The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

A Time Competitive Heterogeneous Multi Robot Path Finding Algorithm

Shahar Sarid and Amir Shapiro

Abstract— **We investigate the path finding problem to a target whose position is known in an unknown and unbounded environment. We present a novel motion planning algorithm which uses a group of heterogeneous robots to search for the path to the target. The algorithm assigns the robots in pairs, each two robots within a pair have the same velocity, and are cooperating to search for the path to the target. The algorithm artificially bounds each pair's search to an ellipse whose focal points are the start and the target points. Each robot pair has a different velocity, thus each pair is assigned to an ellipse with an area corresponding to the search time according to its velocity. The algorithm's performance is analyzed using time competitiveness definitions, and its upper bound is proved to be quadratic in the optimal off-line solution. The algorithm is complete and robust.**

I. INTRODUCTION

The problem of searching by autonomous mobile robots receives a lot of attention in recent years, both in academic research and in industry. Mobile robots already play a significant role in our lives and their applications are growing rapidly. For example, at home, the iRobot Roomba household totally autonomous vacuum cleaner [1] searches and cleans dirt. In space, the partially autonomous Rovers on Mars searching for specimens and exploring its surface [2] can find their way to a predefined target avoiding local obstacles. At the service of police and army forces there are bomb disarming robots, and in industry, AGVs transfer materials in production plants avoiding dynamic obstacles [3].

The advantages of employing multiple robots rather than a single robot in such applications are numerous, the most important are increased efficiency, shorter task duration, and robustness [4]. The use of heterogeneous robots is taken under considerations in cases where different roles are needed for each robot [5], or in cases where a group of robots is formed out of different types of robots due to inventory constraints. Heterogeneity can appear in many flavors, different types of sensors, various sizes, speeds, power units types, loading capabilities, end-effectors, and so on.

It is common to distinguish between searching, exploration and coverage missions, where searching [6] is done for a target whose position is unknown or in an unknown environment. Exploration [7] usually comes together with mapping (SLAM [8]), where an unknown area needs to be explored and mapped for future applications. Coverage tasks [9] deals with moving all over a certain area with some tool, e.g. lawn mowing, vacuum cleaning etc.

This work focuses on searching for a target whose position is known, in an environment which is unknown and unbounded, thus, a path to the target must be searched and found. We use multiple heterogeneous robots, which differ only in their velocities. The robots use tactile sensors and each robot knows its position.

Based on *CBUG* [10] for a single robot search and *MRBUG* [11] for homogeneous group of robots we introduce *HMRBUG* algorithm. *HMRBUG* , Heterogeneous Multi Robot BUG algorithm uses a modified *BUG1* [12] technique to search for a path to a known target in an unknown environment. Since *BUG1* can have very bad performance in certain situations, such as long obstacles and unbounded environments, *HMRBUG* bounds *BUG1* search in ellipses whose focal points are the start and target positions. The uniqueness of the ellipse is that it is the locus of all points in the plane whose distances to the two focal points add to a constant. Using ellipse means searching with equal chance for all paths from start to target with the same lengths. *HMRBUG* utilizes multiple heterogeneous robots, by deploying pairs of robots in ellipses whose areas and search times are growing until reaching the target. In each step the robot pair's search is bounded to an ellipse, however, eventually, the search area is growing until reaching the target.

The structure and contributions of this paper are as follows. First we introduce the problem of path finding by a group of robots which are heterogeneous in their velocity, and their search environment is unknown and unbounded. In the next section setup definitions are presented, including the robots' size, and the notion of time complexity, which is used later for performance analysis. In section III *HMRBUG* , a novel algorithm for the problem of path finding by heterogeneous robots is introduced and explained. In the following section *HMRBUG* performance is analyzed, we find its upper bound to be quadratic in the optimal off-line solution and prove *HMRBUG* is complete and robust. Section V shows execution example for two robot pairs in an office like environment, and last, we conclude and discuss future work.

II. SETUP AND DEFINITIONS

Before introducing *HMRBUG* algorithm and analyzing it, we present some definitions regarding performance measures and basic setup of the robots. The robots are assumed to be of size D, which can be considered as the width of

S. Sarid and A. Shapiro are with the Department of Mechanical Engineering, Ben Gurion University of the Negev, P.O. Box 653, Beer-Sheva 84105, Israel {sarids,ashapiro}@bgu.ac.il

S. Shapiro is also affiliated with Ben Gurion University's Paul Ivanier center for Robotics Research and Production, and with Perlstone Center for Aeronautical Engineering.

978-1-4244-6676-4/10/$25.00 ©2010 IEEE

the robots, a property which will serve us later in analyzing *HMRBUG* performance. The robots have tactile sensors which detects obstacles in close proximity, and optimal self positioning system.

