MRSAM: A Quadratically Competitive Multi-Robot Online 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

Abstract— **We explore an online 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 offline 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 offline 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 group of robots navigation to a target whose position is recognized only upon arrival belongs to a quadratic competitive class. This paper describes a new online navigation algorithm, called *MRSAM* (short for *Multi-Robot Search Area Multiplication*), which requires linear memory and has a quadratic competitive performance. Moreover, it is shown that in general any online 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's performance is illustrated in an 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 industry robotics, humanitarian robotics and military robotics where demining is a good example for the last two. In such problems, area coverage is a corresponding task since the searching unit will cover a certain area before finding the target,[1] An extended problem deals with multiple targets whose position is unknown, examples are 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.

[1]Assuming a finite non zero sized robot equipped with target detection sensors which detect the target upon arrival or within a specific constant range from it. This specific range is at least in order of magnitude smaller than the distance between the start and target points.

The most critical parameter in mobile robot motion tasks is the physical travel time rather than onboard 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 offline 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, as well as 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*, 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 of an online algorithm to the optimal offline 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 times 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 online 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-robot 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 highly related to this 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 offline solutions rely on apriori knowledge. An offline 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 communication setup.

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 online algorithm, there is a worst case path that yields a cost which is cl_{opt}^2, where c is a constant. 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. 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, onboard computation time, and onboard 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 onboard 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 onboard 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 offline 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 online 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 online algorithms for P. This objective requires a universal lower bound on the performance of all online 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 online 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 online algorithms for P, such that the two bounds are identical up to constant coefficients. Note that competitive complexity class characterizes the task P itself, not any specific algorithm for P. The remainder of the paper characterizes the competitive complexity class of the multi-robot online 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 robot from 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.

The following theorem establishes a quadratic lower bound on the performance of all online 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 of size D in an unknown planar environment to a target whose position is recognized only upon arrival. Let l be the path length generated by A for one of the robots, and let l_{opt} be the optimal offline path length. Then l satisfies the quadratic lower bound,

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

where ϵ is an arbitrary small positive parameter.

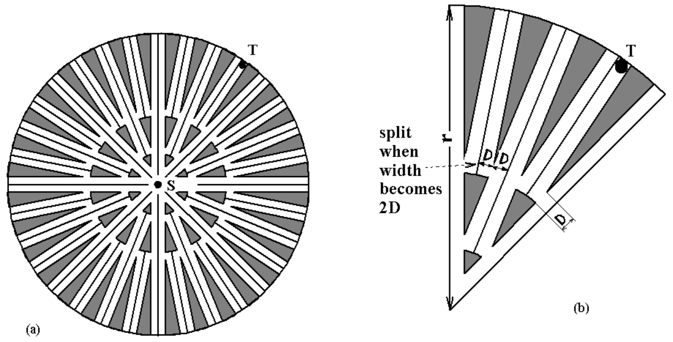

Fig. 1. (a) The radial corridors environment. (b) Close up view of the environment.

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 $l_{tot} \geq 4\pi r^2/3D$. This value of l_{tot} is obtained when none of the robots travel any part of the path of the other robots (In any other situation the total length will be greater). Thus, the total length of the path traveled by one robot satisfies $l \geq 4\pi r^2/3nD - r$, where the subtraction of r is due to the last corridor which need not be traced backward[2]. Since T is placed at a radial distance r from S we have that $l_{opt} = (1+\epsilon')r$, where ϵ' is an arbitrary small positive parameter. (The multiplication by $(1+\epsilon')$ is due to a fixed-length transition between successive radial corridors.) Substituting for r gives $l \geq c l_{opt}^2/(1+\epsilon')^2 - l_{opt}/(1+\epsilon')$,

[2] n is restricted to be as large as the number of corridors in order to maintain $l \geq 4\pi r^2/3nD - r \geq l_{opt}$

