Rendezvous in Planar Environments with Obstacles and Unknown Initial Distance*

Deniz Ozsoyeller† and Andrew Beveridge‡ and Volkan Isler§

November 27, 2018

Abstract

In the *rendezvous search problem*, two or more robots at unknown locations should meet somewhere in the environment as quickly as possible. We study the symmetric rendezvous search problem in unknown planar environments with polygonal obstacles. In the symmetric version of the problem, the robots must execute the same rendezvous strategy. We consider the case where the initial distance between the robots is unknown, and the robot is unaware of its and the other robots' locations in the environment. We first design a symmetric rendezvous strategy for two robots and perform its theoretical analysis. We prove that the competitive ratio of our strategy is $O(d/D)$. Here, d is the initial distance between the robots and D is the length of the sides of the square robots. In unknown polygonal environments, robots should explore the environment to achieve rendezvous. Therefore, we propose a coverage algorithm that guarantees the complete coverage of the environment. Next, we extend our symmetric rendezvous strategy to n robots and prove that its competitive ratio is $O(d/(nD))$. Here, d is the maximal pairwise distance between the robots. Finally, we validate our algorithms in simulations.

Keywords: Multi-robot systems, planning in complex environments, rendezvous search, coverage planning.

1 Introduction

In the rendezvous search problem, n players want to meet as quickly as possible. However, they cannot communicate over long distances and they are unaware of their locations in the environment. Examples of this fundamental problem include parachutists who must regroup after landing in different positions, and a mother who loses her child while wandering the aisles of a supermarket. Rendezvous search has important applications in search and rescue operations. For example, Thomas and Hulme [2] study the rendezvous of a rescue helicopter looking for a walker who is lost in the desert and wants to be found.

*A preliminary version of this paper appeared in [1]. This version includes the proofs omitted in [1] and a new algorithm for the multi-robot case presented in Section 6.

†Department of Computer Engineering, Final International University, Kyrenia, North Cyprus, Turkey
‡Department of Mathematics, Statistics and Computer Science, Macalester College, Saint Paul, MN, USA
§Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA

Rendezvous may be part of a more complicated task. For example, mobile robots are used to explore a dangerous environment to gather data. In situations where the topography of the environment is inconvenient, these robots might not be able to communicate over long distances or without unobstructed line-of-sight between them. A rendezvous algorithm can be used to conglomerate the collected data.

There are two primary versions of the rendezvous search problem. In the symmetric version (such as the parachutists), robots follow the exact same rendezvous strategy. In asymmetric rendezvous search, the players can choose separate roles in advance and execute distinct strategies. For example, one can remain stationary, while the other actively searches; this strategy is often called "wait for mommy."

The most critical parameter in robot motion is the physical distance traveled, since movement is energy-intensive. Therefore, we here aim to minimize the expected distance traveled by the robot. We use the competitive analysis to measure the performance of our symmetric rendezvous strategy. The efficiency of a rendezvous strategy $\mathcal{S}$ between two robots is often measured by its competitive ratio

$$\max_{x,y \in Q} \frac{S_1(x,y) + S_2(x,y)}{d(x,y)}. \tag{1}$$

Here x and y are initial locations of robots in an environment Q, and $S_i(x,y)$ denotes the (expected) distance traveled by robot i before rendezvous, $i = 1, 2$, and $d(x,y)$ is the length of the shortest path connecting x and y. A strategy is said to be *competitive* if its competitive ratio is a constant. In equation (1), the competitive ratio is given with respect to the total distance traveled. Other measures such as the maximum distance traveled can also be used.

We study the symmetric rendezvous search problem in planar environments populated with polygonal obstacles. Like covert parachutists who land in the dark, we assume that the robots start with very little information: they do not have a map of the environment, and they are unaware of the locations of each other, their own locations and the initial distance between them. Starting with two robots, we design and analyze a symmetric randomized algorithm. While obtaining the bounds for the rendezvous algorithm, we propose a coverage algorithm that guarantees the complete coverage of the environment and analyze its performance. We then extend our algorithm for multiple robots. The competitive ratio of our strategy is $O(d/D)$ for two robots, where d is the initial distance between the robots and D is the length of the sides of the square robots. For n robots, the competitive ratio of our strategy is $O(d/(nD))$, where d is the maximal pairwise distance between the robots. Finally, we verify our theoretical results with simulations performed in sample environments.

This paper is organized as follows. In Section 2, we survey the literature on rendezvous search, exploration and coverage problems. We present our rendezvous strategy in planar environments with obstacles in Section 3 and the analysis of our coverage algorithm in Section 4. In Section 5, we compute the performance of our symmetric rendezvous strategy. The extension of our strategy to n robots is given in Section 6. In Section 7, we present the results from the simulation experiments conducted in different environments. Finally, we provide the concluding remarks in Section 8.

2 Related Work

The rendezvous search problem generalizes the *linear search problem*, also known as the 2-lane *lost cow problem*. In this formulation, a near-sighted cow tries to find the only gate in a long, straight fence. The gate is located at an unknown initial distance d, either to the left or the right of the cow. This linear search problem was originally solved in [3] and that solution was rediscovered in [4].

In the deterministic online *Spiral LostCow algorithm*, the cow alternately searches to its left and then to its right starting from its initial location $x = 0$. Spiral LostCow is a 9-competitive algorithm, which is the best possible performance for a deterministic online algorithm. Kao et al. [5] introduced the randomized algorithm *SmartCow*, which follows the same zig-zag strategy except that it uses randomization at beginning of the algorithm to choose the initial search distance and the order of the paths explored. SmartCow guarantees competitive ratio of 4.591, which is the best ratio that can be achieved by any randomized algorithm.

In the m-lane lost-cow problem, the cow at the meeting point of m concurrent rays searches a gate at d distance on one of the rays. The best algorithm is a generalization of the Spiral LostCow algorithm, which is $1+2m^m/((m-1)^{m-1})$-competitive and proved to be optimal by Baezayates et al. [4]. For this problem, Dasgupta et al. [6] showed that there is no depth-first iterative deepening (DFID) strategy which is optimal except when $m = 2$.

The rendezvous search problem on the line has been well studied for both symmetric [7, 8, 9, 10, 11, 12, 13, 14] and asymmetric [15, 13, 16, 17, 18] versions. Alpern [7] introduced a symmetric strategy for the known $2d$ initial distance between two players. Over the years, the analysis of this algorithm has been improved, so that guaranteed competitive ratio has been decreased from Alpern's initial value of 5 to 4.5678 by Anderson and Essegaier [8], to 4.4182 by Baston and Gal [13], and to 4.39306 by Uthaisombut [10]. Baston and Gal [13] considered a symmetric rendezvous formulation in which the probability distribution of the initial distance $2d$ between two players is unknown except its expected value, which is $E[2d] = \mu$. They provide a symmetric algorithm with expected meeting time 13.325μ, corresponding to competitive ratio 26.650. Ozsoyeller et al. [14] proposed a symmetric rendezvous algorithm at an unknown (and arbitrary) initial distance between two robots, that has a competitive ratio of 17.686 for total distance traveled and a competitive ratio of 24.843 for total time.

Another well-studied rendezvous problem is the rendezvous of two mobile agents that have to meet at some node of a graph [19, 20, 21, 22, 23, 24]. Anderson and Weber [25] propose a randomized strategy (AW) for the symmetric rendezvous of two players that are initially placed at two distinct vertices on a completely connected graph with n vertices. The players either stay at their initial vertices or tour the other $n-1$ vertices in random. Weber [26] shows that AW-strategy is optimal for $n = 3$.

In this paper, we study the symmetric rendezvous of two and n robots in planar environments with polygonal obstacles. The most common variant of the rendezvous search problem in obstacle-free planar environments is the asymmetric case [27, 28, 2]. Anderson and Fekete [28] study the asymmetric rendezvous of two players in a planar environment. They consider three cases: (1) when one of the players is placed at an initial position chosen equiprobably from a finite set of points, (2) when both players know the initial distance between them, but not the direction in which they should travel, and (3) when the first player knows the location of the second player but the second player only knows the initial distance to the first player.

Some strategies are designed around a set of pre-determined meeting point options. Roy and Dudek [29] consider the rendezvous of two robots with limited vision, exploring an unknown environment with potential rendezvous points called *landmarks*. Anderson and Weber [25] study the symmetric rendezvous of two friends who separate from each other in a building or a mall with n possible rendezvous locations. The minimal expected time of their rendezvous strategy is 2 for $n = 2$ and $8/3$ for $n = 3$.

Researchers have also explored rendezvous of agents moving on the n dimensional lattice. They consider the rendezvous time for two agents starting at a known initial distance 2. The expected number of steps before meeting is called the rendezvous value, denoted by $R^a(n)$ (asymmetric strategies) and $R^s(n)$ (symmetric strategies). Alpern and Street [30] give a one-dimensional strategy with $R^a(1) = 3.25$, and Alpern and Baston [27] give a two-dimensional strategy with $R^a(2) = 6.16$. For general n, Alpern and Street [30] show that the rendezvous times satisfy $R^a(n) \leq 8n/3$ and $R^s(n) \leq 56n/9$.

Collins et al. [31] study the asynchronous rendezvous of two location-aware agents with limited vision in two dimensional environments without obstacles. Here, asynchronous means that an adversary can interfere with the timing of each action. The cost of their rendezvous algorithm, which is measured by the total distance traveled by the agents, is $O(d^2)$, where d is the initial distance between the agents. The asynchronous rendezvous of two anonymous agents on a $\delta > 0$ dimensional grid is studied in [32]. The agents are initially located at arbitrary positions separated by distance d. Each agent knows its position but not the other agent's position. The proposed algorithm requires the agent to travel a path length of $O(d^\delta \text{poly} \log d)$ to meet.

The *gathering* problem is another variant of rendezvous in which two or more identical robots use the same algorithm to meet in finite time, and this meeting point is not determined in advance. In gathering problem, robots operate in Wait−Look−Compute−Move cycles. Thus, the robots decide on their moves on viewing their surroundings and analyzing the configuration of robot locations. Each robot starts in a waiting state (*Wait*). It then wakes up and observes the positions of all other robots within its visibility range which can be either limited or unlimited (*Look*). The robot then calculates its destination point based on the observed locations of the robots (*Compute*). In the final step, it moves to the computed destination point (*Move*); after the move it goes back to a waiting state. In comparison, rendezvous search assumes extremely limited sensing capabilities, for example collision detection.