Definition 1 (Generalized Time Competitiveness [13]): An on-line algorithm solving a task P in time T is $f(T_{opt})$ *time competitive* when T is bounded from above by a scalable function $f(T_{opt})$ over all instances of P, and T_{opt} is the optimal off-line solution achieved while knowing all the information about the environment's geometry. In particular, $T \leq c_1 T_{opt} + c_0$ is the linear time competitiveness, while $T \leq c_2 T_{opt}^2 + c_1 T_{opt} + c_0$ is a quadratic time competitiveness, where the c_i's are positive constant coefficients that depend on the robot size D, the robots' velocity, the number of robots, and the geometry of the environment.

The meaning of scalability is as follows. When performance is measured in time units such as seconds, one must ensure that both sides of the relationship $T \leq f(T_{opt})$ posses the same units, so that change of scale would not affect the bound. For instance, if T is measured in sec, the coefficient c_2 in the relationship $T \leq c_2 T_{opt}^2 + c_1 T_{opt} + c_0$ must have units of sec^{-1}, c_1 must be dimensionless, and c_0 must have units of sec. Note that the definition of $f(T_{opt})$ *time competitive* focuses on a particular algorithm solving the task P.

The following conditions assure the search area and time of each consequent robot pair will grow in order to prevent redundant search and to eventually reach the target.

Condition 1: (Search ellipse area ratio) The area of each consequent search ellipse is greater than the area of the previous search ellipse.

During the same period of time, robots with different velocities will travel unequal path lengths, and the ratio of the traveled path lengths is the same as the ratio of the velocities. Consequently, a fast robot might finish searching the next search ellipse before the slow robot finished searching in the previous ellipse, thus, for *HMRBUG* , condition 1 does not suffice, and the following condition completes it.

Condition 2: (Search time ratio) The time of search within each consequent search ellipse is greater than the time of search within the previous search ellipse.

III. HMRBUG Motion Algorithm to a Known Target

We now introduce *HMRBUG* algorithm for reaching a known target. *HMRBUG* uses $2n$ robot pairs with n different velocities, $v_j = \beta_j v, \quad j = 1, \ldots, n,$ where $\beta_{j+1} \geq \beta_j \geq 1, \quad j = 1, \ldots, n-1$ and v is the velocity of each of the robots in the slowest robot pair and β_j is the ratio between the velocity of robot pair j and the velocity of the slowest robot pair, v.

HMRBUG , solves the problem of finding a path to a known target using a group of robots. *HMRBUG* deploys each pair of robots to search for the target in a virtual bounding ellipse, and inside uses *PBUG1* algorithm [11] as a sub procedure. *PBUG1* algorithm is described in the following section.

A. PBUG1 *Motion Algorithm for a Pair of Robots [11]*

We now review *PBUG1*, a version of *BUG1* [12] for a pair of robots which uses the same problem's definitions as *BUG1*. In *PBUG1* a pair of robots that start from a common start point $\mathcal{S}$ needs to find a path to a target $\mathcal{T}$ whose position is known, in an unknown planar environment. The pair of robots will move together toward the target in a straight line until they hit an i^{th} obstacle at a point marked as *Hit point* H^i, $i = 1, 2, \ldots$. At that point they split, robot R_L turns left and robot R_R turns right, and they circumnavigate the obstacle from different directions. On that account, each robot encircles half of the obstacle's perimeter. While moving, each robot calculates and remembers the closest point on the obstacle's boundary to the target. Upon meeting, the robots compare the recorded information, decide which point is the closest to the target, join and again move together to that closest point which they mark as *Leave point* L^i, $i = 1, 2, \ldots$. Finally, the robots continue to move together toward the target.

1) Setup and Definitions of PBUG1*:* The basic setup and definitions of *PBUG1* are the same as in [11]. *PBUG1* uses two mobile robots, R_L and R_R, each has a different pre defined local direction for moving around an obstacle, *left* and *right* accordingly. The hit and leave points are common for both of the robots. The procedure *PBUG1* needs only one register for each robot, $Reg1$, which is used to store the coordinates of the current point, Q_m, of the minimum distance between the obstacle's boundary and the target. The robots compare their Q_m points and go together to the one with the shorter value.

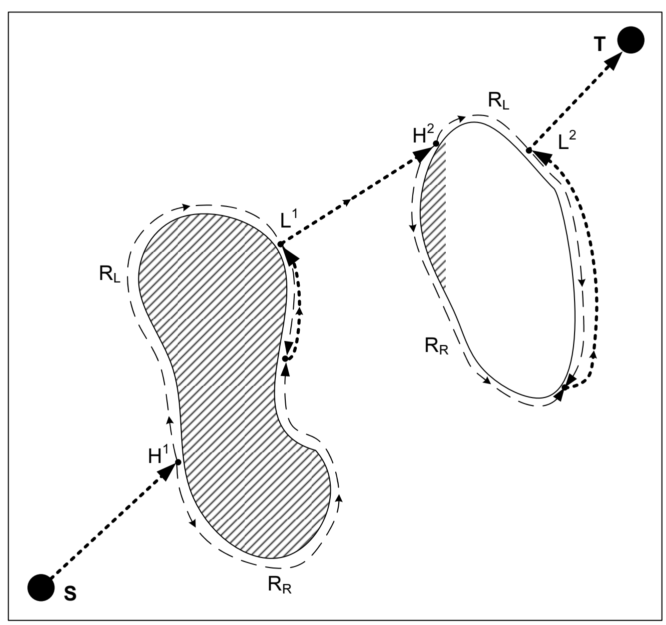

Fig. 1. A pair of robots, R_L and R_R executing *PBUG1*. The dense dashed lines mark a mutual path of the two robots

B. Target Reachability Test

PBUG1 determines that the target is unreachable and trapped inside an obstacle using *BUG1* method [12], which checks the direction to the target after circumnavigating an obstacle. If this direction points into the last obstacle, the target is surrounded by that obstacle, since the leaving point is the closest point to the target on the obstacle's boundary.