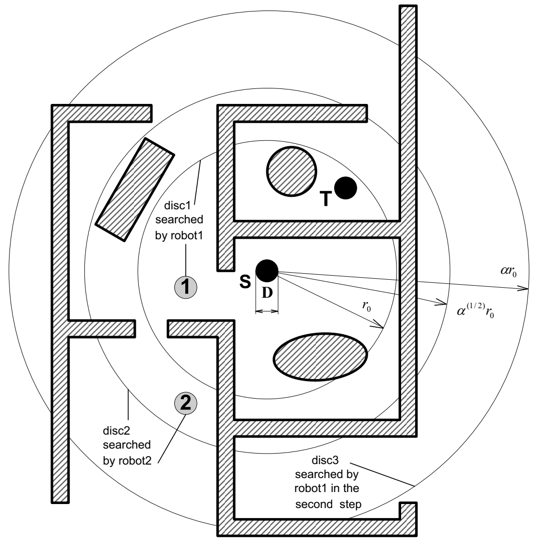

Fig. 2. A group of two robots launched by *MRSAM* searching for the target.

where $c = 4\pi/3nD$. We can write the last inequality as $l \geq cl_{opt}^2\left(\frac{1}{(1+\epsilon')^2} - \frac{1}{cl_{opt}(1+\epsilon')}\right) = cl_{opt}^2\left(1 - \epsilon'' - \frac{1}{cl_{opt}(1+\epsilon')}\right)$, where we substituted $1/(1+\epsilon')^2 = 1 - \epsilon''$. The term $\epsilon = \epsilon'' + 1/cl_{opt}(1+\epsilon')$ contains the quotient D/l_{opt} which can be made arbitrarily small for sufficiently large environments, hence $l \geq c(1-\epsilon)l_{opt}^2$. ■

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 ($j = 1$) 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 [3] $\alpha > 1$, namely, the areas of the discs will be $S_0, \alpha S_0, \alpha^2 S_0, \alpha^3 S_0 \ldots$. 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. 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 online 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.

[3]This is a significant 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.

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. In our example (Figure 2) 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. 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[4] $r_j(p_j) = \alpha^{\frac{p_j - 1}{2}} r_0$.

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 $p_j = p_j + n$,
   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 do not need to communicate with each other apart from a stop signal when the target is found. This implies a decentralized approach with no or with limited communication. Second, a robot that finished searching a disc will immediately proceed to the next disc assigned to it, regardless of the state of the other robots. Next, the coverage tour in step 1a can be executed with a trivial DFS algorithm using linear memory. Last, 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. Clearly this is a limitation of the problem itself, which assumes no knowledge of the target location. A detailed example of *MRSAM* execution appears in Section VI.

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*.

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

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

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

where D is the robot size, r_0 is the initial search radius, α is the multiplication factor which is a function of n, and l_{opt} is the length of the optimal offline path from S to T.

Note that the upper bound is scalable, in the sense that both summands have units of length.

The proof of this proposition was relegated to [12]. The following lemma, inspired by [13], asserts that search area multiplying is indeed an optimal strategy.

Lemma 5.2: Let n be the number of robots searching for the target, The Competitive Complexity of *MRSAM* is minimal when the multiplication factor α equals $\alpha = (n+1)^{1/n}$.

Proof: Suppose the target was found in the i^{th} disc by robot number j after covering that disc entirely. The total area S_j covered by that robot as obtained in (Prop. 5.1) is $S_j < \pi\left(\frac{\alpha^{n+1}}{\alpha^n-1}\right) l_{opt}^2 = \beta_n \pi l_{opt}^2$, Minimizing[5] for α and equating with zero yields, $\alpha = (n+1)^{1/n}$.
This is an extremum value, a second derivative will check the minimality of α, $\frac{\partial^2}{\partial \alpha^2}(\beta_n) > 0$, which implies that l gets minimal values when $\alpha = (n+1)^{1/n}$. ■

The following corollary asserts that if a path from the start point S to the target T exist, *MRSAM* algorithm will find T.

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. The two properties implies the completeness of *MRSAM*. ■

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 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 β_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, β_n approaches 1, as well. On the other hand, for the second case where $n = 1$, $\alpha = (1+1)^{1/1} = 2$, therefore, $\beta_n = \frac{2^{1+1}}{2^1-1} = 4$ 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 β_n is a monotonic rising function, as well.