The gathering problem has been studied under models that differ in synchronicity, visibility, dimension or the obliviousness of the robots. In an asynchronous model (ASYNCH), there is no common notion of time, so the operation cycle is not performed simultaneously for each robot [33, 34, 35, 36, 37, 38, 39, 40, 41, 42], whereas a common clock is used in the synchronous model (SYNCH). The Semi-synchronous model (SSYNCH) is similar to SYNCH model, except that only some of the robots are activated at each clock tick. Suzuki and Yamashita [43] proved that it is impossible to achieve gathering of two oblivious autonomous mobile robots that have no common sense of orientation under the SSYNCH model, in a finite time.

In the unlimited visibility setting, the robots can sense the entire space [40, 44, 43]. This case is easier, however less realistic than the limited visibility case [45, 46], in which a robot can only locate the robots within its visibility range. Robots are usually anonymous and treated as points which are dimensionless objects that do not obstruct each other's visibility or movement. In contrast, Jurek et al. [39] represent the robots by unit disks. Another attribute that differs in the researches is the obliviousness of the robot. Oblivious robots do not remember their previous actions or the previous positions of the other robots [36, 39, 33, 37], while non oblivious robots can use this information to determine their next step [44].

Another variation of the rendezvous problem is the *coalescence* problem in which the goal is to form a single connected network between mobile robots independently searching for peers. The term coalescence emphasizes that the connected component spreads as more robots join it. Poduri and Sukhatme [47] study the coalescence problem with the robots that do not know about the environment or positions of other robots. They show that as the number of robots N increases, coalescence time decreases as $O(1/\sqrt{N})$ and $\Omega(\log(N)/N)$.

Finally, we note that there are other important rendezvous problems, including designing local control strategies [48, 49, 50, 51, 52, 53]. The rendezvous problem in control theory concerns robot tracking and navigation toward a moving object (target) where the agents can observe one another's state. These studies emphasize the control-theoretic aspects of the problem, such as combining the kinematics equations of the robot and the target. In this paper, we study the rendezvous search problem. The lack of state information differs this problem from the rendezvous tracking and navigation problem.

2.1 Exploration and Coverage Problems

The rendezvous problem is related to the exploration problem and the coverage problem. Typically the goal of exploration is to generate a map of the environment using the measurements acquired from the sensors on the robots. In the coverage problem, the map of the environment may be known or unknown and the robot aims to visit every point in the environment (with its sensors).

In the studies where an unknown environment is explored with unlimited vision, a vision sensor is assumed to see an object at any distance. In an influential paper, Deng et al. [54] consider exploration with unlimited vision, providing a 2-competitive algorithm for an obstacle-free rectilinear environment. This value was improved by Hammar et al. [55] to 5/3. Kleinberg [56] proves that the lower bound for any deterministic exploration algorithm is 5/4, and provides a randomized algorithm which achieves that competitive ratio. Turning to environments with polygonal obstacles, Deng et al. [54] showed that there is no randomized competitive strategy for a polygon with an arbitrary number of polygonal obstacles, even if polygons are parallelograms. Albers et al. [57] later showed that no deterministic or randomized online algorithm for exploring a two dimensional environment with n rectangles can be better than $\Omega(\sqrt{n})$-competitive. Interestingly, a competitive ratio of $\Theta(min(k, \sqrt{k\bar{\alpha}}))$ is achieved for convex polygonal obstacles [58], where $\bar{\alpha}$ and k are the average aspect ratio and the number of objects, respectively. The aspect ratio of a convex polygonal object O is defined as R/r where R is the radius of the smallest circle that circumscribes O and r is the largest circle that inscribes O. The average aspect ratio is the sum of the aspect ratios of the k objects divided by k.

Limited vision exploration of polygons has also been well-studied. Gabrielly and Rimon introduce Spanning Tree Covering (STC) algorithms in [59, 60], which guarantee near-optimal covering paths, provided that the perimeter of the obstacles is small compared to the area to be covered. Fekete and Schmidt [61] study exploration by discrete vision robot in rectilinear polygons without holes and obstacles, giving an online algorithm with competitive ratio of

$O(\log A)$, where the aspect ratio A is the ratio of the maximum and minimum edge length of the polygon.

Gabriely and Rimon [62] present an online navigation algorithm ($SAD1$) for a mobile robot with planar body of size D and limited vision to explore an unknown environment to find a target T. Their approach follows a strategy similar to our rendezvous algorithm. The robot centered at S selects a disk of radius R_0 and covers the parts of the disk that is reachable from S. They partition the environment into a grid of D sized cells. Grid is consisted of free, occupied or partially occupied cells. Robot executes a simple DFS Algorithm on the free cells for coverage tour. Their coverage pattern differs from ours, in that they assume that the cost of covering a partially obstructed cell is the same as covering a free cell. We take these partially occupied cells into account, and guarantee complete coverage of the environment.

Ghosh et al. [63] study the exploration of an unknown cellular room. The room is defined as a P polygon that consists of the square cells of a grid with obstacles. Their *CellExplore* strategy explores P in at most $C + \frac{1}{2}E + H - 3$ steps, where P contains C square cells, E edges and H obstacles. Czyzowicz et al. [64] consider exploration of an unknown terrain with impassable obstacles by a mobile robot presented as a point in the plane. Both the terrain and the obstacles are modeled as arbitrary polygons. They give an exploration algorithm with a complexity of $O(P + A + \sqrt{Ak})$, where A is area of the terrain, P is the perimeter of the terrain (including the perimeter of the obstacles) and k is the number of obstacles. Similar to our rendezvous strategy, their algorithm tiles the area with squares and traverses the resulting grid using DFS. In comparison, our strategy has a smaller cell coverage cost.

Turning to coverage problems, Easton and Burdick [65] study the boundary coverage problem in a two-dimensional environment by k identical holonomic point robots. Each robot can accurately localize itself and has an omnidirectional "inspection sensor". Choi et al. [66] uses boustrophedon motions, boundary-following motions, and the Theta* algorithm known as B-Theta* to tackle the online complete coverage task of cleaning robots in unknown workspaces with arbitrarily-shaped obstacles. For known two-dimensional environments populated with arbitrary obstacles, Xu et al. [67] propose a polynomial time complete coverage algorithm which is based on the boustrophedon cellular decomposition technique. They also present the extensions of this algorithm for non-holonomic vehicular dynamics. In [68], Yao presents a modified sweeping line strategy for complete coverage problem in a known environment that aims to minimize the number of non-effective relocation moves. A non-effective move is described as the movement of the robot in the area that has been fully covered before.

There are two classes of coverage problems: single coverage and repeated coverage. In the single coverage problem, the goal is to cover all the accessible points at least once, whereas, it is to cover them repeatedly in the repeated coverage problem. Fazli et al. [69] study the single coverage problem for a known environment by multiple robots with limited visibility. Their strategy locates a set of static guards on the target area and builds a graph based on the constrained Delaunay triangulation. Later work [70] addresses the problem of repeated coverage of a polygonal target area by a team of limited-visibility robots. Strimel and Veloso [71] investigate the coverage planning problem for a single robot with a fixed energy capacity. A battery constrained sweep algorithm (BC Sweep), which extends the boustrophedon cellular decomposition coverage algorithm is introduced.

Zheng et al. [72] study a multi-robot coverage algorithm which employs a balanced tree cover. The cover time of the proposed Multi-Robot Forest Coverage algorithm is at most eight times larger than optimal. Agmon et al. [73] construct a coverage spanning tree for both online and offline coverage. They give a polynomial time offline algorithm and a linear time online algorithm for communication-enabled robots. Khan et al. [74] study an online approach for complete coverage path planning of mobile robots in an unknown workspace. They use a sequence of boustrophedon motions, where the next starting point is determined using the current knowledge of the environment map. Shnaps and Rimon [75] consider coverage of an unknown planar environment with obstacles by a mobile size D robot tethered by a cable of length L. Their online tethered coverage (TC) algorithm has competitive ratio $2L/D$.

3 Symmetric Rendezvous in Planar Environments with Obstacles

In [1], we design a symmetric rendezvous strategy ($\mathcal{SP}$) for obstacle-free planar environments. The robots have very limited information: they do not know their own position, the position of the other robot, nor the initial distance between them. We consider the synchronous case, where the robots share a common clock and start searching at the same time. We adapt the $\mathcal{SP}$ algorithm for planar environments with obstacles by providing a motion path that can handle obstacles. The adapted algorithm is called $\mathcal{SP}_o$. We make the following assumptions.

1. The environment is unknown.
2. The obstacles in the environment are polygonal.
3. A robot moves via translation (it does not rotate).
4. A robot measures $D \times D$ square, and is equipped with obstacle detection sensors.
5. Two robots meet when their $D \times D$ squares touch one another.

3.1 Symmetric Rendezvous Search Algorithm

We present our motion planning algorithm using a configuration-space C in which the $D \times D$ square robot is represented as a point robot. We construct C-obstacles by taking the Minkowski sum of every obstacle with a $D \times D$ square. We assume that this configuration space is connected, making rendezvous achievable. We consider the C-obstacles to be open sets (omitting their boundaries) to accommodate the movement of the robot between two obstacles with a distance D opening between them. In our configuration space, the point robot can traverse the one-dimensional boundary separating these open C-obstacles. Finally, we note that if the point robot is within distance D of a robot in the configuration space, then the actual robots will meet.

The algorithm proceeds in rounds indexed by $i \in \mathbb{Z}^+$. At the beginning of each round, the robot flips a weighted coin to decide whether to move or wait in that round. *If the robot tosses heads*, then it draws a disk $Disk_i$ of radius r^i centered at its initial location where r is the expansion radius of the robot (optimized in Theorem 7). The robot discretizes the continuous area in $Disk_i$ into a grid of square $D \times D$ cells. Then it covers every grid cell in $Disk_i$ using the coverage algorithm CA described below. *If the robot tosses tails*, then it waits long enough so that another robot could have covered $Disk_i$, which keeps the rounds in sync. We calculate this wait time in Section 4. Algorithms 1 and 2 present $\mathcal{SP}_o$ and CA, respectively.

Algorithm 1 : $\mathcal{SP}_o$, Symmetric rendezvous search algorithm in environments with obstacles.