C. HMRBUG *Algorithm for a Group of Robots*

HMRBUG algorithm launches n pairs of robots from a common starting point $\mathcal{S}$ and assigns each pair R_j to a different ellipse to search for a path to the target $\mathcal{T}$ in it, each ellipse's focal points are $\mathcal{S}$ and $\mathcal{T}$.

The first pair of robots R_1 is designated to the initial ellipse of search time T_0, and each of the following robots starts its search in an ellipse of search time larger than the previous ellipse's search time by a factor of $\alpha_j,\ \ \alpha_j > 1$, namely, the search times of the ellipses will be $T_0, \alpha_2 T_0, \alpha_2\alpha_3 T_0, \alpha_2\alpha_3\alpha_4 T_0, \ldots$. For example, in Fig. 2, robots 1_L and 1_R are initially assigned to search for a path to the target inside an ellipse of search time T_0 and robots 2_L and 2_R are assigned to search inside an ellipse of search time $\alpha_2 T_0$. The search for the path inside an ellipse is done by the pair of robots assigned to that ellipse using *PBUG1*.

In *HMRBUG* , the execution of *PBUG1* regards the ellipse as a virtual obstacle's boundary. If the target is detected, the algorithm terminates, otherwise, the pair of robots repeats the process on the next unassigned ellipse in the series. A formal description of the basic algorithm appears in Algorithm 1.

Before analyzing the time competitiveness of the algorithm, we make the following remarks. First, during initialization, after getting the values of $n, T_0, \mathcal{S}$, and $\mathcal{T}$, each robot is assigned to a number j and to a local direction, $Left$ or $Right$ and thus can calculate its future search ellipses parameters, which means that after a pair of robots has finished searching for a path in an ellipse, it can immediately continue to search in the next ellipse regardless of the state of the other robots. Second, the method *PBUG1* used to determine that the target is unreachable and trapped inside an obstacle in step 2 is discussed in subsection III-B. *HMRBUG* assures in step 2a that the robots were not bounded by the ellipse and thus guarantee that the target is unreachable. Third, regarding the memory requirements, in *HMRBUG* , each robot executing *PBUG1* uses the same amount of memory as in *BUG1* with a little modification, plus a constant amount of memory. Namely, the target position $\mathcal{T}$, the current obstacle's hit point, the distance of the closest point on the current obstacle's boundary to the target and the two distances to that point along the obstacle's boundaries from its current position which are *BUG1* necessities. *PBUG1* memory modification is in the last requirement, where these two distances to the leave point are not necessary and should be recorded by each robot, as was discussed in Subsection III-A.1. *HMRBUG* additional memory requirements are the start position $\mathcal{S}$ and the current ellipse's parameters a_p, b_p.

Algorithm 1: *HMRBUG*

Sensors: A position sensor.
An obstacle detection sensor.
Input: Position of a start $\mathcal{S}$ and a target $\mathcal{T}$ points.
An initial ellipse with focal points $\mathcal{S}$ and $\mathcal{T}$
An initial search time T_0
n pairs of robots $\{1_L, 1_R, 2_L, 2_R, \ldots, n_L, n_R\}$
with different velocities, $v_j = \beta_j v,\ j = 1, \ldots, n$
$\beta_{j+1} \geq \beta_j \geq 1,\ j = 1, \ldots, n-1,\ \ \beta_1 = 1$
Initialization: For each robot pair $R_j,\ \ j = 1, \ldots, n$:
Set global step $p = 1$
Set initial leave point $L^0_{e_j}$ = $\mathcal{S}$,
Set multiplication factor $\alpha_j = \frac{(n+1)^{(\frac{1}{n})}\beta_{j-1}^{\frac{n-1}{n}}}{\prod_{k=2,k\neq j-1}^{n}\beta_k^{\frac{1}{n}}}$,
where $\beta_{j-1} = \beta_n$ when $j = 1$
Set initial search ellipse parameters:
focal points are $\mathcal{S}$ and $\mathcal{T}$,
semi major axis[1] $a_0 = a_0(\alpha_j, T_0, \mathcal{S}, \mathcal{T}, \beta_j, v)$,
semi minor axis[1] $b_0 = b_0(a_0, \mathcal{S}, \mathcal{T})$.
For each robot pair R_j, Repeat:
Initialize *PBUG1* with the following parameters:
Create an outer virtual obstacle's boundary
with current search ellipse.
Start point is $\mathcal{S}$, target is $\mathcal{T}$.
Set $i = 1$.
Leave point is L^0.
Execute *PBUG1* until one of the following occurs:
(1) *PBUG1* terminates at $\mathcal{T}$: STOP, target is found.
(2) $\mathcal{T}$ is trapped inside an obstacle:
(a) If the obstacle does not intersect the eR_j ellipse: STOP, the target is unreachable.
(b) Else, move to the next unoccupied ellipse:
Set $p = p + 1$.
Set current search ellipse parameters:
semi major axis[1] $a_p = a_p(a_0, \alpha_j, p)$,
semi minor axis[1] $b_p = b_p(a_0, a_p)$.
Set L^0 at *PBUG1* termination point.
End of Repeat loop

