time the target was found, since all the robots travel the same path length per time unit. We denote the distance traveled by each robot, l, and the optimal off-line solution l_{opt}. Under a uniform power output assumption, travel time corresponds to path traversibility cost. Hence the algorithm discussed in this paper is classified in terms of length or cost of the path traveled by one robot during algorithm execution.

In the problem discussed above, using a group of robots can have many advantages over using only one robot, the most important is the shortening of the total mission time, another advantage is increased robustness, since the multitude of robots can easily overcome a malfunction in one or more of the units, an issue associated with redundancy. The decrease of the individual mechanical wear and power consumption per mission maximizes the life span of each robot and prolongs the whole mission duration and range. Other advantages concerns maintaining radio connectivity between the robots and the base station. Another advantage of robot groups is a decreased sensor uncertainty due to merging of overlapping information, it has been shown in [1], that multiple robots localize themselves more efficiently, especially when they have different sensor capabilities.

The purpose of this paper is to introduce a new algorithm, *MRSAM* (short for *Multi Robot Search Area Multiplication*) to solve the problem of finding a target whose position is unknown in an unknown planar environment with multiple robots, and to prove optimality of the *MRSAM* algorithm. This is done by proving that the problem itself belongs to the quadratic competitive complexity class and that the performance of *MRSAM* belongs to that class. The notion of competitiveness compares the performance

[1]When dealing with limited sensors which detects the target upon arrival or within a specific constant range from the target which is much smaller in magnitude than the typical length of the problem's environment.

of an on-line algorithm to the optimal off-line solution for the same problem. In particular, an algorithm for a task P is said to be competitive if its solution to every instance of P is bounded by a constant time l_{opt} [2]. Generalized *Competitive Complexity* and *Competitive Complexity Classes* are introduced and discussed in [3], however, most of the papers dealing with competitiveness strive to identify specific classes of environments in which constant competitiveness can be achieved. In contrast, our objective is to identify the competitive relationship governing the fully general on-line navigation problem for multiple robots. *MRSAM* is based on the area doubling strategy of $SAD1$ algorithm introduced in [3], which launches one robot to search for a target whose position is unknown in an unknown environment. $SAD1$ assigns a search disc to the robot, which is doubled at each step, if the target is not found.

Recent works related to the subject of mobile multi-robots motion planning deals with many aspects of the problem. A major issue is whether the group architecture is centralized or decentralized, i.e. wether there is only one control agent, or not. In the second case each robot is autonomous and there is neither a centralized component, nor any other global coordination needed. Communication is a very close subject, since, when there is no communication between the robots or when it is limited, the system cannot be centralized. Intermediate systems represent real-world setups better, for example, the semi-decentralized approach in [4], where robot teams cover the space independent of each other, but robots within a team communicate state and share information. Limited communication plays an important role when dealing with ant-like robots, where messages between the robots are passed mainly or only through marking they leave on the terrain, [5]. A solution to a problem can change according to the availability of information on the environment prior to algorithm execution. Online solutions assume no knowledge of the environment when the algorithm starts, while off-line solutions rely on apriori knowledge. An off-line algorithm is presented in the notable early paper [6], and a new work in that area [7] focuses on robustness, and completeness of the algorithm. Robustness measures the performance in case of failures and an algorithm is considered complete if for any input it correctly reports whether or not there is a solution in a finite amount of time. A limited-communication complete algorithm is presented in [8]. Our solution is complete and robust and can be decentralized or centralized, depending on the setup communication.

The structure and contributions of the paper are as follows. In the next section we state a key assumption that the robot has a physical size D such that $D > 0$. While this assumption may seem obvious, only few papers make use of this assumption (e.g. [9], [3]). We also present some definitions regarding competitiveness. In Section III we show that for every on-line algorithm, there is a worst case path that yields a cost which is cl_{opt}^2. The *MRSAM* algorithm is presented in Section IV and its competitiveness is analyzed in Section V. It is shown that the length of the path traveled by the robot during execution of *MRSAM* is at most quadratic in l_{opt}, implying that up to the constant coefficients *MRSAM* has optimal competitiveness. In the same Section (Sec. V) *MRSAM* is proved to be complete. Simulation results of *MRSAM* execution in an office-like environment are described in Section VI .An extension version of the algorithm handling multi-target environment is displayed in Section VII along with some practical speedups of the algorithm. Finally, we conclude and discuss additional research directions and future work.

II. Basic Setup and Definition of Competitiveness

Our basic assumptions are as follows. Each mobile robot is a freely moving planar body of size D, where $D > 0$ is a given constant. One may think of the mobile robots as discs of diameter D. Each robot is equipped with three sensors which are assumed ideal. The first sensor measures the robot position with respect to a fixed reference frame. The second sensor is an obstacle detection tactile or short range sensor which allows tracing of an obstacle boundary. The third sensor is a target recognition sensor. The robots use onboard and/or central calculation unit and can communicate with each other and/or with a central base station, at least upon starting and ending of the execution. The robots or the base station are assumed to have enough memory for the calculations needed and they all move in the same uniform velocity.

Next we describe the parameters governing the performance of mobile robot tasks. The three most significant parameters are physical travel time, on-board computation time, and on-board memory. In order to simplify the ensuing analysis, we associate physical travel time with length l of the path traveled by the robot. As for on-board computation time, we limit our discussion to algorithms that take polynomial time to compute each physical motion step of the robot. Since the time required for a physical motion step is typically several orders of magnitudes longer than the execution time of an on-board computation step, we focus on l as the main performance parameter. Last, we limit the discussion to algorithms whose storage requirement is at most linear in the size of the environment. This memory requirement may prove impractical in tasks such as planetary navigation, and may need to be revised in future work.

Thus l denotes length of the path traveled by the robot while l_{opt} denotes length of the optimal off-line path. The following definition generalizes the traditional notion of linear competitiveness to any functional relationship between l and l_{opt}.