Input: C-obstacle with removed boundary of the expanded obstacles.
Input: D.

```
r ← 1.225
i ← 0
head ← 0
tail ← 1
coin ← random from {0, 1}
while Check_Rendezvous() ≠ true do
    if coin = head then
        start_time ← Current_Time()
        N_i ← πr^{2i} / D^2    /* Number of cells inside the disk with radius r^i */
        grid_size ← ⌈√N_i⌉
        /* C_center is the cell the robot is initially located */
        Create_Grid(C_center, grid_size)
        CA(C_center, Grid)    /* Coverage Algorithm */
        Wait(4 N_i D + start_time - Current_Time())
    else if coin = tail then
        Wait(4 N_i D)   /* Waiting time is proved in Theorem 6 */
    end if
    i ← i + 1
    coin ← random from {0, 1}
end while
```

We describe the CA procedure in more detail. Create the dual graph of the grid cells in $Disk_i$, connecting two cells when their common boundary is not fully obstructed. For example, in Figure 1, Cell E neighbors Cells B, D, H and F, and each of these shared boundaries intersects obstacles. We discover a spanning tree of the dual graph by performing a depth first search, visiting the children of the current vertex in counterclockwise order. Our motion plan within a cell employs counter-clockwise boundary following (CCWBF) to trace the four sides of the cell. We assume that the robots have a relative (local) localization mechanism (e.g. odometry and camera) with which they can determine whether they have reached the boundary of the current cell. We transition into a child cell at first opportunity, recursively following CCWBF in that cell and its children. Except for the initial cell, each cell has a single entrance point, and up to three exit points. In general, we return to a parent cell only after traversing all four sides of its child cell (and of the child's descendents). We return to the parent cell at the same point that we exited, and continue the CCWBF traversal of that cell. Note that we must keep track of distance traveled, so that we do not re-enter a cell that has already been partially explored.

Figure 1 illustrates the CA trajectory in 3×3 environments with and without obstacles. Recall that the environment is unknown to the robot, so it discovers the environment (and its

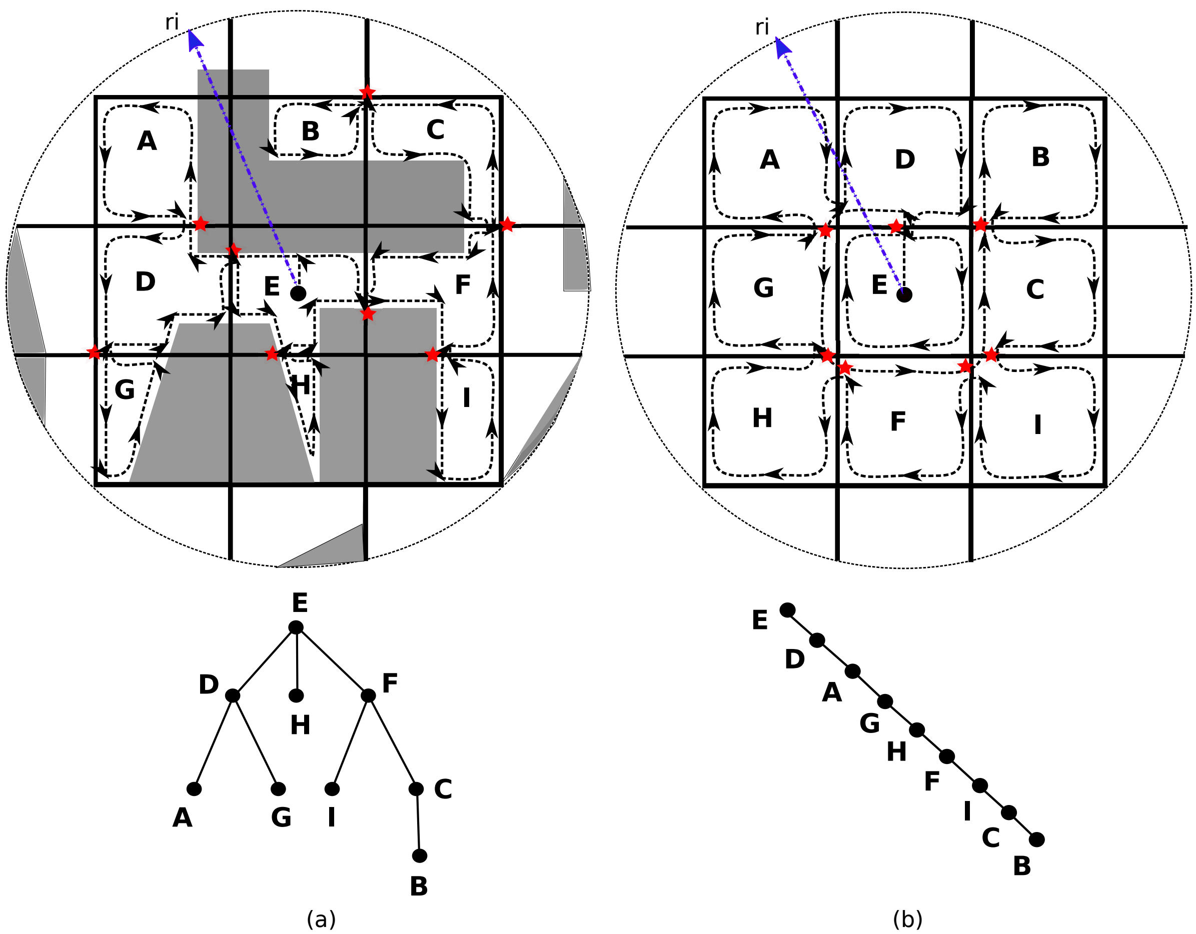


Figure 1: (a) Sample run of the coverage algorithm in an environment with obstacles. (b) Sample run of the coverage algorithm in an environment without obstacles. The corresponding DFS Trees show the order that the cells are visited in these environments. In round i, the robot covers 3×3 grid inside a disk of radius r^i. The robot starts in Cell-E. An arrow denotes the direction of movement and a star denotes the first entrance of a cell.

Algorithm 2 : CA($C_{current}$, G)

// C_{center} : the cell the robot is initially located
// $C_{current}$: the cell the robot is currently moving in
// C_{parent} : the cell from which the robot enters $C_{current}$
// C_{child} : one of the neighbor cells of $C_{current}$

Follow the obstacle/cell boundary of $C_{current}$ in a counter clockwise fashion
$i \leftarrow Get\_Id(C_{current})$
if $G[i].entrance = G[i].getCurrPos()$ **then**
  $G[i].covered \leftarrow true$
  if $i = C_{center}$ **then**
    return
  else
    $C_{current} \leftarrow G[i].parent$
    Enter $C_{current}$
  end if
else
  if $Hit\_Boundary(C_{child})$ **and** $C_{child} \neq C_{parent}$ **then**
    $i \leftarrow Get\_Id(C_{child})$
    $G[i].parent \leftarrow C_{current}$
    $C_{current} \leftarrow C_{child}$
    Enter $C_{current}$
    $G[i].entrance \leftarrow G[i].getCurrPos()$
  end if
end if
$CA(C_{current}, G)$

obstacles) as it methodically explores each cell. The cells are indexed between A-I and the robot is initially located at Cell-E. Arrows show the direction of movement in a cell and the stars show *entrance* of each cell. The DFS trees in Figure 1 show the order that the robot visits the grid cells. In Figure 1(a), the robot first moves in upward direction in Cell-E and it turns left at the upper obstacle to start CCWBF. Upon encountering the boundary of Cell-D, the robot enters that cell and follows CCWBF until it reaches the boundary of cell-A. The robot visits all four boundaries of Cell-A (since cell-A has no children) and the robot comes back to its *entrance* of Cell-A. The robot then continues CCWBF in Cell-D until it reaches the boundary of Cell-G, where it moves to recursively cover that cell. All the other grid cells are covered in a similar way by the robot.

4 The Analysis of the Coverage Algorithm

In this section, we give an upper bound on the coverage time of CA by bounding the time that the robot spends in a given cell, and then bounding the number of cells in $Disk_i$. We define the *regions* in a cell C_i to be the maximal two-dimensional open subsets of C_i after removing the obstacles in the configuration space C. First, we show that the geometry of C-

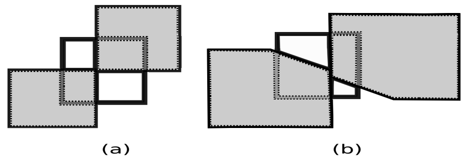


Figure 2: Examples of a cell split into two regions by different C-obstacles, which are shown lightly shaded. (a) The two regions are connected by a point. (b) The two regions are connected by a line segment.

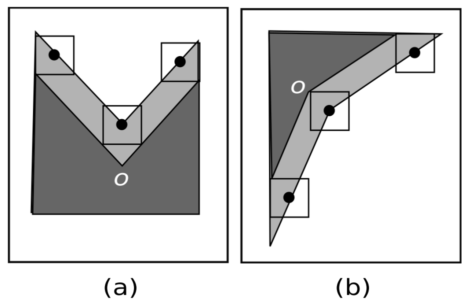


Figure 3: Concave corner of obstacle P (dark) and its corresponding corner in C-obstacle P' (dark plus light). (a) The edges meeting at O are in adjacent quadrants. (b) The edges meeting at O are in opposite quadrants.

obstacles ensures that a cell C_i can be split into at most two disjoint regions; Figure 2 shows two examples of this phenomenon. Next, we argue that covering a cell C_i requires traveling distance of at most $4D$.

4.1 At Most Two Regions Per Cell

We prove that the C-obstacles create at most two regions per cell. We start by characterizing the possible corners of an obstacle P and its corresponding C-obstacle P'. We assume the corner of P' is located at the origin, so that the environment splits naturally into four quadrants of the plane which meet at the current corner. Moreover, we assume that the robot traverses the boundary of P in a counter-clockwise fashion.

Recall that we construct C-obstacle P' by taking the Minkowski sum of the obstacle P with the translating $D \times D$ square robot. Horizontal and vertical edges are swept by an edge of the robot. Other edges are swept by a single corner of the translating robot. The corners of the original obstacle P can be convex (interior angle $0° < \theta < 180°$) or concave (interior angle $180° < \theta < 360°$) . We show that the angle at a corner of a C-obstacle is never acute.