[1]Calculations of a_0, b_0, a_p, b_p is presented in section IV.

IV. PERFORMANCE ANALYSIS

In this section, the performance of *HMRBUG* is analyzed and an upper bound on the traveling time to reach the target is formed. First, two conditions regarding the ellipses areas and search times are formulated. Those conditions assist the calculations and the convergence of the upper bound of *HMRBUG* . Then, using the ellipse geometrical properties and the optimal off-line solution, the upper bound is achieved.

In the following two lemmas we use terms related to *Configuration Space*. The configuration space (or $\mathcal{C}$-space) of a disc shaped robot is $\mathbb{R}^2$, and the $\mathcal{C}$-space obstacle $\mathcal{CB}_i$ consist of all robot configurations where it intersects the obstacle $\mathcal{B}_i$.

Definition 2 ([10]): Let $\mathcal{CB}_i$ be the $\mathcal{C}$-space obstacle induced by an obstacle $\mathcal{B}_i$ for a disc robot of size D. The *traceable obstacle* induced by $\mathcal{B}_i$, denoted $\boldsymbol{\mathcal{B}_i}$, is obtained by filling any internal holes in $\mathcal{CB}_i$ and then shrinking $\mathcal{CB}_i$ inward by a distance of $D/2$.

Lemma 4.1 ([10]): Let a planar environment contain z disjoint traceable obstacles $\boldsymbol{\mathcal{B}_i}$, $i = 1, \ldots, z$. Let a disc robot of size D trace the i^{th} obstacle's boundary, and let q_i be the total area swept by the robot during tracing of the i^{th} boundary. Let $\mathcal{C}$ be any simple closed curve which surrounds the z regions swept by the robot. Then $\sum_{i=1}^{z} q_i \leq 4\mathcal{A}(\mathcal{C})$, where $\mathcal{A}(\mathcal{C})$ is the area of the traceable obstacle-free points enclosed by $\mathcal{C}$.

Note that the regions swept during tracing of the individual boundaries may overlap, so that in general the sum $\sum_{i=1}^{z} q_i$ may be larger than $\mathcal{A}(\mathcal{C})$. The following lemma is written in the spirit of [11].

Lemma 4.2: The path travel time t_{ji} of the the i^{th} ellipse traversed by each robot of the j^{th} pair searching for the path to the target is bounded by

$$t_{ji} \leq 4\frac{A_i}{\beta_j v D} + (||L_i^0 - \mathcal{T}|| - ||L_{i+n}^0 - \mathcal{T}||)/(\beta_j v),$$

where A_i is the area of the i^{th} ellipse, D is the size of each robot, $\beta_j v$ is the velocity of each robot of the j^{th}'s pair. L_i^0 is the start point at the i^{th} ellipse, n is the number of robot pairs, L_{i+n}^0 is the start point at the next ellipse of the pair of robots, and $||\gamma - \delta||$ denotes the Euclidean distance between γ and δ.

Proof: When moving toward the target in the i^{th} ellipse, each robot of the pair of robots assigned by *HMRBUG* to that ellipse is executing *PBUG1*. The regions swept by the robots during circumnavigation of obstacles in this ellipse (including the ellipse itself) are surrounded by the ellipse's boundary. Identifying the latter boundary with the curve $\mathcal{A}(\mathcal{C})$ from Lemma 4.1, the total path length of the two robots during circumnavigation of the obstacles is at most $4A_i/D$, where A_i is the i^{th} ellipse's area. Since each robot travel exactly half of the way, the path length of one robot is not more than $2A_i/D$. Recall now that under *BUG1* the robot circumnavigates the boundary of each obstacle at most 1.5 times, here each of the two robots will circumnavigates the boundary of each obstacle only one time at most. Hence, the total length of each robot's path during boundary following is at most $4A_i/D$. Recall, too, that under *BUG1* motion between obstacles is always directly to the target. The total length of these motion segments equals to the net decrease of the distance of the robot from $\mathcal{T}$, which is $||L_i^0 - \mathcal{T}|| - ||L_{i+n}^0 - \mathcal{T}||$. Adding the two terms and dividing by the robots' velocity gives the result. ■

The following lemma states that the last ellipse search time is bounded from above. This lemma and its proof are inspired by [11].