Definition 1 (generalized competitiveness): An on-line algorithm solving a task P is $f(l_{opt})$- competitive when l is bounded from above by a scalable function $f(l_{opt})$ over all instances of P. In particular, $l \leq c_1 l_{opt} + c_0$ is the traditional linear competitiveness, while $l \leq c_2 l_{opt}^2 + c_1 l_{opt} + c_0$ is quadratic competitiveness, where the c_i's are positive constant coefficients that depend on the robot size D.
The meaning of scalability is as follows. When performance is measured in physical units such as meters m, one must ensure that both sides of the relationship $l \leq f(l_{opt})$ posses the same units, so that change of scale would not affect the bound . For instance, the coefficient c_2 in the relationship $l \leq c_2 l_{opt}^2 + c_1 l_{opt} + c_0$ must have units of m^{-1}, c_1 must be unitless, and c_0 must have units of m. Note that the definition of $f(l_{opt})$-competitiveness focuses on a particular algorithm solving the task P. However, our objective is to characterize the lowest upper bound that can be achieved over all on-line algorithms for P. This objective requires a universal lower bound on the performance of all on-line algorithms for P. If the lower and upper bounds satisfy the same functional relationship, we associate the functional relationship with P itself. This notion is made formal in the following definition.

Definition 2 (Competitive Complexity Class): A universal lower bound on the competitiveness of a task P is a lower bound $l \geq g(l_{opt})$ over all on-line algorithms for P. If a competitive upper bound $f(l_{opt})$ and a universal lower bound $g(l_{opt})$ for P are the same function up to constant coefficients, this function is the competitive complexity class of P.

The competitive complexity class of a task P is thus a pair of lower and upper bounds on the competitive performance of all on-line algorithms for P, such that the two bounds are identical up to constant coefficients. Note that competitive complexity characterizes the task P itself, not of any specific algorithm for P. The remainder of the paper characterizes the competitive complexity class of the multi-robots on-line navigation problem.

III. UNIVERSAL LOWER BOUND

In this section we establish a universal lower bound on the competitive complexity for the problem of navigation to a target which is recognized only upon arrival by a group of robots. The environment that serves to establish the lower bound is a disc containing D-width corridors that emanate radially from the start point S (Figure 1). Initially a small disc centered at S is free of obstacles. At a certain distance from S eight equally spaced point-size obstacles appear, such that the distance between the obstacles is D (the number eight has no special meaning here). The eight obstacles extend radially as lines and form the boundary of eight passable corridors for the robot. The width of the eight corridors increases as they stretch radially away from S. When the width of a corridor becomes $2D$, the corridor splits into two D-width corridors separated by a cone-shaped obstacle (Figure 1(b)). By symmetry all corridors split simultaneously. Hence the cone obstacles that separate corridors just before splitting are truncated at the splitting radius, and become radial lines from this radius onward. A close inspection of Figure 1(b) reveals that the cone obstacles occupy one third of the disc's total area. Finally, the tip of each cone obstacle is symmetrically ”shaved”, so that its tip would lie at a distance D away from the truncated cone lying closer to S (Figure 1(b)). This shaving allows a D-size robot to enter the two corridors generated by splitting. Note that the shaved off area becomes negligible relative to the cone's total area as the radius of the environment increases.

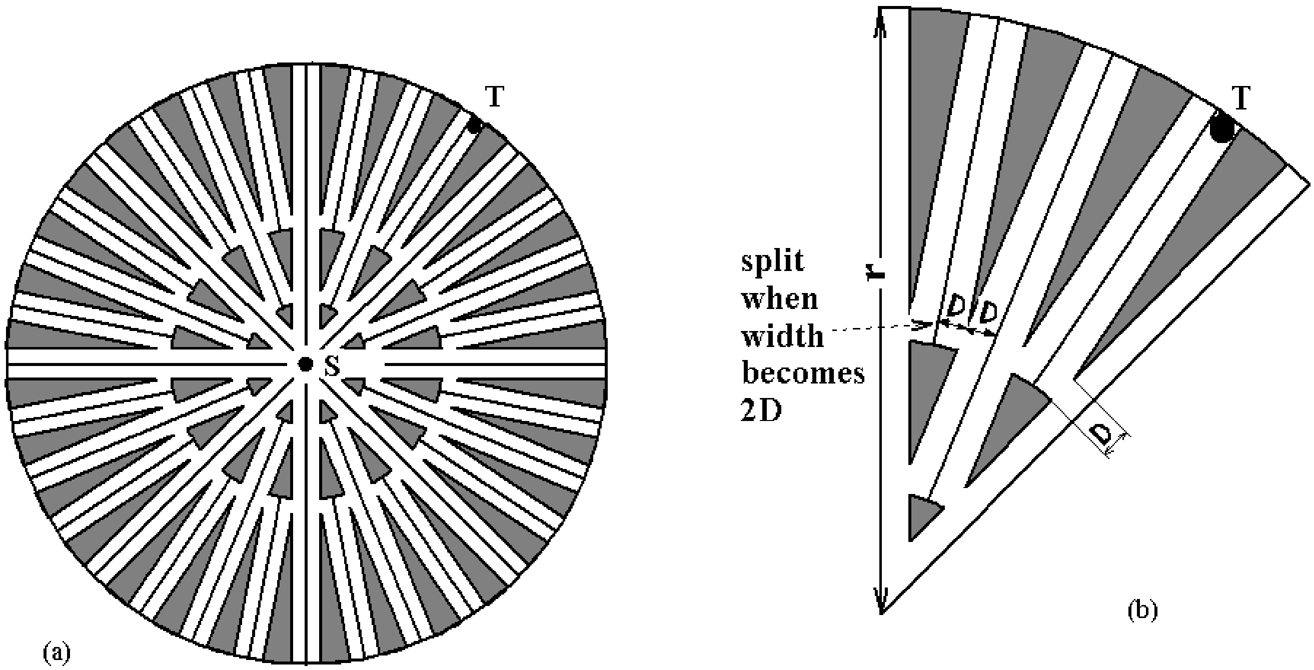

Fig. 1. (a) The radial corridors environment. (b) Close up view of the environment.

The following theorem establishes a quadratic lower bound on the performance of all on line navigation algorithms for a group of robots to a target whose position is recognized only upon arrival.

Lemma 3.1: Let A be any navigation algorithm for n robots in an unknown planar environment to a target whose position is recognized only upon arrival. Let l be the length of the path generated by A for one of the robots, and let l_{opt} be the length of the optimal off-line path. Then l satisfies the quadratic lower bound,

$$l \geq \frac{4\pi}{3nD}(1-\epsilon)l_{opt}^2$$

where D is the robot size and ϵ is an arbitrary small positive parameter.