Lemma 1. *The interior angle of a C-obstacle corner always satisfies $\theta \geq 90°$.*

Proof. First, suppose that original corner O of obstacle P is concave. If the edges that meet at O are in adjacent quadrants, then the corresponding corner of C-obstacle P' is also concave with the same angle (Figure 3(a)). Similarly, we obtain a matching concave corner of P' when the the edges are in opposite quadrants (Figure 3(b)). Next, suppose that the corner is convex. If both of the edges of the original corner O of obstacle P are in the same quadrant, then the

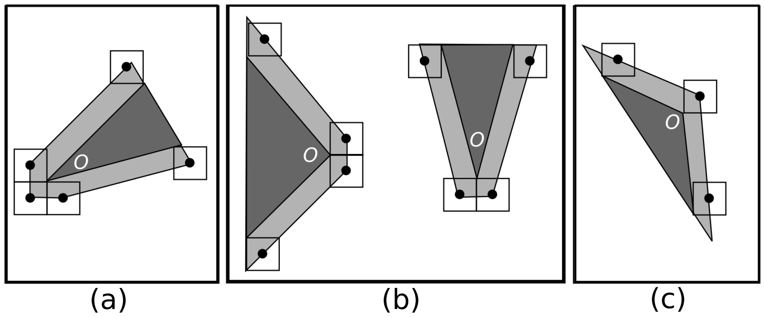


Figure 4: Convex corner of obstacle P (dark) and its corresponding corner in C-obstacle P' (dark plus light). (a) The edges meeting at O are in the same quadrant. The boundary of the C-obstacle has a right angle near O, where a vertical side and a horizontal side of length D meet. (b) The edges meeting at O are in adjacent quadrants. The C-obstacle has a side of length $D/2$ near O. (c) The edges meeting at O are in opposite quadrants.

angle must be acute. The C-obstacle P' has three corners corresponding to O: one right angle flanked by two obtuse angles (Figure 4(a)). If the edges are in adjacent quadrants (with an acute, right or obtuse angle), then the C-obstacle has two obtuse corners corresponding to O (Figure 4(b)). Finally, if the two edges are in opposite quadrants (so that the angle is obtuse), then the C-obstacle has a matching obtuse corner (Figure 4(c)). □

Next, we confirm that a C-obstacle has guaranteed minimum vertical and horizontal lengths.

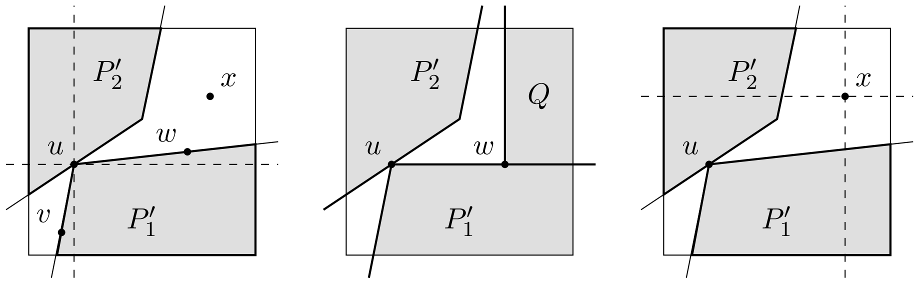
Proposition 2. *Consider any C-obstacle P' along with a point u on the boundary of P'. Let ℓ_1 and ℓ_2 be the vertical and horizontal lines through u, and for $i = 1, 2$ let $v_i \in \ell_i$ be the furthest point from u such that $v \cap P' \neq \emptyset$. Then $\mathrm{dist}(u, v_i) \geq D$ for $i = 1, 2$.*

Proof. If ℓ_i passes through the original obstacle then there is a line segment s_i of ℓ_i which lies inside the obstacle. The line segment connecting u and v includes the segment s and its Minkowski sum with the square robot, which extends the endpoints of s by at least $D/2$, so $\mathrm{dist}(u, v) \geq D$.

If ℓ_1 (resp. ℓ_2) does not pass through the original obstacle, then the point u must pass through a vertical (resp. horizontal) edge of the C-obstacle P'. Such edges are only introduced when the corner of the obstacle is one of types illustrated in Figure 4 (a) or (c), where the robot sweeps the original corner with one of its sides of length D. □

Lemma 3. *The C-obstacles split each grid cell into at most two regions.*

Proof. Recall that a region is a maximal two-dimensional open set in the configuration space. Suppose that C-obstacles P_1' and P_2' touch within our grid cell, creating two disjoint regions. Let u be a point on the boundary of both P_1' and P_2'. By Proposition 2, each obstacle contains both a vertical line segment and a horizontal line segment at u with length at least D, as shown in Figure 5(a). If another C-obstacle Q touches P_1' (resp. P_2'), then Proposition 2 forces



(a) (b) (c)

Figure 5: (a) P_1' and P_2' are C-obstacles that touch at point u, splitting the cell into two regions. (b) If C-obstacle Q touches P_1' at point w, then the boundary must be horizontal and shared with Q, so there are still only two regions. A similar statement holds for point v. (c) Two additional C-obstacles cannot meet at x since one or both would intersect an existing C-obstacle.

Q to share a horizontal or vertical side with P_1' (resp. P_2'), so there are still only two regions in the cell, as shown in Figure 5(b). Finally, we cannot have two additional objects touching within a region because Proposition 2 forces one or both of those obstacles to intersect an existing obstacle. □

4.2 Analysis of the Coverage Algorithm

In this section, we prove that the coverage algorithm guarantees complete coverage of $Disk_i$. We also establish the upper bound of the algorithm based on the distance traveled by the robot.

We next find the bound for the cost of covering a cell. The small size of our $D \times D$ cell $\mathcal{C}$ and the nature of our C-obstacles ensure that we can bound the distance traveled with $4D$, the perimeter of the cell. Observe that Proposition 2 guarantees that a C-obstacle P is at least $D \times D$, so it must extend beyond $\mathcal{C}$.

Lemma 4. *Let S be the connected piece of the boundary of the C-obstacle P that enters cell $\mathcal{C}$ at point $a = (a_1, a_2)$ and leaves cell $\mathcal{C}$ at point $b = (b_1, b_2)$. The length of S is at most $|b_1 - a_1| + |b_2 - a_2|$.*

Proof. Without loss of generality, assume that $a_1 \leq b_1$. We claim that the boundary S is

1. a horizontal line segment if $a_2 = b_2$,
2. a vertical line segment if $a_1 = b_1$,
3. piecewise linear and monotone nonincreasing if $a_2 < b_2$, and
4. piecewise linear and monotone nondecreasing if $a_2 > b_2$,

where "monotone" here also allows for vertical line segments. We can then use the corners of P to partition S into line segments (as shown in Figure 6), and use the Pythagorean theorem (repeatedly) to conclude that the length of S is at most $|b_1 - a_1| + |b_2 - a_2|$.

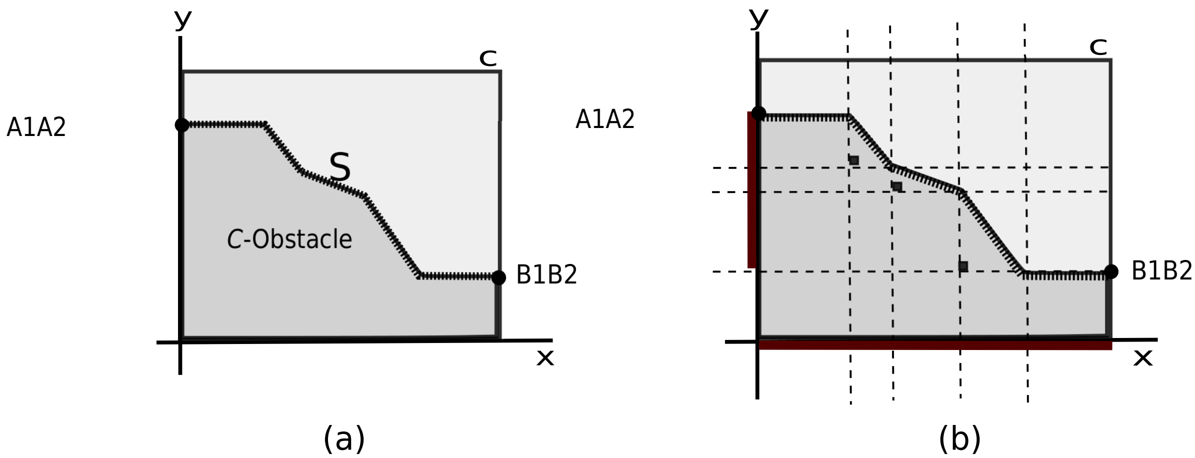


Figure 6: (a) The C-obstacle boundary S between $a = (a_1, a_2)$ and $b = (b_1, b_2)$ (b) Repeated use of the Pythagorean Theorem shows that the length of S is at most $|b_1 - a_1| + |b_2 - a_2|$.

We now prove our four-part claim. As shown in the proof of Lemma 1, the slopes of the sides of the C-obstacle match the slopes of the original obstacle, except at the extreme points, where we add a horizontal side and/or a vertical side. To change from negative slope to positive slope (or vice versa) on a C-obstacle, we must encounter an extreme point of the original obstacle. This creates a horizontal or vertical C-obstacle side of length at D between them, as shown in Figure 4(a). Therefore, the second side is too far from the first side to appear in the current cell. This proves claims 3 and 4.

By a similar argument, we cannot travel up one vertical side and down another (or rightward on one horizontal side and leftward on another), since such sides of a C-obstacle must be separated by distance D. This proves claims 1 and 2. □

Lemma 5. *The cost (distance traveled) of covering a cell is* $T_c \leq 4D$.

Proof. Lemma 3 explains what a cell with multiple obstacles can look like. We cover the cell using counter-clockwise boundary following (CCWBF). We alternately move along the cell boundary and the boundary of a C-obstacle. Lemma 4 shows that following a C obstacle from cell boundary point a to cell boundary point b, the path we take is no longer than following the (obstructed) counter-clockwise cell boundary from a to b instead. □

Theorem 6. *The time to cover a* $Disk_i$ *is* $T_i < 4N_iD$ *where* N_i *denotes the number cells inside* $Disk_i$.

Proof. From Lemma 5, we prove that the coverage cost of a cell is $\leq 4D$. Therefore, time to cover a $Disk_i$ is bounded by $4N_iD$. N_i is bounded by $\pi r^{2i}/D^2$. Placing N_i into the bound for the coverage time of $Disk_i$ gives $T_i < 4\pi r^{2i}/D$. □

5 The Analysis of the $\mathcal{SP}_o$ Algorithm

In this section, we analyze our algorithm's performance to upper bound the expected distance traveled by the robot. We only show the performance of robot-1 in our analysis. Due to the symmetric strategies, the performance of the robot-2 is the same.

