Probabilistic Network Formation through Coverage and Freeze-Tag

Eric Meisner[1], Wei Yang[1], and Volkan Isler[2*]

[1] Department of Computer Science, Rensselaer Polytechnic Institute, {meisne,yangw}@cs.rpi.edu
[2] Department of Computer Science, University of Minnesota, isler@cs.umn.edu

Abstract: We address the problem of propagating a piece of information among robots scattered in an environment. Initially, a single robot has the information. This robot searches for other robots to pass it along. When a robot is discovered, it can participate in the process by searching for other robots. Since our motivation for studying this problem is to form an ad-hoc network, we call it the *Network Formation Problem.* In this paper, we study the case where the environment is a rectangle and the robots' locations are unknown but chosen uniformly at random. We present an efficient network formation algorithm, Stripes, and show that its expected performance is within a logarithmic factor of the optimal performance. We also compare Stripes with an intuitive network formation algorithm in simulations. The feasibility of Stripes is demonstrated with a proof-of-concept implementation.

1 Introduction

Consider the following scenario: a number of robots are performing independent tasks autonomously in an environment where there is no communication infrastructure. Suppose, at a certain point, mission priorities change and a piece of information must be propagated to all nodes. For example, robots could be initially stationed to perform monitoring or surveillance in a large environment. Upon detection of an event, they may have to form a network or perform a collaborative task. What is a good strategy to get all robots involved as quickly as possible?

In this paper, we study this process of propagating information as quickly as possible. Specifically, we study the case where the process is initiated by a single robot. This robot could, for example, be sent out from the command and control center. Alternatively, it could be the robot that detects an intruder. The primary difficulty in solving the problem arises from the fact that the

[*] This work is supported in part by NSF IIS-0745537, NSF CCF-0634823 and NSF CNS-0707939.

robots do not know each others' positions. The first robot must therefore start a search. Once discovered, other robots can participate in propagating the information. Since our primary motivation for studying this problem is to form a connected network, throughout the paper we will refer to it as the *Network Formation Problem.*

Our Contributions: In this paper, we study a probabilistic scenario where the locations of robots are chosen uniformly at random in a rectangular environment. For this scenario, we present a network formation strategy (Stripes) and prove that its expected performance (i.e. network formation time) is within a logarithmic factor of the optimal performance. To obtain this result, we also obtain a lower bound on the expected performance of *any* network formation strategy. We also compare Stripes with a natural, intuitive "Split and Cover" strategy and show that in large environments, Split and Cover has inferior performance to Stripes. In addition to formal performance bounds, we demonstrate the utility of Stripes with simulations and its feasibility with a proof-of-concept implementation. In the next section, we start with an overview of related work where we establish connections between the network formation problem and other fundamental problems such as rendezvous, coverage and freeze-tag.

1.1 Related Work

Considerable work has been done in designing decentralized protocols for propagating information in networks (also known as gossip protocols) [7]. Gossip protocols are mainly designed for stationary networks. In contrast, we focus on information propagation among mobile robots. The network formation problem is closely related to the Freeze-Tag problem [4]. In freeze-tag, a number of players are "frozen". A single unfrozen player must visit each frozen player in order to unfreeze it, at which point it can aid in unfreezing other players. In freeze tag, it is assumed that the players know each others' positions. In network formation, we focus on the case where the node locations are unknown to each other.

Recently, Poduri and Sukhatme explicitly addressed the problem of forming a connected network under the name *coalescence* [11, 12]. In their model, all of the nodes (other than the base station) are performing a random walk on a torus. The authors obtain bounds on the network formation time. The advantage of the random-walk strategy is that it does not require localization with respect to a global reference frame. However, the network formation is rather slow because nodes visit most locations many times. This may not be acceptable in time-critical applications. In the present work, we address the problem of explicitly designing motion strategies for network formation with guaranteed performance. Since we focus on the case where the robot locations are unknown, the network formation problem is related to rendezvous and coverage.