Proof: Consider the corridor environment with the target T placed at the end of a distal corridor, at a distance r from S. Since the robots have no knowledge of the environment and has no information where T might lie, they must in worst case inspect every corridor including all distal corridors. (If A is deterministic, we can enforce this worst case scenario by first watching the behavior of A, then placing the target in the last inspected corridor. If A is non-deterministic, we can only guarantee that one outcome of the algorithm would match this worst case scenario.) By construction every distal corridor can be approached from S along a simple radial path. Moreover, the robots must eventually move twice through every corridor of the environment - once in order to inspect a distal corridor and once in order to exit the corridor. An exception to this rule is the last corridor which is considered below. The total area of the obstacles in the corridor environment is almost one third of the disc area, with the approximation becoming arbitrary close to one third as the disc's radius increases. The total area inspected by the robots is therefore $2\pi r^2/3$. Since all corridors have a width D which is identical to the robots size, the total length of the path traveled by the robots satisfies in worst case the inequality $l_{tot} \geq 4\pi r^2/3D - r$, where the substraction of r is due to the last corridor which need not be traced backward. In the best case where none of the robots travel any part of the path of the other robots, the total length of the path traveled by one robot satisfies $l \geq 4\pi r^2/3nD - r/n$. Since T is placed at a radial distance r from S we have that $l_{opt} = r + \epsilon'$, where ϵ' is an arbitrary small positive parameter. Substituting for r gives $l \geq (4\pi/3nD)l_{opt}^2 - l_{opt}/n - \epsilon'$ .We can write the last inequality as $l \geq l_{opt}^2(c - (l_{opt}/n + \epsilon')/l_{opt}^2)$, where

$c = 4\pi/3nD$. Since the quantity $\epsilon = (l_{opt}/n + \epsilon')/l_{opt}^2$ can be made arbitrarily small for sufficiently large environments, we obtain the lower bound $l \geq c(1-\epsilon)l_{opt}^2$. ■

IV. MRSAM ALGORITHM

MRSAM algorithm launches multiple robots from a common starting point S and assigns each robot j to a disc to search for the target T in it, all the discs are concentric and S is their center. The first robot

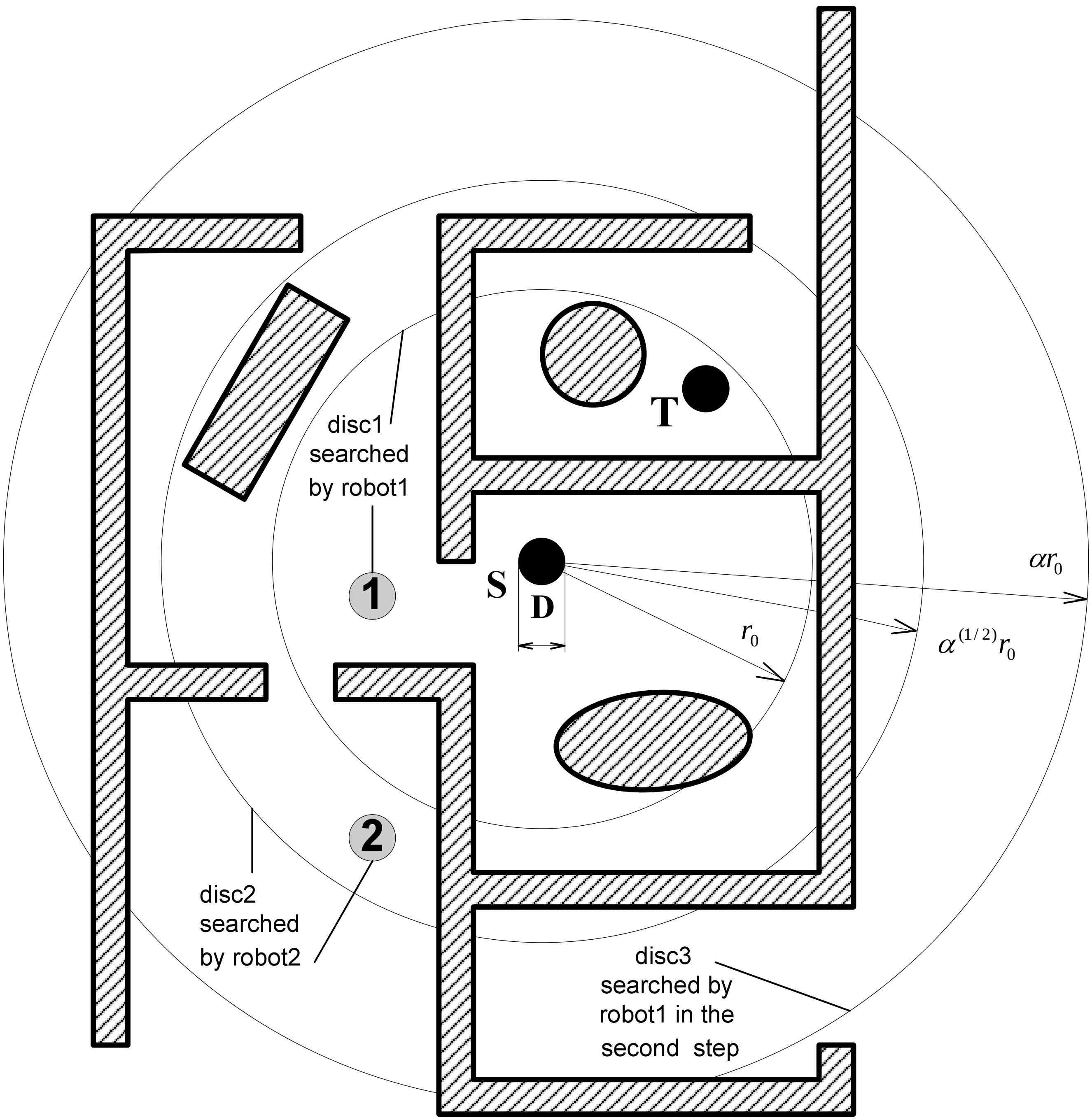

Fig. 2. A group of two robots launched by *MRSAM* searching for the target.