Lemma 4.3: Let $\mathcal{T}$ be reachable from $\mathcal{S}$. If the initial ellipse contains no path from $\mathcal{S}$ to $\mathcal{T}$, robot pair j in *HMRBUG* reaches the target in an ellipse whose search time T_{f_j} is bounded by,

$$T_{f_j} \leq \frac{\pi \alpha_j \beta_n t_{opt}}{4\beta_{j-1} D}\sqrt{(\beta_n v t_{opt})^2 - ||\mathcal{S} - \mathcal{T}||^2}, \qquad (1)$$

where, t_{opt} is the travel time of the optimal off-line path from $\mathcal{S}$ to $\mathcal{T}$ by the fastest robot pair, α_j is the multiplication factor of the robot pair which reached the target, β_j is the velocity factor of robot pair j, and for $j = 1$, $\beta_{j-1} = \beta_n$, v is the slowest robot's velocity and D is the robots' width.

Proof: An ellipse with focal points $\mathcal{S}$ and $\mathcal{T}$ satisfies the inequality $\varepsilon = \{x : ||x - \mathcal{S}|| + ||x - \mathcal{T}|| \leq 2a\}$, where $2a$ is the length of the ellipse's major axis. Consider now the optimal off-line path from $\mathcal{S}$ to $\mathcal{T}$ of length l_{opt}. Every point x along this path satisfies the inequality $||x - \mathcal{S}|| + ||x - \mathcal{T}|| \leq l_{opt}$. It follows that the entire optimal off-line path lies in an ellipse with focal points $\mathcal{S}$ and $\mathcal{T}$ and major axis $2a \leq l_{opt}$. Next recall that the area of an ellipse is given by πab, where $2b$ is the length of the ellipse's minor axis. In ellipse with focal points $\mathcal{S}$ and $\mathcal{T}$ $||\mathcal{S} - \mathcal{T}||^2/4 + b^2 = a^2$. Hence, $b \leq \frac{1}{2}\sqrt{l_{opt}^2 - ||\mathcal{S} - \mathcal{T}||^2}$, where we used the inequality $a \leq l_{opt}/2$. Let A_{opt} denote the area of the smallest ellipse with focal points $\mathcal{S}$ and $\mathcal{T}$ which contain the optimal off-line path. Substituting the expressions for a and b in πab gives the upper bound:

$$A_{opt} = \pi ab \leq \frac{\pi}{4} l_{opt}\sqrt{l_{opt}^2 - ||\mathcal{S} - \mathcal{T}||^2}. \qquad (2)$$

By assumption the initial ellipse contain no path from $\mathcal{S}$ to $\mathcal{T}$. Hence, *HMRBUG* multiplies the search time of the ellipse by a factor of α_j at least once. In worst case scenario, robot pair R_{j-1} searched for a path in an ellipse whose area is $A_{opt} - \epsilon$ and search time $T_{opt_{j-1}} - \epsilon$, substituting $t_{opt} = l_{opt}/(\beta_n v)$ and (2) into $T_{opt_{j-1}} = A_{opt}/(\beta_{j-1} v D)$, yields $T_{opt_{j-1}} \leq \frac{\pi \beta_n t_{opt}}{4\beta_{j-1} D}\sqrt{(\beta_n v t_{opt})^2 - ||\mathcal{S} - \mathcal{T}||^2}$.

Robot pair R_{j-1} could not reach the target in the current ellipse. Consequently, R_j is assigned afterwards to search for it in an ellipse whose search time satisfies the inequality $T_{f_j} \leq \alpha_j T_{opt_{j-1}}$. Substituting for $T_{opt_{j-1}}$ in the inequality $T_{f_j} \leq \alpha_j T_{opt_{j-1}}$ gives the result. ■

The following proposition establishes a quadratic time competitive upper bound on *HMRBUG* .

Theorem 1: (Quadratic time competitive complexity) Assume target $\mathcal{T}$ is reachable from $\mathcal{S}$. Let *HMRBUG* use n robot pairs with velocities: $v_j = \beta_j v,\ \forall j,\ j = 1, 2 \ldots n$, $\beta_{j+1} \geq \beta_j \geq 1,\ \forall j,\ j = 1, 2 \ldots n-1$.
Then the traveling time of robot pair j which reached the target:

$$T_{R_j} < \frac{\pi \alpha_j \beta_n \left(\prod_{i=1}^{n} \alpha_i\right) t_{opt}\sqrt{(\beta_n v t_{opt})^2 - ||\mathcal{S} - \mathcal{T}||^2}}{4D\beta_{j-1}\left(\prod_{i=1}^{n} \alpha_i - 1\right)}.$$

Proof: The path to the target is assumed to lie within an ellipse whose area is given in (2) and search time (1). First, we will inspect the case in which robot pair no. 1, R_1 reaches the target. In worst case scenario, the last robot pair, R_n, searched for a path in an ellipse whose area is $A_{opt} - \epsilon$ and search time $T_{opt_n} - \epsilon$ thus could not reach the target.