To establish an upper bound on the rendezvous time, we divide the execution into two stages. Suppose that the initial distance is $d = r^{k+\delta}$, where r is the expansion radius and $0 \le \delta < 1$. (We optimize the choice of expansion radius r in Theorem 7 below.) Stage-1 consists of the early rounds $0 \le i \le k$ in which the moving robot(s) does not travel far enough to meet the other robot. Stage-2 consists of the later rounds $i \ge (k+1)$. During Stage-2, rendezvous will succeed the very first round in which one robot is active and the other is waiting. The consistency of the Stage-2 rendezvous behavior allows us to compute the expected distance traveled using an infinite sum. Finally, we note that the robots are agnostic of the distinction between Stage-1 and Stage-2, since they do not know the initial distance d.

We now introduce the variables used in our analysis. Let R_i be the event that the algorithm is still active in round i. Assuming that R_i holds, we define the following: A_i is the event that robot-1 is moving (executing CA) in round i; B_i is the event that robot-2 is moving (executing CA) in round i. S_i is the event that the robots successfully meet in round i. For $i \le k$, we have $\mathbb{P}[S_i] = 0$ because the robots do not travel far enough to meet. For $i > k$, we define $S_i = (A_i \wedge \overline{B}_i) \vee (\overline{A}_i \wedge B_i)$. Note that this definition purposely omits the event that the robots meet while both are on the move, but the probability of this event is zero, unless the initial robot configuration obeys an exceptional geometry. The algorithm is always active in the first round, thus $\mathbb{P}[R_0] = 1$. The probability that the algorithm is still active in round i is the joint probability of the events that the robots do not meet in the rounds up to round i. That is, $\mathbb{P}[R_i] = \mathbb{P}[\overline{S}_0 \wedge \cdots \wedge \overline{S}_{i-1}]$ for $i > 0$. Furthermore, we have

$$\mathbb{P}[R_i] = \begin{cases} 1 & 0 \le i \le k, \\ 2^{-i+k} & k < i. \end{cases} \tag{2}$$

Equation (2) shows that the algorithm achieves rendezvous almost surely. Indeed, the probability that the robots have not met after $i > k$ rounds is at most 2^{-i+k}, which is exponentially small. Furthermore, the expected number of rounds before rendezvous is upper bounded by $k + 2$.

We are now ready to study Algorithm 1. Next two sections present the analysis of stages 1 and 2 respectively. We prove the following bound on the algorithm's distance competitive ratio, and its time competitive ratio.

Theorem 7. *The optimal expansion radius r of Algorithm 1 is $r = 1.225$. This choice of r gives an algorithm that has a distance competitive ratio of $62.832\frac{d}{D}$. The time competitive ratio is twice this bound.*

5.1 Analysis of Stage-1

We compute the expected distance traveled during round $i \le k$. The robots can not meet in this stage since their displacement is at most $r^k < r^{k+\delta}$. Suppose that event A_i holds: the robot is moving in round i. Let D_i denote the distance traveled in round i. Using Theorem 6, the expected distance traveled in this stage is

$$\begin{aligned} \mathbb{E}[D_i \mid A_i] &\le 4N_i D \\ &\le \frac{4\pi r^{2i}}{D} \end{aligned} \tag{3}$$

We now compute the expected distance traveled in this stage.

Lemma 8. *The expected distance traveled during Stage-1 is bounded by*

$$\sum_{i=1}^{k} \mathbb{E}\left[D_i\right] < d^2 \frac{2\pi r^2}{D(r^2-1)}.$$

Proof. Since the robots to do not travel far enough to meet, every round up to k is performed. In other words, $\mathbb{P}[R_i]$=1 for $0 \le i \le k$. Within a given round, the robot is active half of the time. Using equation (3), we find that the expected distance traveled satisfies

$$\begin{aligned}\sum_{i=0}^{k} \mathbb{E}\left[D_i\right] &= \sum_{i=0}^{k} \mathbb{E}[D_i|R_i]\mathbb{P}[R_i] = \frac{1}{2}\sum_{i=0}^{k} \mathbb{E}[D_i|A_i] \\ &= \frac{2\pi}{D}\sum_{i=0}^{k} r^{2i} \\ &< \frac{2\pi r^{2(k+1)}}{D(r^2-1)} \\ &< d^2 \frac{2\pi r^{2-2\delta}}{D(r^2-1)},\end{aligned}$$

which is maximized for $\delta = 0$. □

5.2 Analysis of Stage-2

In this section, we consider rounds $i \ge (k+1)$. We separately calculate the expected distance traveled during unsuccessful rounds and the expected distance traveled in the (final) successful round.

Lemma 9. *The expected total distance traveled during unsuccessful Stage-2 rounds is bounded by*

$$\sum_{i=k+1}^{\infty} \mathbb{E}\left[D_i \mid \overline{S}_i\right] < d^2 \frac{\pi r^2}{D\left(2-r^2\right)}.$$

Proof. We bound the expected distance traveled by robot-1; the calculation for robot-2 is analogous. Assuming that the algorithm is active in round i, we know that the robots rendezvous precisely when $S_i = (A_i \wedge \overline{B}_i) \vee (\overline{A}_i \wedge B_i)$ occurs. Furthermore, the distance traveled by robot-1 is nonzero only when A_i holds. This means that $\mathbb{E}[D_i \mid \overline{S}_i \wedge R_i] = \mathbb{E}[D_i \mid A_i \wedge B_i \wedge R_i]$, and $\mathbb{E}[D_i \mid S_i \wedge R_i] = \mathbb{E}[D_i \mid A_i \wedge \overline{B}_i \wedge R_i]$.

Using equations (2) and (3), we find that the total expected distance traveled in unsuccessful Stage-2 rounds is

$$
\begin{aligned}
&\sum_{i=k+1}^{\infty} \mathbb{E}[D_i \mid A_i \wedge B_i \wedge R_i]\mathbb{P}[A_i \wedge B_i \wedge R_i] \\
&\quad < \sum_{i=k+1}^{\infty} \left(\frac{4\pi r^{2i}}{D}\right)\left(\frac{1}{2}\right)^{i-k+2} \\
&\quad = \frac{\pi}{D}\sum_{j=0}^{\infty} r^{2(j+k+1)}\left(\frac{1}{2}\right)^{j+1} \\
&\quad = \frac{\pi r^{2k+2}}{2D}\sum_{j=0}^{\infty}\left(\frac{r^2}{2}\right)^{j} \\
&\quad = d^2\frac{\pi r^{2-2\delta}}{D\left(2-r^2\right)},
\end{aligned}
$$

which is maximized for $\delta = 0$. □

Lemma 10. *The expected distance traveled during the successful round is bounded by*

$$\sum_{i=k+1}^{\infty} E[D_i \mid S_i] \leq d^2\frac{\pi}{D}.$$

Proof. We have $S_i = (A_i \wedge \overline{B}_i) \vee (\overline{A}_i \wedge B_i)$, and only the event $A_i \wedge \overline{B}_i$ contributes to the distance traveled by robot-1. In a successful round, robot-1 covers a circular area with radius $r^{k+\delta}$ before discovering robot-2. The total distance traveled by robot-1 is calculated similarly to equation (3), using $r^i = r^{k+\delta} = d$. Independent of i, we have

$$E[D_i \mid A_i \wedge \overline{B}_i \wedge R_i] \leq \frac{4\pi d^2}{D}.$$

Therefore, the expected distance traveled in successful rounds during Stage-2 is

$$
\begin{aligned}
&\sum_{i=k+1}^{\infty} \mathbb{E}[D_i \mid A_i \wedge \overline{B}_i \wedge R_i]\mathbb{P}[A_i \wedge \overline{B}_i \wedge R_i] \\
&\quad = \frac{4\pi d^2}{D}\sum_{i=k+1}^{\infty}\left(\frac{1}{2}\right)^{i-k+2} \\
&\quad = d^2\frac{\pi}{D}.
\end{aligned}
$$

□

5.3 Computing the Competitive Ratio

Having found bounds for the expected distance traveled in both stages, we are ready to calculate the distance competitive ratio of our algorithm.

Proof of Theorem 7. The expected distance traveled is the sum of the bounds in Lemmas 8, 9 and 10. In the best offline case, the robots are omniscient and their initial locations are within unobstructed line of sight of each other. Then, they would travel $d/2$ distance toward each other to meet. Therefore, to obtain competitive ratio, we divide the expected distance traveled by $d/2$, which yields

$$d\frac{4\pi r^2}{D(r^2-1)} + d\frac{2\pi r^2}{D\,(2-r^2)} + d\frac{2\pi}{D}$$

The $\Theta(d)$ dominates this expression for large d, so we choose our r value to minimize the coefficient of d. The best value is $r = \sqrt{1.5} \approx 1.225$, giving the distance competitive ratio of $62.832\frac{d}{D}$.

The analysis for the time competitive ratio is analogous to the distance calculations. In each round, a robot is inactive with probability 1/2. In this case, the robot wait time in round i is identical to the time for an active robot to complete the round (since the robots move unit distance in unit time). Therefore, doubling the bounds in each of the three lemmas gives a bound on the expected time until completion. Of course, optimal $r = \sqrt{1.5} \approx 1.225$ is same for the time competitive ratio.

□

6 Multi-Robot Symmetric Rendezvous in Planar Environments

In this section, we extend the strategy $\mathcal{SP}_o$ from two robots to $n > 2$ robots. In the modified strategy $\mathcal{MSP}_o$, each robot decides its action in round i by flipping a weighted coin. With probability $1/n$, the robot moves, otherwise it remains stationary. We analyze the performance of $\mathcal{MSP}_o$ by finding an upper bound on the expected distance traveled by the robot.

Two robots meet when their $D \times D$ squares touch one another. For our analysis, we consider only the rendezvous cases where one robot is moving and the other one is waiting. We also assume that when two robots meet, they have access to coordination mechanisms which allow them to "stick together", that is move as a single robot. This requires communication and low level coordination about the robot's motion.

We first develop a lower bound for the best offline solution for the multi-robot case. Let C be the circle with the smallest possible radius that encloses all the robots in the environment A. This classic optimization problem, known as the smallest-circle problem and the 1-center problem, can be solved in linear time [76]. The best meeting point is the center of C. Recall that we define the distance d to be the maximal pairwise distance between the robots. The distance that one robot must travel to reach the center of C is at most $d/2$. Therefore, we will calculate our algorithm's distance competitive performance with respect to $d/2$.

As in the two-robot case, our analysis divides the execution of our multi-robot strategy into two stages. Suppose that the maximum initial pairwise distance is $d = r^{k+\delta}$, where r is the expansion radius and $0 \le \delta < 1$. Stage-1 consists of the early rounds $0 \le i \le k$ in which the moving robot(s) does not travel far enough to meet all the other robots. In calculating our upper bound, we choose to ignore any pairwise rendezvous in Stage 1. Stage-2 consists of the later rounds $i \geq (k+1)$. During Stage-2, rendezvous will succeed the very first round in which exactly one robot is active.