$(j = 1)$ is designated to the initial disc of area S_0, and each of the following robots starts its search in a disc of area larger than the previous disc by a factor of [2] $\alpha > 1$, namely, the areas of the discs will be S_0,αS_0,$\alpha^2 S_0$,$\alpha^3 S_0 \ldots$. For example, in Figure 2, robot 1 is initially assigned to search for the target inside a disc of area S_0 and robot 2 is assigned to search inside a disc of area αS_0. After robot 1 completes covering the entire portion of disc 1 which is accessible from S, it starts searching for the target inside disc 3 of area $\alpha^2 S_0$, in this case, robot 1 will find the target while searching in disc 3, before or after robot 2 completed searching in disc 2 and moved on to disc 4 of area $\alpha^3 S_0$. Each robot searches for the target in the accessible portion of the disc allocated to him until the target is detected, or until the entire region accessible from S is explored without finding T. The search process in each disc is as follows. The robot imposes an on line discretization of the continuous area into a grid of D-size cells [10], [11]. The grid consists only of free cells and is surrounded by partially occupied cells. The robot executes a standard area coverage tour on the grid of free cells, while scanning each new cell for the target. Once entering a new cell, the robot additionally scans the neighboring partially occupied cells for T. If the discretization preserves the connectivity of the accessible region (this assumption can be relaxed by a more sophisticated algorithm that monitors local connectivity breakage), then clearly all free and partially occupied cells in the region accessible from S are eventually inspected by the robot.

The robots cover the area of the discs until they reach the Target. If the target was not detected in the initial disc, the robot is assigned to the next non-occupied disc. A formal description of the algorithm follows.

Basic MRSAM Algorithm:

Sensors: A position and orientation sensors. An obstacle detection sensor.
Input: A Start point S, An initial Search Radius r_0, A group of n searching robots $\{R_1, R_2, \ldots, R_n\}$.
Initialization: For each robot R_j, $j = 1, \ldots, n$:
Set Multiplication factor $\alpha = (n+1)^{1/n}$.
Set current search disc $p_j = j$,
Set initial search radius[3] $r_j(p_j) = \alpha^{\frac{p_j - 1}{2}} r_0$.
For each robot j, Repeat:

1) Execute a coverage tour on the grid contained in the disc of radius $r_j(p_j)$ with center at S.
   a) Scan each new free cell and its partially occupied neighbor cells for T.
   b) STOP if T is found.
2) If no new free cell is encountered during the i^{th} coverage tour: STOP, the target is unreachable.
3) Set $p_j = p_j + j$,
   Set $r_j(p_j) = \alpha^{\frac{p_j - 1}{2}} r_0$

(End of Repeat loop)

Rather than give a formal proof of correctness, we make some informal remarks on the algorithm. First, during the initialization section, after getting the values of n and r_0, each robot can calculate its future search discs and the corresponding radii, which means that the robots does not need to communicate with each other till the rest of the execution, apart from a stopping signal when the target is found. This implies a decentralized approach with no or limited communication. Second, a robot that finished searching a disc will immediately proceed to the next disc assigned to it. The coverage tour in step IV can be executed with a trivial DFS algorithm using linear memory. The algorithm's average performance can be improved if each robot covers in each step only those cells which lie in the ring added to the previous disc, this is discussed in detail later in this paper. Fourth, if the target is inaccessible from S, the algorithm would stop only when it has completely covered the connected component of the environment containing S. A detailed example of *MRSAM* execution appears in Section VI.

[2]This is an important property, since the search area must be extended in each step in order to reach the target when the target is positioned outside the first search disc.

[3]$r_j(p_j) = \alpha^{\frac{p_j - 1}{2}} r_0$ is the radius of the i^{th} disc which is assigned to robot j.

V. Competitive Complexity Analysis of MRSAM

We now establish an upper bound on the path length of *MRSAM* in terms of l_{opt}. The following proposition establishes a quadratic competitive upper bound on *MRSAM* .

Proposition 5.1: If the target T is reachable from S, *MRSAM* finds the target using n robots and the path length l_j traveled by the robot which found the target satisfies the quadratic inequality,

$$l_j < \frac{2\pi\alpha^{n+1}}{D(\alpha^n - 1)} l_{opt}^2 + \frac{2\pi r_0^2}{D}$$

where D is the robot size, r_0 is the initial search radius, α is the multiplication factor which is a function of n only, and l_{opt} is the length of the optimal off-line path from S to T. Note that the upper bound is scalable, in the sense that both summands have units of length.

Proof: First consider the case where $l_{opt} > r_0$. In this case the initial search disc area is expanded at least once before the target is found. Suppose the search disc is expanded $i-1$ times until the target is found (in disc i). Since the radius of a disc is increased by a factor of $\sqrt{\alpha}$ at each step, its area increases in each step by a factor of α. Let $S_0 = \pi r_0^2$, and let S_j denote the total area of the regions inspected by robot j which found the target after searching in k discs[4] (these areas include free as well as partially occupied cells inspected by the robot.), then S_j is bounded by:

$$\begin{aligned} S_j &\le S(j) + S(j+n) + S(j+2n) + ... + S(i) = \\ &= \alpha^{j-1}S_0 + \alpha^{j-1+n}S_0 + \alpha^{j-1+2n}S_0 + ... + \alpha^{(i-1)}S_0. \end{aligned}$$

Substituting $i = j + n(k-1)$ yields

$$\alpha^{(i-1)}S_0 = \alpha^{j-1+n(k-1)}S_0.$$

And thus we get:

$$\begin{aligned} S_j \le& \alpha^{j-1}S_0 + \alpha^{j-1+n}S_0 + \alpha^{j-1+2n}S_0 + \dots \\ & \dots + \alpha^{j-1+n(k-1)}S_0 = \\ =& \alpha^{j-1}S_0[(\alpha^n)^0 + (\alpha^n)^1 + (\alpha^n)^2 + \dots + (\alpha^n)^{k-1}] = \\ =& \alpha^{j-1}S_0\frac{(\alpha^n)^k - 1}{\alpha^n - 1} = S_0\frac{\alpha^{j-1}(\alpha^n)^k - 1}{\alpha^n - 1} \le \\ \le& S_0\frac{\alpha^{j-1}(\alpha^n)^k}{\alpha^n - 1}, \end{aligned}$$

where we used $q = \alpha^{j-1}$, $\lambda = (\alpha^n)$, and $w = k$ in the formula:

$$q + q\lambda + q\lambda^2 + q\lambda^3 + ... + q\lambda^{w-1} = \frac{q(\lambda^w - 1)}{(\lambda - 1)}.$$