Consequently, R_1 is assigned afterwards to search for it in an ellipse whose area is (1)

$$T_{f_1} \leq \frac{\pi\alpha_1 t_{opt}}{4D}\sqrt{(\beta_n v t_{opt})^2 - ||\mathcal{S}-\mathcal{T}||^2}. \tag{3}$$

The sum of the ellipses' search times of R_1 is,

$$\begin{aligned} T_{R_1} &\leq t_{1,1} + t_{1,1+n} + t_{1,11+2n} + \cdots + t_{1,i} \\ &= T_0 + \left(\prod_{k=1}^{n}\alpha_k\right)^1 T_0 + \left(\prod_{k=1}^{n}\alpha_k\right)^2 T_0 \\ &\quad + \cdots + \left(\prod_{k=1}^{n}\alpha_k\right)^{\left(\frac{i-1}{n}\right)} T_0 \end{aligned} \tag{4}$$

This series of times is a converging geometric series, which its sum is,

$$T_{R_1} \leq \frac{\left(\prod_{k=1}^{n}\alpha_k\right)^{\left(\frac{i-1}{n}+1\right)} - 1}{\prod_{k=1}^{n}\alpha_k - 1}T_0 < \frac{\left(\prod_{k=1}^{n}\alpha_k\right)^{\left(\frac{i-1}{n}+1\right)}}{\prod_{k=1}^{n}\alpha_k - 1}T_0 \tag{5}$$

Comparing the two expressions for the time to cover the last ellipse, $t_{1,i}$ from (4), and T_{f_1} from (1), T_0 can be calculated, $T_0 = \frac{\pi\alpha_1 t_{opt}\sqrt{(\beta_n v t_{opt})^2 - ||\mathcal{S}-\mathcal{T}||^2}}{4D\left(\prod_{i=1}^{n}\alpha_i\right)^{\left(\frac{i-1}{n}\right)}}$. Substituting T_0 into (5) and simplifying yields,

$$T_{R_1} < \frac{\pi\alpha_1 t_{opt}\sqrt{(\beta_n v t_{opt})^2 - ||\mathcal{S}-\mathcal{T}||^2}\left(\prod_{i=1}^{n}\alpha_i\right)}{4D\left(\prod_{i=1}^{n}\alpha_i - 1\right)}.$$

Accordingly, for R_2, in worst case scenario, R_1 covered an ellipse whose search time $T_{opt_1} - \epsilon$ and thus did not find target. Consequently, R_2 is assigned afterwards to search for it in an ellipse whose search time is according to Lemma 4.3

$$T_{f_2} \leq \alpha_2 T_{opt_1} = \frac{\pi\alpha_2\beta_n t_{opt}}{4D\beta_1}\sqrt{(\beta_n v t_{opt})^2 - ||\mathcal{S}-\mathcal{T}||^2}. \tag{6}$$

The sum of the search times by R_2 is, $T_{R_2} \leq t_{2,1} + t_{2,2+n} + t_{2,2+2n} + \cdots + t_{2,i}$
$= \alpha_2 T_0 + \left(\prod_{k=1}^{n}\alpha_k\right)^1 \alpha_2 T_0 + \cdots + \left(\prod_{k=1}^{n}\alpha_k\right)^{\left(\frac{i-2}{n}\right)}\alpha_2 T_0$
Following the calculations used for robot pair no. 1, this sum equals,

$$T_{R_2} < \frac{\alpha_2\left(\prod_{k=1}^{n}\alpha_k\right)^{\left(\frac{i-2}{n}+1\right)}}{\prod_{k=1}^{n}\alpha_k - 1}T_0 \tag{7}$$

Comparing the two expressions for the time to cover the last ellipse, $t_{2,i}$, and T_{f_2} from (6) T_0 can be calculated, $T_0 = \frac{\pi\beta_n t_{opt}\sqrt{(\beta_n v t_{opt})^2 - ||\mathcal{S}-\mathcal{T}||^2}}{4D\beta_1\left(\prod_{i=1}^{n}\alpha_i\right)^{\left(\frac{i-2}{n}\right)}}$. Substituting T_0 into (7) and simplification yields,

$$T_{R_2} < \frac{\pi\alpha_2\beta_n\left(\prod_{i=1}^{n}\alpha_i\right)t_{opt}\sqrt{(\beta_n v t_{opt})^2 - ||\mathcal{S}-\mathcal{T}||^2}}{4D\beta_1\left(\prod_{i=1}^{n}\alpha_i - 1\right)}.$$

Accordingly, for R_3, the sum of the search times is,

$$T_{R_3} < \frac{\pi\alpha_3\beta_n\left(\prod_{i=1}^{n}\alpha_i\right)t_{opt}\sqrt{(\beta_n v t_{opt})^2 - ||\mathcal{S}-\mathcal{T}||^2}}{4D\beta_2\left(\prod_{i=1}^{n}\alpha_i - 1\right)}.$$

Accordingly, generally,

$$T_{R_j} < \frac{\pi\alpha_j\beta_n\left(\prod_{i=1}^{n}\alpha_i\right)t_{opt}\sqrt{(\beta_n v t_{opt})^2 - ||\mathcal{S}-\mathcal{T}||^2}}{4D\beta_{j-1}\left(\prod_{i=1}^{n}\alpha_i - 1\right)}.$$