The rendezvous search problem [1] asks how two players with the same speed can locate each other when placed at arbitrary locations in a known environment. Typically, each player has a radius of detection within which the players can establish communication. The goal of the players is to find each other as quickly as possible. The rendezvous problem has been studied extensively for two players on a line. Optimal or provably good strategies for various versions of this problem have been established [2].

The problem of multi-player rendezvous search on a complete graph is studied in [14]. This work addresses the question of whether players that meet should stick together or split and meet again later. The result of this work shows that, if the players have no memory, then the optimal strategy is to stick together. Also, simulations show that as the number of players increases, the split and meet strategy becomes less effective. In general, this work has limited applications because the environment is a complete graph. In other words, players can teleport to arbitrary locations at every time step.

Work in [3] describes a time optimal strategy for two-player limited visibility rendezvous in the plane with known and unknown distances. The optimal solution in this case is for one of the players to follow a semi-circular spiral trajectory. This work also relates rendezvous in the continuous domain to coverage problems. Deterministic and randomized multi-robot rendezvous strategies were proposed in [13]. More recently, results on rendezvous in simply-connected polygons have been obtained [8]. In [9], the authors provide upper and lower bounds on the time complexity of two geometric laws that result in rendezvous.

The coverage or lawn mowing problem [5, 6] has been extensively studied and is known to be closely related to Traveling Salesperson Problem. The primary difference between network formation and standard multi-robot coverage is that in coverage, all robots participate in the coverage process from the beginning. In network formation, the process starts with one robot, who must "recruit" others to participate in coverage.

In short, the algorithm presented in this paper can be considered a novel algorithm for (i) online freeze-tag, (ii) probabilistic multi-robot coverage, and (iii) network formation.

2 Problem Formulation

Since the robot locations are unknown, network formation is an online problem. Online problems are typically analyzed using competitive analysis where the input is chosen by an adversary. The competitive ratio of an online algorithm $\mathcal{O}$ is the worst case ratio of the performance of $\mathcal{O}$ to the optimal offline performance. In other words, we compare $\mathcal{O}$ with the optimal algorithm that has access to all of $\mathcal{O}$'s input in advance and consider the worst case deviation from this optimal behavior.

It is easy to see that there is no online algorithm for the network formation problem with bounded competitive ratio: no matter which path the first robot chooses, the adversary can place all other robots at the last location the first robot will visit. In the case of a square environment with area a^2, the online algorithm would take time proportional to a^2 whereas the optimal offline cost would be at most in the order of a. Therefore the competitive ratio would be a and it would grow unboundedly with the size of the environment.

Since there are no competitive online algorithms for network formation, in this paper we focus on the probabilistic case where the locations of the robots are sampled from a distribution. In the absence of any information, it is reasonable to assume that the locations are chosen uniformly at random from the environment. *The goal is to minimize the expected time to discover all robots.* In this paper, we focus on rectangular environments and uniform distributions. We believe that the rectangular environment case has practical relevance for robots operating in an open field as well as for Unmanned Aerial Vehicles. We discuss extensions to general convex environments in Section 6.

2.1 Robot Model

In network formation, several identical mobile robots are distributed uniformly at random within the bounded rectangle A with one of these robots possessing information that needs to be propagated to all of the other robots. The rectangle A has a width w and height h where $w \geq h$. We assume that the robots are initially stationary. Once discovered, they can move anywhere within the rectangle A at a constant speed and can communicate with each other within a limited range. Once a robot is within the communication range of another robot, they can exchange any information that either robot might have, including the information that needs to be propagated. In this paper, we assume that the robots can localize themselves within A. We also ignore energy limitations and assume that the robots will always move at their maximum speed and can utilize their wireless radio anytime.

Notation and Conventions: We normalize the units so that the robots move at unit speed per time-unit. In the rest of the paper, we use A to denote both the rectangle environment and its area. This convention implies that a single robot can cover the entire environment in A time units. Hence, A is a trivial upper bound on network formation since inactive nodes are stationary. The number of robots in the environment is k. We assume that the first robot enters the environment at a corner of A.