Since the disc of radius l_{opt} already contains at least one path from S to T, *MRSAM* finds the target in a disc of radius at most $\sqrt{\alpha}l_{opt}$. It follows that the area of the i^{th} search disc, $\alpha^{i-1}S_0$, satisfies the inequality $\alpha^{i-1}S_0 < \alpha\pi l_{opt}^2$. Substituting this inequality into the bound on S_j gives:

$$\begin{aligned} S_j <& \frac{\pi\alpha^j(\alpha^n)^k}{(\alpha^n - 1)\alpha^{i-1}} l_{opt}^2 = \frac{\pi\alpha^j\alpha^{nk}}{(\alpha^n - 1)\alpha^{j+n(k-1)-1}} l_{opt}^2 = \\ =& \frac{\pi\alpha^j\alpha^{nk}}{(\alpha^n - 1)\alpha^j\alpha^{nk}\alpha^{-n}\alpha^{-1}} l_{opt}^2 = \frac{\pi}{(\alpha^n - 1)\alpha^{-n}\alpha^{-1}} l_{opt}^2 = \\ =& \frac{\pi\alpha}{1 - \alpha^{-n}} l_{opt}^2 = \frac{\pi\alpha^{n+1}}{\alpha^n - 1} l_{opt}^2 \end{aligned}$$

$$S_j < \pi\frac{\alpha^{n+1}}{\alpha^n - 1} l_{opt}^2 \quad (1)$$

[4] k can be considered a global step. In the initial global step the n robots search in the first n consequent discs, in the next global step the robots search in the next n consequent discs and so on.

Now recall that *MRSAM* guides the robot only through free grid cells. The total area of the free grid cells is at most S_j. Hence the total number of grid cells visited by the robot (including repetitive visits), denoted m, is bounded by $m \leq S_j/D^2$. On line coverage algorithms such as DFS guide the robot along a path whose length in cells is at most twice the total number of grid cells. Since each cell has size D, the total length of the path traveled by the robot is $l_j \leq 2mD$. Thus,

$$l_j \leq 2mD \leq \frac{2}{D}S_j < \frac{2\pi\alpha^{n+1}}{D(\alpha^n-1)}l_{opt}^2 = C_n\frac{2\pi}{D}l_{opt}^2$$

where we substituted the inequality $S_j < \frac{\pi\alpha^{n+1}}{\alpha^n-1}l_{opt}^2$, and $C_n = \frac{\alpha^{n+1}}{\alpha^n-1}$ is constant per execution since it depends on n alone. Finally, the constant term $2\pi r_0^2/D$ bounds the path length traveled by the robot in the case where $l_{opt} \leq r_0$. In this case the target is found inside the initial search disc of radius r_0. ■

The following lemma, inspired by [12], asserts that search area multiplying is indeed an optimal strategy.

Lemma 5.2: The Competitive Complexity of *MRSAM* is minimal when the multiplication factor α equals $\alpha = (n+1)^{1/n}$.

Proof: Let n be the number of robots searching for the target, and suppose the target was found in the i^{th} disc by robot number j after covering that disc entirely. The area of the x^{th} disc is $S(x) = \alpha^{x-1}S_0$ where α is the area multiplying factor and $S(1) = S_0$. The total area S_j covered by that robot as obtained in Eq.(1) is:

$$S_j < \pi C_n l_{opt}^2, \text{ where } C_n = \frac{\alpha^{n+1}}{\alpha^n-1}$$

Minimizing for α while taking into account $\alpha > 1$ as explained before, and $n \geq 1$ for realistic execution,

$$\frac{\partial}{\partial\alpha}(C_n) = \frac{\partial}{\partial\alpha}\left(\frac{\alpha^{n+1}}{\alpha^n-1}\right) = \frac{\alpha^n(\alpha^n-n-1)}{(\alpha^n-1)^2}$$

Equating with zero yields

$$\alpha = (n+1)^{1/n}.$$

This is an extremum value, a second derivative will check the minimality of α,

$$\frac{\partial^2}{\partial\alpha^2}(C_n) = \frac{\partial^2}{\partial\alpha^2}\left(\frac{\alpha^{n+1}}{\alpha^n-1}\right) = \\ = \frac{(\alpha^{2n}-\alpha^n-n\alpha^n)(2n\alpha^{2n-1}-2n\alpha^{n-1})}{(\alpha^n-1)^4} \quad (2)$$

Since $\alpha > 1, n \geq 1$, which implies $\alpha^n - 1 > 0$, the denominator of Eq. (2) is always positive, thus the numerator determines the sign. Let E denote the numerator. Simplification of E yields,

$$E = 2n\alpha^{2n-1} + \left(n^2+n\right)\alpha^{n-1}\left(\alpha^{2n}-1\right)$$

Since $\alpha > 1, n \geq 1$ and $\alpha^n - 1 > 0$, All the terms in the righthand side of the equation above are positive and therefore $E > 0$, which implies that l_j gets minimal values when $\alpha = (n+1)^{1/n}$. ■

Corollary 5.3: MRSAM is complete.

Proof: The first important property established in Proposition 5.1, is that if the target T is reachable, *MRSAM* will find it. The second property is that *MRSAM* will find the target in a finite and limited time and is deduced from the bound on the path length introduced in Proposition 5.1. ■

In order to compare the performance of *MRSAM* running more than one robot with the performance of other algorithms running only one robot, we will compare the upper bound on the path length l_j of the robot that found the target for *MRSAM* with multi-robot execution $(n \to \infty)$ and for *MRSAM* execution with only one robot $(n = 1)$. This is done by calculating C_n for the two cases above. First, for the case where $n \to \infty$, it can easily be shown that α goes to 1,

$$\lim_{n\to\infty}(n+1)^{\frac{1}{n}} = 1$$

and thus, C_n approaches 1, as well.

$$\lim_{n\to\infty} C_n = \lim_{n\to\infty}\frac{\alpha^{n+1}}{\alpha^n - 1} = \lim_{n\to\infty}\frac{n+1}{n} = 1$$

On the other hand, for the second case, where $n = 1$,

$$\alpha = (1+1)^{1/1} = 2$$

therefore,

$$C_n = \frac{2^{1+1}}{2^1 - 1} = 4$$

And thus,

$$l < \frac{8\pi}{D} l_{opt}^2 + \frac{2\pi r_0^2}{D}$$

The last result coincides with previous results of an optimal algorithm for the same problem with one robot [3]. It can immediately be seen that when $n \to \infty$, *MRSAM* performs 4 times faster than the optimal algorithm which solves the same problem using one robot. It should be noted that for the constraints $\alpha > 1$ and $n \geq 1$ mentioned above, α is a monotonic rising function and thus C_n is a monotonic rising function, as well.