We now introduce the variables used in our analysis. We restrict our analysis to a single robot; the same analysis holds for the others by symmetry. For $i \geq k+1$, we define our (proxy) success event S_i as we did in Section 5. Namely, S_i corresponds to the event that exactly one robot is moving in round $i \geq k+1$. Let A_i be the event that the robot is moving in round i, so that $\mathbb{P}[A_i] = 1/n$. Assuming that the round is active, the probability of event S_i is

$$\mathbb{P}\left[S_i\right] = n\left(\frac{1}{n}\right) \cdot \left(\frac{n-1}{n}\right)^{n-1} = \left(1 - \frac{1}{n}\right)^{n-1} \approx \frac{1}{e} \tag{4}$$

for large n, where $e \approx 2.71828$ is the base of the natural logarithm. The probability of event $\overline{S}_i$ is

$$\mathbb{P}\left[\overline{S}_i\right] = 1 - \mathbb{P}\left[S_i\right] \approx 1 - 1/e. \tag{5}$$

Recall that R_i is the event that the algorithm is still active in round i. That is, $\mathbb{P}[R_i] = \mathbb{P}[\overline{S}_0 \wedge \cdots \wedge \overline{S}_{i-1}]$ for $i > 0$. Since the robots only meet when event S_i occurs, $\mathbb{P}[R_i]$ is,

$$\mathbb{P}[R_i] = \begin{cases} 1 & 0 \leq i \leq k, \\ \mathbb{P}\left[\overline{S}\right]^{i-k} & k < i. \end{cases} \tag{6}$$

As in the two-robot case, the probability that rendezvous has not occurred after round $i > k$ decays exponentially. The expected number of rounds prior to rendezvous is upper bounded by $k + e \leq k + 3$.

6.1 Analysis of Stage-1

In this section, we compute the expected distance traveled by the robot for round $i \leq k$. A moving robot can not meet all the other robots in this stage, since its displacement is at most $r^k < r^{k+\delta}$.

Lemma 11. *The expected distance traveled by the robot during Stage-1 is bounded by*

$$\sum_{i=1}^{k} \mathbb{E}\left[D_i\right] < d^2 \frac{4\pi r^2}{nD\left(r^2-1\right)}.$$

Proof. We have $\mathbb{P}[R_i]$=1 for $0 \leq i \leq k$ because in these early rounds, a moving robot does not travel far enough to encounter every other robot. If a robot decides to move in round i, then it executes Algorithm CA. Therefore, using Theorem 6, the expected distance traveled by a robot in this stage satisfies

$$\begin{aligned}\sum_{i=0}^{k}\mathbb{E}\left[D_i\right] &= \sum_{i=0}^{k}\mathbb{E}[D_i|A_i]\mathbb{P}[A_i] = \frac{1}{n}\sum_{i=0}^{k}\mathbb{E}[D_i|A_i] \\ &= \frac{1}{n}\sum_{i=0}^{k}\frac{4\pi r^{2i}}{D} = \frac{4\pi}{nD}\sum_{i=0}^{k} r^{2i} \\ &< \frac{4\pi r^{2k+2}}{nD\left(r^2-1\right)} \\ &< d^2\frac{4\pi r^{2-2\delta}}{nD\left(r^2-1\right)}, \end{aligned} \tag{7}$$

which is maximized for $\delta = 0$. □

6.2 Analysis of Stage-2

In this section, we consider rounds $i \geq (k+1)$, in which the robots meet with nonzero probability.

Lemma 12. *The expected total distance traveled in Stage-2 is bounded by*

$$\sum_{i=k+1}^{\infty}\mathbb{E}\left[D_i \mid \overline{S}_i\right] < d^2\frac{4\pi r^2\mathbb{P}\left[\overline{S}_i\right]}{nD(1-r^2\mathbb{P}\left[\overline{S}_i\right])}.$$

Proof. Recall that S_i is the event that exactly one robot is active in the current round. For $i > k$, we have $\mathbb{P}[R_i] = \mathbb{P}[\overline{S}_i]^{i-k}$, where $\mathbb{P}[S_i]$ is given in equation (4). Furthermore, the distance traveled by the robot in round i is nonzero only when A_i holds. This means that the total expected distance traveled by the robot in Stage-2 satisfies

$$\begin{aligned}\sum_{i=k+1}^{\infty}\mathbb{E}[D_i] &= \sum_{i=k+1}^{\infty}\mathbb{E}[D_i \mid A_i \wedge R_i]\mathbb{P}[A_i|R_i]\mathbb{P}[R_i]. \\ &\leq \sum_{i=k+1}^{\infty}\frac{4\pi r^{2i}}{D}\cdot\frac{1}{n}\cdot\mathbb{P}\left[\overline{S}_i\right]^{i-k} = \frac{4\pi}{nD}\sum_{i=k+1}^{\infty} r^{2i}\mathbb{P}\left[\overline{S}_i\right]^{i-k} \\ &= \frac{4\pi}{nD}\sum_{i=0}^{\infty} r^{2(i+k+1)}\mathbb{P}\left[\overline{S}_i\right]^{i+1} = \frac{4\pi r^{2(k+1)}\mathbb{P}[\overline{S}_i]}{nD}\sum_{i=0}^{\infty}(r^2\mathbb{P}\left[\overline{S}_i\right])^i \\ &= \frac{4\pi r^{2(k+1)}\mathbb{P}\left[\overline{S}_i\right]}{nD(1-r^2\mathbb{P}\left[\overline{S}_i\right])} = d^2\frac{4\pi r^{2-2\delta}\mathbb{P}\left[\overline{S}_i\right]}{nD(1-r^2\mathbb{P}\left[\overline{S}_i\right])},\end{aligned}$$

which is maximized for $\delta = 0$. □

6.3 Computing the Distance Competitive Ratio

Having found bounds for the expected distance traveled in both stages, we are ready to calculate the distance competitive ratio of our algorithm.

Theorem 13. *The best choice for the expansion radius is $r = 1.12$ which gives a distance competitive ratio of $220.151d/(nD)$, where d is the maximal distance between the robots, n is the number of robots, and D is the side length of a square robot.*

Proof of Theorem 13. The expected distance traveled is the sum of the bounds in Lemmas 11 and 12. To obtain competitive ratio, we divide this sum by $d/2$, obtaining

$$\frac{8\pi d}{nD} r^2 \left(\frac{1}{r^2 - 1} + \frac{\mathbb{P}\left[\overline{S}_i\right]}{1 - r^2 \mathbb{P}\left[\overline{S}_i\right]} \right)$$

where $\mathbb{P}[\overline{S}_i] \approx 1 - 1/e$ by equation (5). The value $r = 1.12$ minimizes this expression, giving a distance competitive ratio of $O\left(d/(nD)\right)$. □

The analysis for the time competitive ratio is analogous to the distance calculations. The time competitive ratio is n times the distance competitive ratio, and $r = 1.12$ is the optimal expansion radius for both.

7 Simulation Experiments

In this section, we explore the results of Theorems 6, 7 and 13 in simulations. We verify the bounds obtained in these theorems, test the performance and show the execution of Algorithms $\mathcal{SP}_o$, $\mathcal{MSP}_o$ and CA in real world environments.

Simulation experiments are conducted in the ROS platform [77]. Robots are equipped with Hokuyo laser sensors. We executed Algorithms $\mathcal{SP}_o$ and $\mathcal{MSP}_o$ in three different indoor environments populated with obstacles. All of the simulation experiments were run with the following parameters: the size of each grid cell is $D = 0.6m$; the size of the square robot is $D \times D = 0.6m \times 0.6m$; robot speed is $0.25m/sec$; and the expansion radius value is $r = 1.12$. The maximum grid size G_{max} for full coverage of Environment-1, Environment-2 and Environment-3 are respectively $19m \times 19m$, $18m \times 18m$ and $17m \times 17m$. The paths (corridors) that would lead the robot to the small rooms in the environments are blocked. These blocked areas appear dotted and gray colored in the figures.

Figure 7(b) shows the sample environments and the paths followed by the robot executing CA. The initial location I_j of the robot in Environment-j is shown by a (red colored) square, for $j \in [1, 3]$. Note that the robot cannot enter the corridors or rooms with very narrow openings due to its size and the minimum distance it keeps from an obstacle, which is set to $0.6m$.

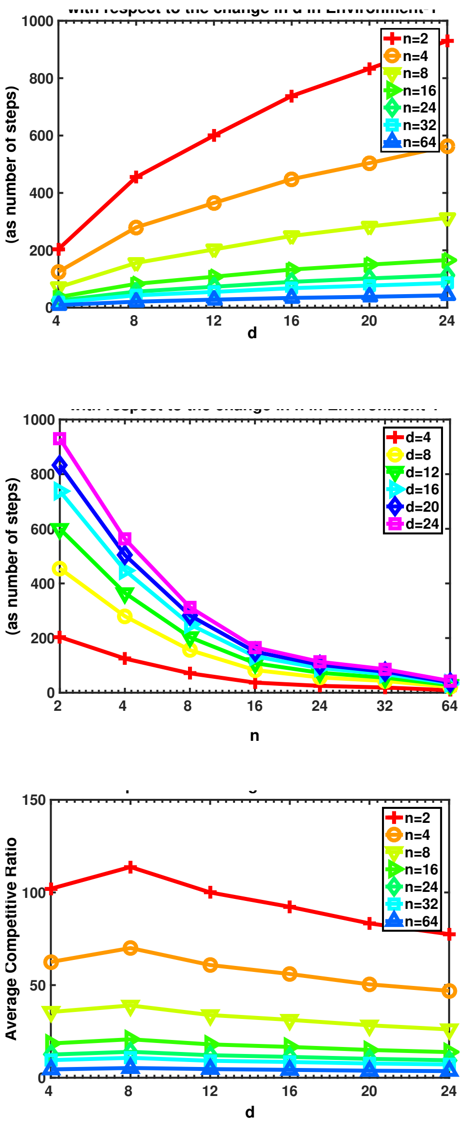
Let i_j denote the first round that the robot's disk encircles Environment-j. In another set of experiments, we assume that the robot flipped a head, and hence executed CA, in all rounds $0 < i \leq i_j$. The left plot in Figure 8 shows the coverage time in terms of the simulation time, and and the right plot shows the number of steps of CA with respect to changes in grid size. Grid size here represents the area that lays in $Disk_i$ with radius r_i, which increases as round i increases. The results show that Environment-1 has the longest coverage time for all grid sizes. As expected, both the number of steps and the simulation time increase as the grid size increases. The results verify the upper bound of CA Algorithm we obtained in Theorem 6. In the trial executions of CA in each environment, the robot accomplished full coverage. Note that the coverage time is constant for the environment that has G_{max} exceeding the grid size that appears in the x-axis.