3 Stripes: An Efficient Network Formation Strategy

In this section, we present our main result: an efficient network formation strategy which we refer to as “Stripes”. This strategy relies on dividing the rectangular environment into n equal sized vertical stripes $S_1, \ldots, S_n$ (Figure 1). For now, we treat n as a variable whose value will be fixed later. Let S denote the area of a single stripe which is equal to the time to cover a single stripe with one robot.

In the beginning, a single robot is active and proceeds to cover S_1 with a simple back and forth sweeping motion. That is, the robot follows the well-known boustrophedon[1] path. When an inactive robot is encountered and activated, this newly active robot does not join in the coverage of S_1 right away [2]. Instead, it heads to a designated location on the line that separates S_1 and S_2 and waits for the first active robot to finish its coverage of S_1 and arrive at the designated location. When the first robot is finished, it meets all newly active robots at the same designated location. Let k_1 denote the number of active robots. The robots evenly divide S_2 among themselves so that it can be covered in parallel in time S_2/k_1.

When these k_1 robots encounter other inactive robots in S_2, the same procedure is repeated and all active robots meet at a designated meeting location on the line which separates S_2 and S_3. This process repeats for the remaining stripes until all of the stripes are covered, at which point, the entire bounded area will be covered and all of the robots will be activated.

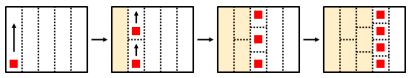


Fig. 1. Stripes strategy: The environment is divided into equal vertical stripes which are covered sequentially. Active robots split the current stripe equally. Once a stripe is covered, all active robots (including newly discovered robots) meet at the boundary of the stripe.

For most of the paper, we will focus on rectangular environments which are large with respect to the number of robots. In other words, we focus on the case where $w \geq h \gg k$.

In the remaining case, the number of robots exceeds the size of the shorter side of the rectangle (i.e. $k \geq h$). We refer to this case as the *saturated case*. When the number of active robots reach h, the remaining $m \times h$ area can be covered in m steps by the active robots. We present the analysis of the algorithm for the saturated case in the appendix.

3.1 Network Formation with Stripes

In this section, we establish an upper bound on the network formation time of Stripes for an unsaturated environment. We will use the following lemma.

[1] boustrophedon = "the way of the ox" [6].

[2] In practice, this robot can participate in covering the stripe. However, this does not improve the analysis. Hence, we ignore this benefit for analysis purposes.

Lemma 1. *If k balls are assigned to n bins randomly, the expected number of empty bins is at most $ne^{-k/n}$.*

Proof. We say that bin i is "hit" if it is non-empty after the k^{th} throw. The probability p_i that the i^{th} bin is not hit is $(1 - \frac{1}{n})^k$. Let X_i be a random variable which takes the value one if bin i is not hit and zero otherwise. Then, $E[X_i] = p_i$. The number of empty bins is given by the random variable $\sum_{i=1}^{n} X_i$ whose expected value is np_i. The lemma follows from the inequality $(1 + u) \leq e^u$.

To obtain an upper bound on the network formation time with Stripes, we treat the robots as balls and stripes as bins. Note that the coverage time is maximized when the empty stripes occur at the beginning, and non-empty stripes have the following property: let m be the number of non-empty stripes. $m - 1$ stripes have only one robot inside and all the remaining robots are in a single stripe which occurs after all other stripes. We compute the expected coverage time for this worst case.

If we pick $n = \frac{k}{\log k}$, by Lemma 1 the number of empty stripes is at most $\frac{1}{\log k}$, which is less than or equal to 1 for $k > 1$.