Some more values of α and C_n for several cases of n are shown in Table I. When using one hundred robots, C_n approaches one, and *MRSAM* multiplies the performance compared to execution with one robot by a factor of 3.78. Using 4 robots, *MRSAM* doubles the performance and with 13 robots and above it triples the performance compared to one robot execution.

TABLE I

SOME α AND C_n VALUES CORRESPONDING TO n ENTRIES

n	α	C_n
1	2	4
2	1.732	2.598
4	1.495	1.869
13	1.225	1.319
100	1.04	1.058

Theorem 1: Quadratic competitive complexity class
The online multi-robots navigation problem belongs to the quadratic competitive complexity class.

Proof: A competitive complexity class, as defined in Definition 2, is formed from two bounds, lower and upper bounds on the competitiveness of a task. According to Lemma 3.1, the lower bound of the problem discussed above has a quadratic-competitive complexity and is

$$l \geq \frac{4\pi}{3nD}(1-\epsilon) l_{opt}^2$$

Since the upper bound of *MRSAM* , as demonstrated in Proposition 5.1, is also quadratic in l_{opt},

$$l_j < \frac{2\pi\alpha^{n+1}}{D(\alpha^n - 1)} l_{opt}^2 + \frac{2\pi r_0^2}{D}$$

this navigation problem belongs to the quadratic competitive complexity class. ■

VI. SIMULATION RESULTS

In the following example *MRSAM* algorithm launches 4 robots from a starting point S to search for the target T in an unknown office-cubical environment[5] as depicted in figure 4. The area multiplication factor for $n = 4$ is $\alpha = 1.495$, and $\sqrt{\alpha} = 1.223$, and in the initial global step $k = 1$ (local step $i = 1, 2, 3, 4$) each robot is assigned to one of the first four discs according to its number. In this early stage of the algorithm all the robots are assigned to a series of search discs until termination. It can be seen that the target resides in disc 7, which belongs to robot 3, and that the target is unreachable to that robot from disc 7. At first, each robot searches for the target until it covers all the reachable area of the disc it is assigned to. The area that was not reachable in the current step, but is connected, like the gray areas depicted in Figure 3(a), will be covered in the next steps. Robot 1 finishes its local step in the first place and thus start its next global step ($k = 2$) which is local step ($i = 5$) searching in disc 5. Now the entire area of the first disc can be covered, yet some parts of disc 5 cannot be covered (Figure 3(b)). In the next two steps, robot 2 finishes its disc coverage and moves on to disc 6 ($k = 2, i = 6$) (Figure 3(c)), and robot 3 moves on to disc 7 ($k = 2, i = 7$) (Figure 3(d)). In both steps, again, some parts cannot be reached and the target in particular. At last, robot 4 reaches the next global step ($k = 2, i = 8$), where it moves to search in disc 8, and reaches the target that lies in disc 7. We simulated an execution of *MRSAM* with four robots, and compared it to execution with one robot on the same environment, including common starting and target points and identical initial disc. l_{opt} is marked in figure 4 with bold dashed line. For the initial radius $r_0 = 22mm$ and $D = 5mm$ the simulation results are, the optimal off-line solution $l_{opt} = 126.8mm$, the path length generated by robot 4, which found the target, during *MRSAM* execution, $l_j = 12121mm = 0.75l_{opt}^2 mm$, and the path length when running with one robot, $l = 18730mm = 1.17l_{opt}^2 mm$. These results show that *MRSAM* execution with 4 robots was 1.545 times faster than one robot execution. Observing table I one may wonder how come *MRSAM* was not 4 times faster? The answer lies in the initial radius r_0 that was chosen in relation to the the distance of T from S (or l_{opt}) which is discussed in detail in the conclusion. It should be clear that the values in table I are the maximal values and as was seen in this example, the actual performance, which depends on real environments and optimal parameter initialization, is sometime reduced.

VII. EXTENSIONS AND PRACTICAL SPEEDUPS OF *MRSAM*

Robustness is an important issue in practical situations when completion of the task is a matter of great significance and the units involved tend to fail or malfunction due to hardware or software flaws. *MRSAM* is robust in the sense that $n - 1$ out of the n robots can cease to work and yet the target will be found not unboundedly. This robustness is achieved thanks to the fact that each consequent disc is larger and contains the previous disc, such that if the setup includes communication between the robots, *MRSAM* can adjust itself each time a robot fails by recalculating the new multiplication factor αaccording to the new n, and if there is no communication between the robots, each robot will continue its original search plan and eventually find the target, producing a longer path length.

A. MRSAM *Extended to Deal with Multiple Targets*

The basic *MRSAM* Algorithm is designed to find a single target whose position is unknown in an unknown environment. The algorithm can easily be extended to solve a similar problem with more than one target, with a finite number of targets (a) or with unlimited number of targets in a bounded area (b). Changing the algorithm is done by removing the direction to the robot to STOP after reaching the first target (a) and (b), and adding a target counter and a direction to STOP after that counter has reached a specific value (b), both in step 1b. In order to restrain the range of search, definition of the limitation on the number of steps (i) the algorithm should perform before terminating is entered.

[5]This is merely a quarter of the symmetric environment.