Next, in order to find the optimal multiplication factors, $\alpha_j's$, a new objective function which combines all the sums of times is formed,

$$\begin{aligned} T_{tot} &= T_{R_1} + T_{R_2} + \ldots + T_{R_n} \\ &< \frac{\pi\beta_n\left(\prod_{i=1}^{n}\alpha_i\right)t_{opt}\sqrt{(\beta_n v t_{opt})^2 - ||\mathcal{S}-\mathcal{T}||^2}}{4D\left(\prod_{i=1}^{n}\alpha_i - 1\right)} \cdot \\ &\quad \cdot \sum_{i=1}^{n}\frac{\alpha_i}{\beta_{i-1}}. \end{aligned}$$

Differentiating T_{tot} according to each of the α_j's, comparing each function to zero and finding the common roots yields,

$$\alpha_j = \frac{(n+1)^{\left(\frac{1}{n}\right)}\beta_{j-1}^{\frac{n-1}{n}}}{\prod_{k=2,k\neq j-1}^{n}\beta_k^{\frac{1}{n}}}, \qquad \beta_{j-1} = \left\{\begin{array}{ll} \beta_{j-1} & , j \neq 1 \\ \beta_n & , j = 1 \end{array}\right\}$$

■

In order to calculate a_0, b_0, a_p, b_p, it is necessary to solve the following set of equations, assuming $||\mathcal{S}-\mathcal{T}||, T_0$, and β_j are given:

$$\begin{aligned} a_p^2 &= ||\mathcal{S}-\mathcal{T}||^2/4 + b_p^2, \\ A_p &= \pi a_p b_p, \\ t_{jp} &\leq 4\frac{A_p}{\beta_j v D} + (||\mathcal{S}-\mathcal{T}||)/(\beta_j v), \\ t_{jp} &= \left(\prod_{k=1}^{n}\right)^{p-1}\left(\prod_{k=2}^{j}\right)T_0. \end{aligned}$$

Corollary 4.4: HMRBUG is complete.

Proof: The first important property established in Theorem 1, is that if the target $\mathcal{T}$ is reachable, *HMRBUG* will find a path to it. The second property is that *HMRBUG* will find that path in a finite and limited time. ■

V. EXECUTION EXAMPLE

In the following example depicted in Fig. 2, *HMRBUG* launches two pairs of robots, $1, 2$ to search for a path to the target in an office like environment. Each robot pair is initially assigned to a bounding ellipse, e_1, e_2 to execute *PBUG1* in it, and each robot in a pair is assigned to a different local direction, $Left, Right$: $1_L, 1_R, 2_L, 2_R$. At first, the robot pairs are moving directly toward the target, and as they encounter an obstacle they split, and each robot moves in its local direction. It can be observed that a part of the path is traversed by all the robots together at the same time, and the robots will move together as long as they are within the boundary of the first ellipse. While robot pair no. 2 is traversing the second ellipse, robot pair no.1 finishes traversing the obstacle's boundary which lies inside the first bounding ellipse and the ellipse itself. After meeting each other, the robots of pair no. 1 move toward the closest point to the target they encountered in their traverse, there they conclude that they cannot reach the target from ellipse no. 1. In Fig. 2(a) robot pair 1 is just about to execute *PBUG1* in ellipse no. 3, and robot pair 2 already meet on ellipse no. 2 and is on its way to the closest point to the target. While robot pair no. 1 is busy with its search in ellipse no. 3, robot pair no. 2 reaches the closest point to the target (Fig. 2(b)). Next, robot pair 2 is assigned to ellipse no. 4 and while searching in it robot pair no.1 meets on ellipse no.3 (Fig. 2(c)). While robot pair no. 2 searches in ellipse no. 4, robot pair 1 moves toward the closest point to the target (Fig. 2(d)) and from there it continues without any more obstacles in its way and reaches the target.