[5] while taking into account $\alpha > 1$ and $n \geq 1$ for realistic execution, which implies $\alpha^n - 1 > 0$

It can easily be calculated that using 4 robots, *MRSAM* doubles the performance compared to execution with one robot, when using 13 robots *MRSAM* triples the performance and with one hundred robots, β_n approaches one, and *MRSAM* multiplies the performance by a factor of 3.78.

Theorem 1: The online multi-robot 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}, $l < \frac{2\pi}{D}\left(\frac{\alpha^{n+1}}{\alpha^n-1}\right) l_{opt}^2 + \frac{2\pi r_0^2}{D}$ 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-cubicle environment depicted in Figure 3. The area multiplication factor for $n = 4$ is $\alpha = 1.495$, $\sqrt{\alpha} = 1.223$, and in the initial global step $k = 1$ (local steps $p_1 = 1, p_2 = 2, p_3 = 3, p_4 = 4$) 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. 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, $p_1 = 5$, searching in disc 5. Now the entire area of the first disc can be covered, yet some parts of disc 5 cannot be covered. In the next two steps, robot 2 finishes its disc coverage and moves on to disc 6, $k = 2, p_2 = 6$, and robot 3 moves on to disc 7, $k = 2, p_3 = 7$ (Figure 3(b)). It can be seen that the target resides in disc 7, which is assigned to robot 3, and that the target is unreachable to that robot from disc 7. In both steps, again, some parts cannot be reached and the target in particular. At last, robot 4 reaches the next global

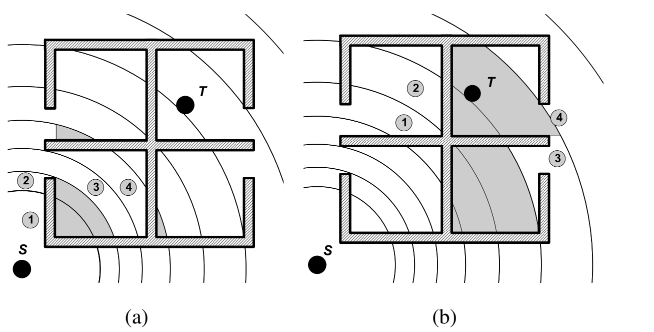

Fig. 3. The gray area marks the unreachable parts of the discs for each robot for (a) the first 4 steps, (b) the next 4 steps.

step, $k = 2, p_4 = 8$, 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. For the initial search disc radius $r_0 = 22mm$ and robot radius $D = 5mm$ the simulation results are as follows: the path length of the optimal offline solution $l_{opt} = 126.8mm$, the path length generated by robot 4, which found the target, during *MRSAM* execution, $l = 12121mm \cong 0.75 l_{opt}^2 mm$, and the path length when running with one robot, $l = 18730mm \cong 1.17 l_{opt}^2 mm$. These results show that *MRSAM* execution with 4 robots was 1.545 times faster than one robot execution, but, according to the previous section, *MRSAM* suppose to perform more than two times faster. The difference result from the selection of r_0 that was chosen relatively large in comparison to l_{opt}, which is discussed in the conclusion. It should be clear that the calculated values of β_n 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. Conclusion

The robustness of *MRSAM* as well as extensions of the algorithm to deal with multiple targets and some practical speedup are discussed in detail in [12].

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 online performance and offline optimal solution. In particular, we have shown that online multi-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* 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* 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 its 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 affects the overall path length of *MRSAM*. It should be noted that r_0 is closely related to l_{opt}, such that decreasing r_0 corresponds to enlargement of l_{opt}, resulting in more execution steps and eventually in reduced search area.

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 important. 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 practical speedup subject discussed in [12], 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 did not 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 onboard 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*, May 2002, pp. 961–967.

[5] S. Koenig and Y. Liu, "Terrain coverage with ant robots: a simulation study," in *Proceedings of the Fifth International Conference on Autonomous Agents (AGENTS-01)*, May 28-June 1 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] S. Sarid, A. Shapiro, and Y. Gabriely, "Mrsam: A quadratically competitive multi-robot online navigation algorithm," Department of Mechanical Engineering, Ben Gurion University, Israel, http://www.bgu.ac.il/~ashapiro, Tech. Rep., 2006.

[13] 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.