This results in a coverage time of $\frac{A}{n}$ for the first non-empty stripe, $\frac{A}{n}\left(\frac{1}{2}\right)$ for the second, $\frac{A}{n}\left(\frac{1}{3}\right)$ and so on for the remaining stripes. Since there is at most one non-empty stripe, the total coverage time will be bounded by:

$$\frac{A}{n} + \sum_{i=1}^{n} \frac{A}{n}\left(\frac{1}{i}\right) = \frac{A}{n} + \frac{A}{n}\sum_{i=1}^{n}\frac{1}{i} \approx \frac{A}{n} + \frac{A}{n}\log n \qquad (1)$$

By plugging in $n = \frac{k}{\log k}$, we have the following result.

Lemma 2. *The expected network formation time for k robots in a rectangular unsaturated environment with area A is $O(\frac{A}{k}\log^2 k)$ when robots execute the Stripes algorithm with $n = \frac{k}{\log k}$ stripes.*

Since robots group together before and after covering a stripe, there is some overhead associated with the different strategies that can be used to cover an individual stripe. In establishing Lemma 2, this overhead is ignored. We justify this in the next section, where we present the strategy to cover individual stripes.

3.2 Covering a Single Stripe

The Stripes algorithm calls for the set of active robots to meet and regroup between covering stripes. In this section, we present a strategy for multiple robots to cover a single stripe in a way that minimizes the overhead of meeting to redistribute assignments. Our proposed single stripe strategy ensures that each robot travels the same distance during an epoch, and arrives at a common meeting location.

In order to minimize the coverage time of a single stripe, its area must be split equally between all of the active robots. This can be done by simply dividing the strip along the long side to create equal sized rectangular areas, one for each of the active robots. However, this will result in some robots traveling longer distances to reach their assigned coverage objectives. Since this area will eventually be covered by the other robots, this produces an overhead of $2h$ per stripe, adding up to $2nh$ for the entire bounded area A. Our single stripe strategy divides the active stripe into sections such that, 1) the area covered by each robot is equal and 2) the time required to cover each section and then reach the meeting location is equal. In the end, the paths of the robots resemble archways with different sized horizontal areas that are inversely proportional to the distance a robot needs to travel to the meeting location (Figure 2). The "legs" of the archways represent the areas covered by each robot as they travel up to their assigned rectangular areas.

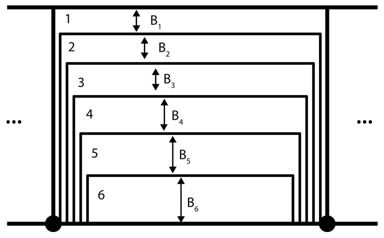


Fig. 2. Our single stripe coverage strategy divides a vertical stripe into equally sized areas that take into account any additional area covered by a robot as it travels away from the meeting locations (large black dots). This figure shows the resulting divisions for six active robots and the designated B_i values used to reference the height of the horizontal rectangular areas.

Given a particular vertical stripe to cover with r active robots, let $A_1, A_2, \ldots, A_r$ denote the total assigned coverage area for each robot, with robot i covering A_i and robot $i = 1$ covering the area that is furthest away from the meeting location. Since each stripe is vertical, its area is $h\frac{w}{n} = hw'$. We use B_i to denote the distance between the upper boundaries of each successive region (see Figure 2). This parameter can be varied to offset the additional cost of traveling. Each robot is assumed to cover one unit area per time step so the additional travel area covered by r_1 is simply twice the distance from the meeting point to its assigned coverage area or $2(h - B_1)$. This produces the following equation for calculating the area A_i of each region (equation 5):

$$A_1 = B_1 w' + 2(h - B_1) = B_1(w' - 2) + 2h \tag{2}$$

$$A_2 = B_2(w' - 2) + 2(h - B_1 - B_2) = B_2(w' - 4) + 2h - 2B_1 \quad (3)$$

$$\begin{aligned} A_3 &= B_3(w' - 4) + 2(h - B_1 - B_2 - B_3) \\ &= B_3(w' - 6) + 2h - 2B_1 = 2B_2 \end{aligned} \quad (4)$$

$$A_i = B_i(w' - 2(i-1)) + 2(h - \sum_{j=1}^{i} B_j) \quad (5)$$

Since each robot must cover the same area, we can determine the relationship between B_i and B_{i-1} by setting the generic area formula at i and $i-1$ equal to each other (Equation 6). The remaining A_i and B_i values can be determined by successively applying Equations 7 and 8. In Equation 7, A_1 is equated to $\frac{hw'}{r}$, producing equation 8 where B_1 is expressed using the number of active robots, r, the width of a stripe $w' = \frac{w}{n}$, and the height of the overall environment, h.