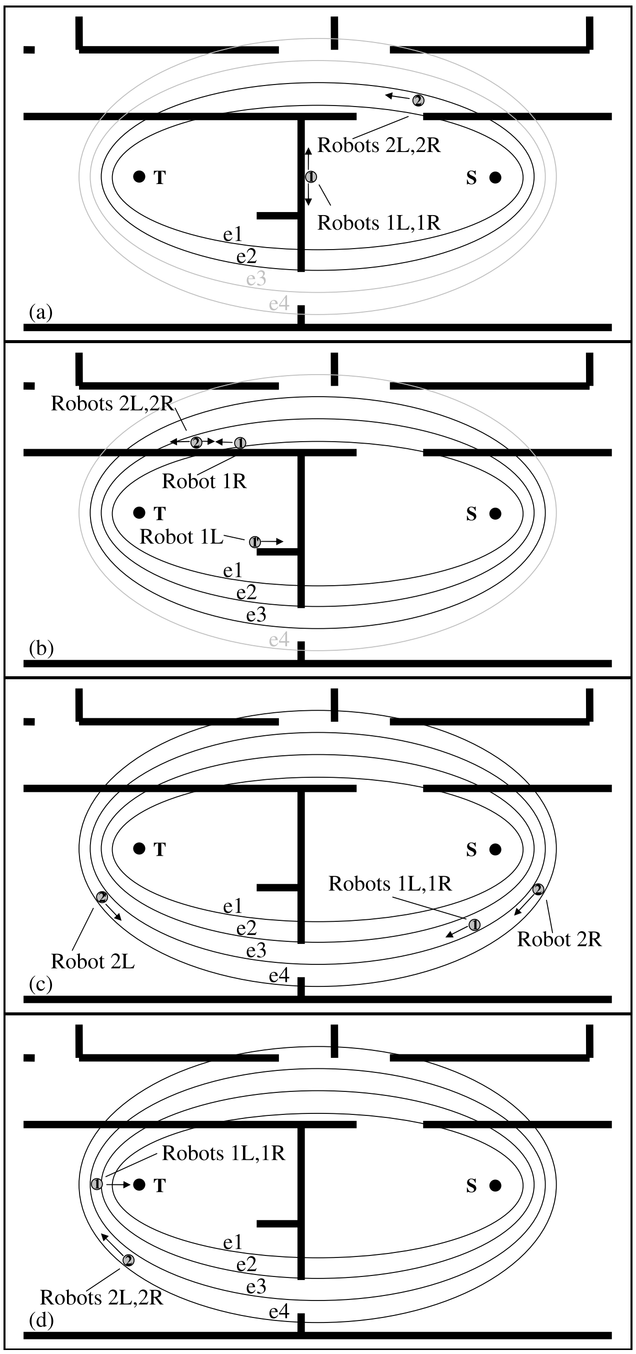

Fig. 2. Execution example of *HMRBUG*

VI. CONCLUSIONS AND FUTURE WORKS

We presented the problem of finding a path to a target whose position is known in an unknown and unbounded environment. *HMRBUG* , a novel algorithm for a group of velocity heterogeneous robots was introduced. *HMRBUG* performance was proved to be time competitive with a function quadratic in the optimal off-line solution. *HMRBUG* algorithm was proved to be complete and robust, and an execution example exhibits its main principle.

We are currently working on proving that the problem's lower bound is quadratic in the optimal off-line solution and then *HMRBUG* will be proved to be optimal since its performance will be within the same class. Future work includes using different velocities within each robot pair, and using communication between the robots to share information e.g. a common map of the obstacles in the environment and thus to enhance the performance. Allowing the robots not to follow the bounding ellipse after bypassing an obstacle, i.e., when the way directly towards the target is clear is a practical speedup that should be considered.

VII. ACKNOWLEDGMENTS

The authors would like to thank Yael Edan for her invaluable remarks and Yoav Gabriely for priceless discussions.

REFERENCES

[1] J. Jones, "Robots at the tipping point: the road to iRobot Roomba," *Robotics and Automation Magazine, IEEE*, vol. 13, no. 1, pp. 76 – 78, march 2006.

[2] J. Carsten, A. Rankin, D. Ferguson, and A. Stentz, "Global path planning on board the mars exploration rovers," in *IEEE Aerospace Conference*, Big Sky, Montana, USA, March 2007, pp. 1–11.

[3] S. Berman and Y. Edan, "Decentralized autonomous AGV system for material handling," *Int. Journal of Production Research*, April 2002.

[4] 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.

[5] 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.

[6] 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.

[7] J. Chen and L.-R. Li, "Path planning protocol for collaborative multi-robot systems," in *Proc. 2005 IEEE Int. Sym. on Comput. Intell. in Robotics and Automation*, Espoo, Finland, 2005, pp. 721–726.

[8] N. Trawny and S. I. Roumeliotis, "A Unified Framework for Nearby and Distant Landmarks in Bearing-Only SLAM," in *Proc. IEEE Int. Conf. on Robotics and Automation*, May 2006, pp. 1923–1929.

[9] 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. of the 2002 IEEE Int. Conf. on Robotics and Automation (ICRA '02)*, vol. 1, May 2002, pp. 961 – 967.

[10] Y. Gabriely and E. Rimon, "CBUG: A quadratically competitive mobile robot navigation algorithm," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2005, pp. 954–960.

[11] S. Sarid and A. Shapiro, "Classifying the Multi Robot Path Finding Problem into a Quadratic Competitive Complexity Class," *Annals of Math. and Artificial Intell.*, vol. 52, no. 2-4, pp. 169–203, April 2008.

[12] V. J. Lumelsky and A. A. Stepanov, "Path-Planning Strategies for a Point Mobile Automaton Moving Amidst Unknown Obstacles of Arbitrary Shape," *ALGORITHMICA*, vol. 2, pp. 403–430, 1987.

[13] S. Sarid and A. Shapiro, "H-MRSTM: A Competitive Search Algorithm for Heterogeneous Group of Robots," *Int. J. of Factory Automation, Robotics and Soft Computing*, no. 3, pp. 51–58, 2009.