In Figures 7(a) and 9, we report the results of trial executions of $\mathcal{SP}_o$ and $\mathcal{MSP}_o$ in Environment-1. We varied n between 4 and 64 and d between 4 and 24. For each n and d combination, we performed 10000 trials. Arbitrary initial positions are assigned to the robots for each environment such that the maximum initial distance between two robots is d. The initial locations of the robots are the same for each trial. The top plot in Figure 7(a) shows the average distance traveled (as number of steps) for rendezvous with respect to the change in d. The middle plot in Figure 7(a) shows the average distance traveled (as number of steps) for rendezvous with respect to the change in n. Not surprisingly, the average number of steps increases as d increases, while the average number of steps per robot shows less variation as n increases. The bottom plot shows that the average competitive ratio remains constant as both n and d change when $r = 1.12$. Finally, Figure 9 shows the average number of rounds and the standard deviation in the average number of steps, each plotted against maximum initial distance d.

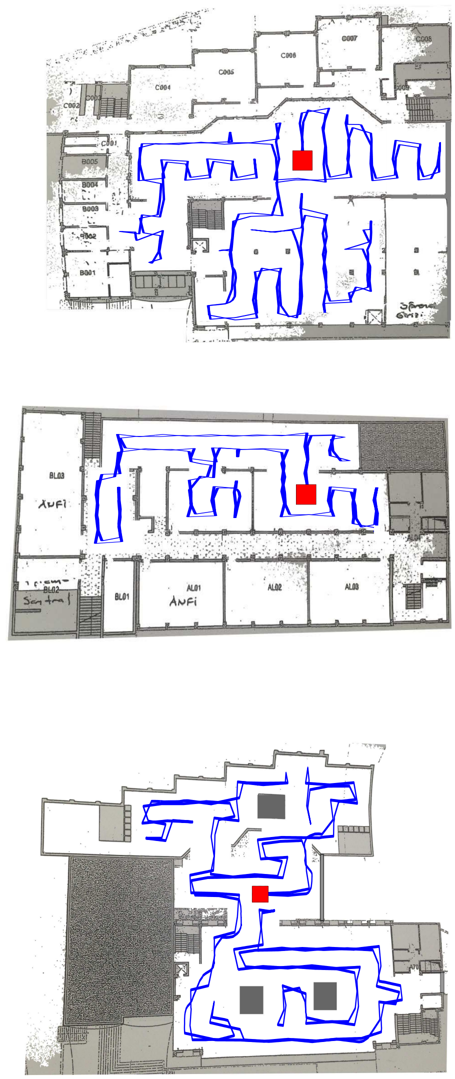
8 Future Work

In this paper, we studied the symmetric rendezvous search problem in planar environments with obstacles. We presented a randomized rendezvous strategy for two robots and extended our strategy to multiple robots. We proved that the competitive ratio of our strategy is $O(d/D)$ for two robots, where d is the initial distance between the robots and D is the side length of the square robots. For n robots, we showed that its competitive ratio is $O(d/(nD))$, where d is the maximal pairwise distance between the robots. In addition to the rendezvous algorithm, we proposed a coverage algorithm that provides complete coverage of an unknown environment. Finally, we presented the results of various simulations in different environments.

An interesting future research direction is to study rendezvous search in three dimensional environments such as buildings with multiple floors or underwater scenarios. Other avenues of research include instances with communication constraints (e.g., the robots can communicate once they are within a given range) and problems involving heterogeneous teams (e.g., ground and aerial vehicles).



(a)



(b)

Figure 7: (a) TOP: Average number of steps until rendezvous for various d values with respect to changes in n. MIDDLE: Average number of steps until rendezvous for various n values with respect to changes in d. BOTTOM: Average distance competitive ratio until rendezvous for various d values with respect to changes in n. (b) Sample environments and the paths followed by a robot to cover the environments. Red square shows the initial location of the robot in the environment. TOP: Environment-1. MIDDLE: Environment-2. BOTTOM: Environment-3.

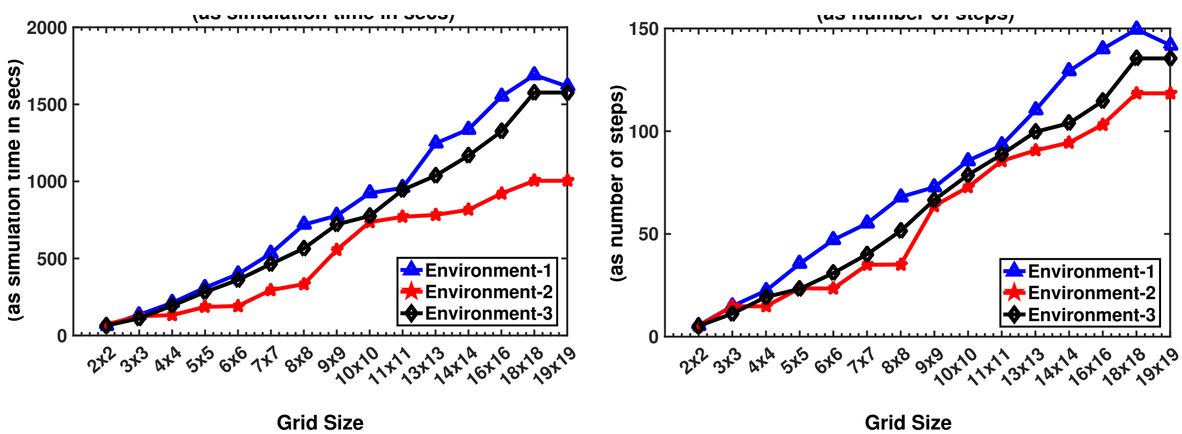


Figure 8: LEFT: Simulation time of coverage in seconds in Environments 1-3. RIGHT: Number of steps for coverage in Environments 1-3.

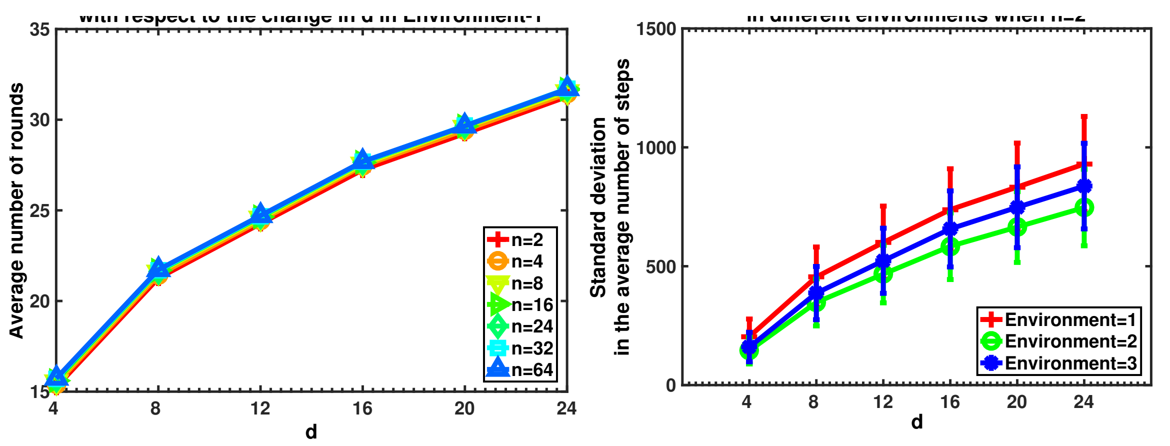


Figure 9: LEFT: Average number of rounds with respect to the change in d. RIGHT: Standard deviation in average number of steps with respect to the change in d.

References

[1] N. Agmon, N. Hazon, G. A. Kaminka, M. Group, et al. The giving tree: constructing trees for efficient offline and online multi-robot coverage. *Annals of Mathematics and Artificial Intelligence*, 52(2-4):143–168, 2008.

[2] N. Agmon and D. Peleg. Fault-tolerant gathering algorithms for autonomous mobile robots. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, page 1078. Society for Industrial and Applied Mathematics, 2004.

[3] S. Albers, K. Kursawe, and S. Schuierer. Exploring unknown environments with obstacles. *Algorithmica*, 32(1):123–143, 2002.

[4] S. Alpern. The rendezvous search problem. *SIAM Journal on Control and Optimization*, 33(3):673–683, 1995.

[5] S. Alpern and V. Baston. Rendezvous in higher dimensions. *SIAM journal on control and optimization*, 44(6):2233–2252, 2006.

[6] S. Alpern and A. Beck. Asymmetric rendezvous on the line is a double linear search problem. *Mathematics of Operations Research*, 24(3):604–618, 1999.

[7] S. Alpern and A. Beck. Pure strategy asymmetric rendezvous on the line with an unknown initial distance. *Operations Research*, 48(3):498, 2000.

[8] S. Alpern and S. Gal. Rendezvous search on the line with distinguishable players. *SIAM Journal on Control and Optimization*, 33(4):1270–1276, 1995.

[9] S. Alpern and H. Street. Rendezvous in one and more dimensions. Technical report, CDAM Research Report LSE-CDAM-2001-05, 2001.

[10] E. Anderson and S. Essegaier. Rendezvous search on the line with indistinguishable players. *SIAM Journal on Control and Optimization*, 33:1637, 1995.

[11] E. Anderson and S. Fekete. Two dimensional rendezvous search. *Operations Research*, pages 107–118, 2001.

[12] E. Anderson and R. Weber. The rendezvous problem on discrete locations. *Journal of Applied Probability*, 27(4):839–851, 1990.

[13] H. Ando, I. Suzuki, and M. Yamashita. Formation and agreement problems for synchronous mobile robots with limited visibility. In *Intelligent Control, 1995., Proceedings of the 1995 IEEE International Symposium on*, pages 453–460. IEEE, 1995.

[14] R. A. Baezayates, J. C. Culberson, and G. J. Rawlins. Searching in the plane. *Information and computation*, 106(2):234–252, 1993.

[15] E. Bampas, J. Czyzowicz, L. Gasieniec, D. Ilcinkas, and A. Labourel. Almost optimal asynchronous rendezvous in infinite multidimensional grids. In *International Symposium on Distributed Computing*, pages 297–311. Springer, 2010.