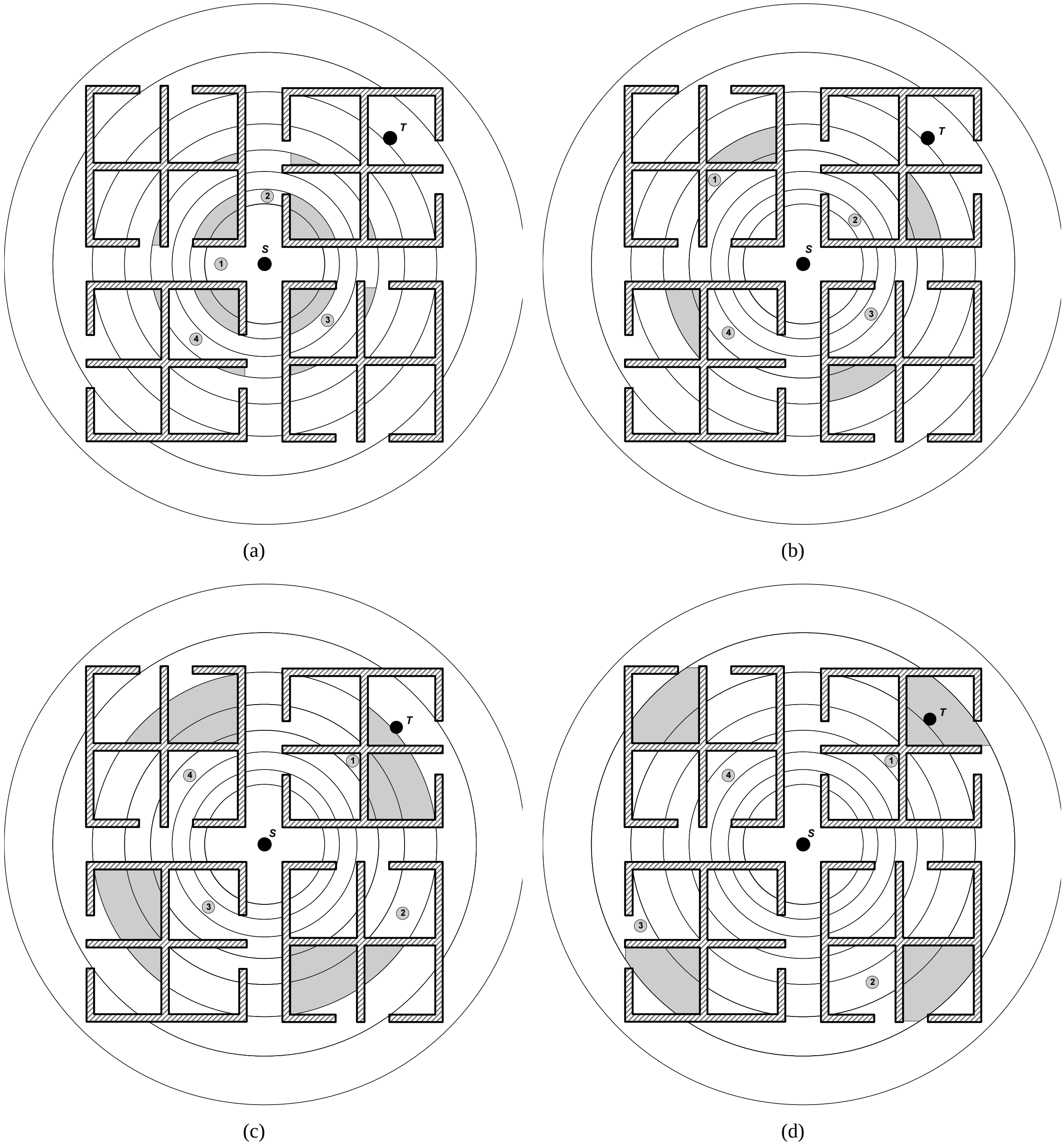

Fig. 3. Four steps of *MRSAM* execution. The gray area is the unreachable parts of the disc's robot in each step.

B. Practical Speedup

The individual robot in the basic *MRSAM* algorithm is assigned to a disc in which it performs a *coverage tour* searching for the target. The average performance of the algorithm can be improved by directing each robot to search only in the ring added to the last disc that has already been searched by itself or by one of the other robots, and that will be done only if the previous disc connected area was totaly covered, otherwise, the robot will perform a full search on the whole disc. Further improvement to that subcase is discussed in the second paragraph in the Conclusion (Section VIII).