[16] V. Baston. Note: Two rendezvous search problems on the line. *Naval Research Logistics*, 46(3):335–340, 1999.

[17] V. Baston and S. Gal. Rendezvous on the line when the players' initial distance is given by an unknown probability distribution. *SIAM Journal on Control and Optimization*, 36:1880, 1998.

[18] A. Beck and D. Newman. Yet more on the linear search problem. *Israel journal of mathematics*, 8(4):419–429, 1970.

[19] F. Belkhouche, B. Belkhouche, and P. Rastgoufard. Multi-robot rendezvous in the plane. In *2005 IEEE International Conference on Systems, Man and Cybernetics*, pages 2293–2298, 2005.

[20] S. Choi, S. Lee, H. H. Viet, and T. Chung. B-theta*: an efficient online coverage algorithm for autonomous cleaning robots. *Journal of Intelligent & Robotic Systems*, pages 1–26, 2017.

[21] M. Cieliebak. Gathering non-oblivious mobile robots. *LATIN 2004: Theoretical Informatics*, pages 577–588, 2004.

[22] M. Cieliebak, P. Flocchini, G. Prencipe, and N. Santoro. Solving the robots gathering problem. *Automata, Languages and Programming*, pages 192–192, 2003.

[23] M. Cieliebak and G. Prencipe. Gathering autonomous mobile robots. In *Proc. 9th Int. Colloq. on Structural Information and Communication Complexity*, pages 57–72. Citeseer, 2002.

[24] A. Collins, J. Czyzowicz, L. Gasieniec, and A. Labourel. Tell me where i am so i can meet you sooner. In *International Colloquium on Automata, Languages, and Programming*, pages 502–514. Springer, 2010.

[25] J. Czyzowicz, D. Ilcinkas, A. Labourel, and A. Pelc. Worst-case optimal exploration of terrains with obstacles. *Information and Computation*, 225:16–28, 2013.

[26] J. Czyzowicz, A. Pelc, and A. Labourel. How to meet asynchronously (almost) everywhere. *ACM Transactions on Algorithms (TALG)*, 8(4):37, 2012.

[27] P. Dasgupta, P. Chakrabarti, and S. DeSarkar. Agent searching in a tree and the optimality of iterative deepening. *Artificial Intelligence*, 71(1):195–208, 1994.

[28] X. Deng, T. Kameda, and C. Papadimitriou. How to learn an unknown environment. i: the rectilinear case. *Journal of the ACM (JACM)*, 45(2):215–245, 1998.

[29] A. Dessmark, P. Fraigniaud, D. Kowalski, and A. Pelc. Deterministic rendezvous in graphs. *Algorithmica*, 46(1):69–96, 2006.

[30] Y. Dieudonné, A. Pelc, and V. Villain. How to meet asynchronously at polynomial cost. In *Proceedings of the 2013 ACM symposium on Principles of distributed computing*, pages 92–99. ACM, 2013.

[31] K. Easton and J. Burdick. A coverage algorithm for multi-robot boundary inspection. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 727–734. IEEE, 2005.

[32] J. Fang, A. Morse, and M. Cao. Multi-agent rendezvousing with a finite set of candidate rendezvous points. In *American Control Conference, 2008*, pages 765–770, 2008.

[33] P. Fazli, A. Davoodi, and A. K. Mackworth. Multi-robot repeated area coverage. *Autonomous robots*, 34(4):251–276, 2013.

[34] P. Fazli, A. Davoodi, P. Pasquier, and A. K. Mackworth. Complete and robust cooperative robot area coverage with limited range. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 5577–5582. IEEE, 2010.

[35] S. P. Fekete and C. Schmidt. Polygon exploration with time-discrete vision. *Computational Geometry*, 43(2):148–168, 2010.

[36] P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Gathering of asynchronous oblivious robots with limited visibility. *STACS 2001*, pages 247–258, 2001.

[37] P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Gathering of asynchronous robots with limited visibility. *Theoretical Computer Science*, 337(1):147–168, 2005.

[38] P. Flocchini, N. Santoro, G. Viglietta, and M. Yamashita. Rendezvous of two robots with constant memory. In *International Colloquium on Structural Information and Communication Complexity*, pages 189–200. Springer, 2013.

[39] Y. Gabriely and E. Rimon. Spanning-tree based coverage of continuous areas by a mobile robot. *Annals of Mathematics and Artificial Intelligence*, 31(1-4):77–98, 2001.

[40] Y. Gabriely and E. Rimon. Competitive on-line coverage of grid environments by a mobile robot. *Computational Geometry*, 24(3):197–224, 2003.

[41] Y. Gabriely and E. Rimon. Competitive complexity of mobile robot on line motion planning problems. pages 155–170, 2005.

[42] S. Gal. Rendezvous search on the line. *Operations Research*, pages 974–976, 1999.

[43] A. Ganguli, J. Cortes, and F. Bullo. Multirobot rendezvous with visibility sensors in nonconvex environments. *IEEE Transactions on Robotics*, 25(2):340–352, 2009.

[44] S. K. Ghosh, J. W. Burdick, A. Bhattacharya, and S. Sarkar. Online algorithms with discrete visibility-exploring unknown polygonal environments. *Robotics & Automation Magazine, IEEE*, 15(2):67–76, 2008.

[45] M. Hammar, B. J. Nilsson, and S. Schuierer. Improved exploration of rectilinear polygons. *Nordic Journal of Computing*, 9(1):32–53, 2002.

[46] Q. Han, D. Du, J. Vera, and L. Zuluaga. Improved bounds for the symmetric rendezvous value on the line. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, page 78. Society for Industrial and Applied Mathematics, 2007.

[47] C. Jurek, G. Leszek, and A. Pelc. Gathering few fat mobile robots in the plane. *Theoretical Computer Science*, 410(6-7):481–499, 2009.

[48] B. Kalyanasundaram and K. Pruhs. A competitive analysis of algorithms for searching unknown scenes. *Computational Geometry*, 3(3):139–155, 1993.

[49] M.-Y. Kao, J. H. Reif, and S. R. Tate. Searching in an unknown environment: An optimal randomized algorithm for the cow-path problem. *Information and Computation*, 131(1):63–79, 1996.

[50] A. Khan, I. Noreen, H. Ryu, N. L. Doh, and Z. Habib. Online complete coverage path planning using two-way proximity search. *Intelligent Service Robotics*, 10(3):229–240, 2017.

[51] R. Klasing, A. Kosowski, and A. Navarra. Taking advantage of symmetries: gathering of asynchronous oblivious robots on a ring. *Principles of Distributed Systems*, pages 446–462, 2008.

[52] J. M. Kleinberg. On-line search in a simple polygon. In *Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 8–15. Society for Industrial and Applied Mathematics, 1994.

[53] D. Kowalski and A. Pelc. Polynomial deterministic rendezvous in arbitrary graphs. *Algorithms and Computation*, pages 644–656, 2005.

[54] D. R. Kowalski and A. Malinowski. How to meet in anonymous network. *Theoretical Computer Science*, 399(1):141–156, 2008.

[55] J. Lin, A. Morse, and B. Anderson. The multi-agent rendezvous problem. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, volume 2, pages 1508–1513, 2003.

[56] J. Lin, A. Morse, and B. Anderson. The multi-agent rendezvous problem. Part 2: The asynchronous case. *SIAM Journal on Control and Optimization*, 46(6):2120–2147, 2007.

[57] G. Marco, L. Gargano, E. Kranakis, D. Krizanc, A. Pelc, and U. Vaccaro. Asynchronous deterministic rendezvous in graphs. *Theoretical Computer Science*, 355(3):315–326, 2006.

[58] N. Megiddo. Linear-time algorithms for linear programming in r^3 and related problems. *SIAM J. Comput.*, 12(4):759–776, 1983.

[59] D. Ozsoyeller, A. Beveridge, and V. Isler. Symmetric rendezvous search on the line with an unknown initial distance. *IEEE Transactions on Robotics*, 29(6):1366–1379, 2013.

[60] D. Ozsoyeller, V. Isler, and A. Beveridge. Symmetric rendezvous in planar environments with and without obstacles. In *AAAI*, 2012.

[61] A. Pelc. Deterministic gathering with crash faults. *Networks*, 2018.

[62] S. Poduri and G. Sukhatme. Achieving connectivity through coalescence in mobile robot networks. In *Proceedings of the 1st international conference on Robot communication and coordination*, page 4. IEEE Press, 2007.

[63] G. Prencipe. On the feasibility of gathering by autonomous mobile robots. *Structural Information and Communication Complexity*, pages 246–261, 2005.

[64] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, 2009.

[65] N. Roy and G. Dudek. Collaborative robot exploration and rendezvous: Algorithms, performance bounds and observations. *Autonomous Robots*, 11(2):117–136, 2001.

[66] I. Shnaps and E. Rimon. Online coverage by a tethered autonomous mobile robot in planar unknown environments. *IEEE Transactions on Robotics*, 30(4):966–974, 2014.

[67] S. Souissi, X. Défago, and M. Yamashita. Gathering asynchronous mobile robots with inaccurate compasses. *Principles of Distributed Systems*, pages 333–349, 2006.

[68] G. Stachowiak. Asynchronous Deterministic Rendezvous on the Line. *SOFSEM 2009: Theory and Practice of Computer Science*, pages 497–508, 2009.

[69] G. P. Strimel and M. M. Veloso. Coverage planning with finite resources. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 2950–2956. IEEE, 2014.

[70] I. Suzuki and M. Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. *SIAM Journal on Computing*, 28(4):1347–1363, 1999.

[71] L. Thomas and P. Hulme. Searching for targets who want to be found. *Journal of the Operational Research Society*, 48(1):44–50, 1997.

[72] P. P. Uthaisombut. Symmetric rendezvous search on the line using move patterns with different lengths. 2006.

[73] R. Weber. The optimal strategy for symmetric rendezvous search on k_3. *arXiv preprint arXiv:0906.5447*, 2009.

[74] A. Xu, C. Viriyasuthee, and I. Rekleitis. Efficient complete coverage of a known arbitrary environment with applications to aerial operations. *Autonomous Robots*, 36(4):365–381, 2014.

[75] Z. Yao. Finding efficient robot path for the complete coverage of a known space. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 3369–3374. IEEE, 2006.

[76] J. Yu, S. M. LaValle, and D. Liberzon. Rendezvous without coordinates. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 1803–1808. IEEE, 2008.

[77] X. Zheng, S. Jain, S. Koenig, and D. Kempe. Multi-robot forest coverage. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3852–3857. IEEE, 2005.