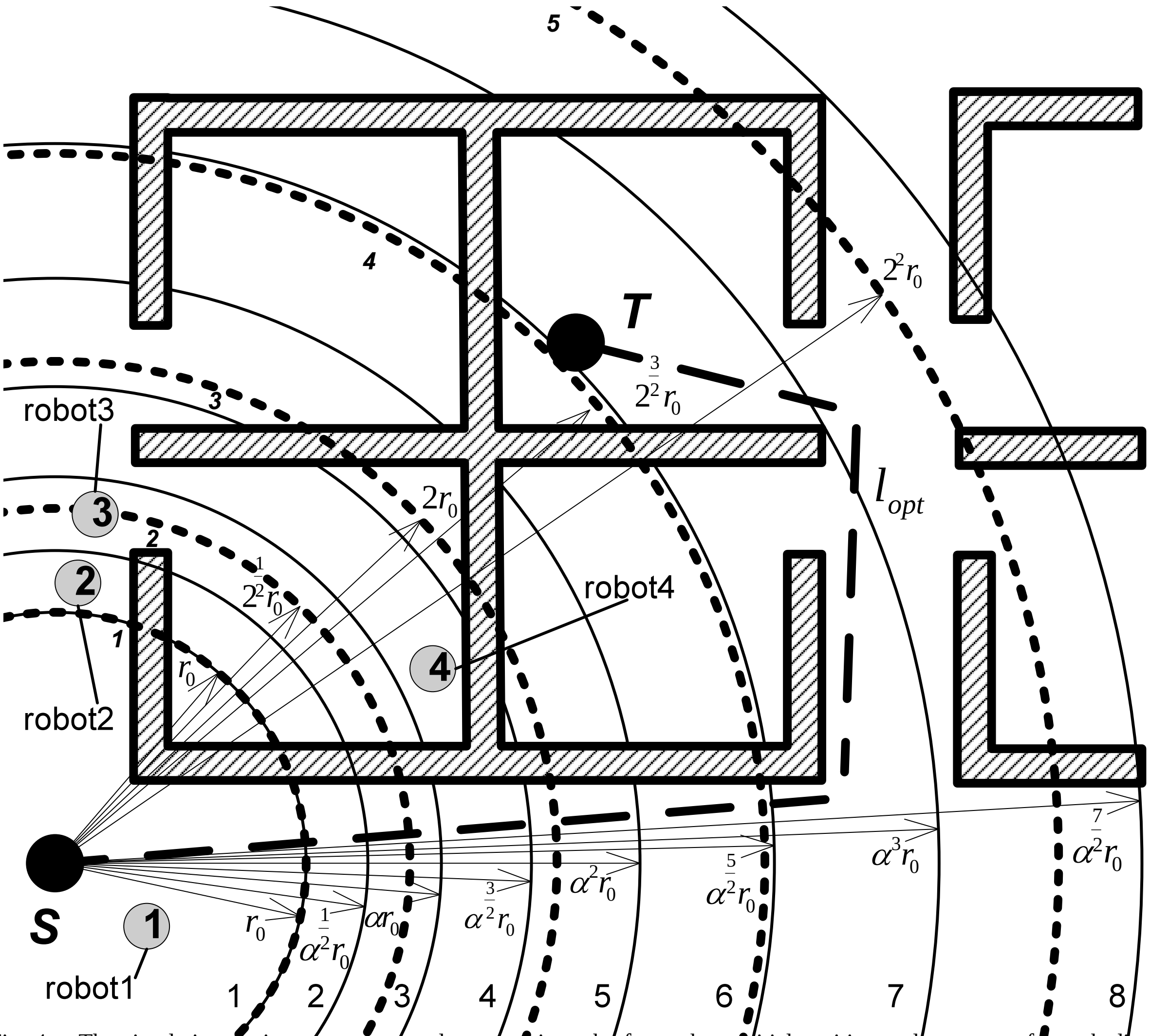

Fig. 4. The simulation environment, start and target points, the four robots initial position and two sets of search discs for two cases of *MRSAM* execution. The five dashed circles and the numbers attached to, denote the search discs for a single robot execution, and the eight regular circles denote the 4-robot execution of *MRSAM* .

VIII. Conclusion

The notion of competitive complexity classes generalizes the traditional notion of linear competitiveness to a pair of bounds which up to constant coefficients satisfy the same functional relationship between on-line performance and off-line optimal solution. In particular, we have shown that on-line multi mobile robot navigation to a target whose position is unknown belongs to the quadratic competitive complexity class. The *MRSAM* algorithm achieves the optimal quadratic bound while requiring only a linear amount of memory. The basic *MRSAM* algorithm has been consequently modified in order to exhibit a more efficient average case behavior, which was illustrated in office-like environments. An average performance comparison of the modified *MRSAM* algorithm in comparison with earlier algorithms is currently under preparation and will be reported later. In addition, we are working on an extension to *MRSAM* where each robot begins it search from a different starting point.

A matter worth mentioning is the fact that the radius of the initial search disc r_0, is determined prior to algorithm execution and it affects the overall path length and running time of *MRSAM* . It should be noted that r_0 is closely related to the position of the target T and to l_{opt}. Decreasing r_0 in the simulation corresponds to enlargement of the distance of T from S, resulting in more execution steps and thus better relative performance.

The following are some related open problems for further research. First, *MRSAM* assumes tactile senors. More sophisticated sensors such as vision and laser sensors do not have a significant advantage on tactile sensors in highly congested environments. However, practical environments tend to be reasonably sparse, and an adaptation of *MRSAM* to such sensors is an important open problem. Second, the constant coefficients in the quadratic upper bound on *MRSAM* and in the quadratic universal lower bound differ by values of $\frac{3}{2}(n+1)^{\frac{n+1}{n}}$.

Closing of this gap is a major challenge that can yield new algorithms that possess the quadratic competitiveness of *MRSAM* but perform much better on average. Third, the case discussed in the Practical Speedup Subsection VII-B, where a robot needs to ”return” to a partially uncovered previous disc, can be improved by means of a common environment map which is updated and shared by all the robots. That way, the robot that needs to make a full disc cover will be able to calculate the shortest path to return to the area it didn't cover. Creating and maintaining a common geometric map of the environment that includes information about the area that was already covered can save excess searching for the robots and thus speedup the average performance of *MRSAM* .Last, we assumed linear on-board memory. However, many mobile robot tasks are sufficiently complex as to allow only constant memory. Given this stricter memory limitation, one must re-explore the competitive complexity class of the basic problem considered in this paper.

References

[1] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, “Collaborative multi-robot localization,” in *Proc. of the 23rd German Conference on Artificial Intelligence (KI), Germany*, 1999.

[2] C. Icking and R. Klein, “Competitive strategies for autonomous systems,” in *Modelling and Planning for Sensor Based Intelligent Robot Systems*, H. Bunke, T. Kanade, and H. Noltemeier, Eds. World Scientific, 1995, pp. 23–40.

[3] Y. Gabriely and E. Rimon, “Competitive complexity of mobile robot on line motion planning problems,” in *Workshop on Algorithmic Foundations of Robotics*, 2004, pp. 249–264.

[4] D. T. Latimer, IV, S. Srinivasa, V. L. Shue, S. Sonne, H. Choset, and A. Hurst, “Towards sensor based coverage with robot teams,” in *Proc. IEEE Int. Conf. on Robotics and Automation*. IEEE, May 2002, pp. 961–967.

[5] S. Koenig and Y. Liu, “Terrain coverage with ant robots: a simulation study,” in *AGENTS '01: Proceedings of the fifth international conference on Autonomous agents*. New York, NY, USA: ACM Press, 2001, pp. 600–607.

[6] D. Kurabayashi, J. OTA, T. Arai, and E. Yoshida, “Cooperative sweeping by multiple mobile robots,” in *Proc. IEEE Int. Conf. on Robotics and Automation*, Minneapolis, Minnesota, April 1996.

[7] N. Hazon and G. A. Kaminka, “Redundancy, efficiency, and robustness in multi-robot coverage,” in *Proc. IEEE Int. Conf. on Robotics and Automation*, April 2005.

[8] I. Rekleitis, V. Lee-Shue, A. P. New, and H. Choset, “Limited communication, multi-robot team based coverage,” in *Proc. IEEE Int. Conf. on Robotics and Automation*, New Orleans, LA, April 2004, pp. 3462–3468.

[9] A. Datta and S. Soundaralakshmi, “Motion planning in an unknown polygonal environment with bounded performance guarantee,” in *Proc. IEEE Int. Conf. on Robotics and Automation*, 1999, pp. 1032–1037.

[10] C. Icking, T. Kamphans, R. Klein, and E. Langetepe, “On the competitive complexity of navigation tasks,” in *Revised Papers from the International Workshop on Sensor Based Intelligent Robots*. London, UK: Springer-Verlag, 2002, pp. 245–258.

[11] Y. Gabriely and E. Rimon, “Competitive on-line coverage of grid environments by a mobile robot,” *Comput. Geom. Theory Appl.*, vol. 24, no. 3, pp. 197–224, 2003.

[12] R. A. Baeza-Yates, J. C. Culberson, and G. J. E. Rawlins, “Searching in the plane,” *Inf. Comput.*, vol. 106, no. 2, pp. 234–252, 1993.