$$\frac{B_i}{B_{i-1}} = 1 + \frac{4}{w' - 2i} \quad (6)$$

$$A_1 = \frac{hw'}{r} = B_1(w' - 2) + 2h \quad (7)$$

$$B_1 = \frac{\frac{hw'}{r} - 2h}{w' - 2} = \frac{h(w' - 2r)}{r(w' - 2)} \quad (8)$$

The limitation of this single stripe strategy is that the number of active robots must be less than half of a single stripe's width, $r < \frac{w'}{2}$. This does not present a problem for the analyzed environment, where it is assumed that $w = h >> k$. Therefore, from the single stripe strategy, each stripe can be divided into r equally sized areas with minimal repeated coverage and thus, eliminating the overhead coverage time from the Stripes equation.

3.3 Lower Bound

In this section, we establish a lower bound on the expected network formation time that can be achieved by any algorithm. It is easy to see that the best coverage time for any area occurs when all of the robots are active at the beginning and the area is evenly split between all of them. Therefore, the absolute lower bound for total coverage time, regardless of algorithm, is $T(A, k) = \frac{A}{k}$. For the case when the robots are uniformly distributed, we can obtain a better bound using the following result.

Lemma 3 ([10], pp.45). *When n balls are assigned uniformly at random to n bins, with probability at least $1 - \frac{1}{n}$, no bin has more than $\alpha = \frac{\log n}{\log \log n}$ balls in it.*

Now consider any network formation strategy for k robots in area A. During the execution of this strategy, we divide the coverage process into epochs where i^{th} epoch ends when all active robots cover a (previously uncovered) total area of A/k. Let S_i denote the subset of A covered during epoch i. Let E_1 be the event that no S_i has more that α balls. By Lemma 3, E_1 happens with probability $(1 - 1/k)$.

When E_1 happens, the maximum number of robots in S_1 is α. Therefore, the minimum time it takes to cover S_1 is $T_1 = \frac{A}{k\alpha}$. There will be α new robots in S_2, therefore its coverage time T_2 is at least $\frac{A}{k2\alpha}$. Similarly, the i^{th} epoch will last at least $T_i = \frac{A}{ki\alpha}$ steps. Since there are k epochs, the total time for all epochs to finish will be:

$$\sum_{i=1}^{k} \frac{A}{ki\alpha} \approx \frac{A}{\alpha k} \log k = \frac{A}{k} \log\log k \tag{9}$$

When E_1 does not happen (with probability $\frac{1}{k}$), the coverage time is at least A/k as discussed earlier. This gives us the lower bound on the expected coverage time of any algorithm.

Lemma 4. *The expected time for k robots to cover rectangular area A is at least* $(1 - \frac{1}{k})\frac{A}{k}\log\log k + (\frac{1}{k})\frac{A}{k}$.

Ignoring $o(\frac{1}{k^2})$ terms, we establish a lower bound of $\Omega(\frac{A}{k}\log\log k)$. Using Lemmata 2 and 4, we establish our main result:

Theorem 1 *The performance of the Stripes strategy is within a factor* $O(\frac{\log^2 k}{\log\log k})$ *of the optimal performance for a rectangular environment where* $w \geq h \gg k$.

3.4 Split and Cover

In this section, we discuss the performance of the following intuitive and natural network formation strategy: The first robot moves up and down the environment following a boustrophedon path as before. When it meets an undiscovered node, the two nodes split the undiscovered parts of the environment evenly and recursively cover their assigned partitions (see Figure 3). The justification for this natural "Split-and-Cover" strategy is that since the

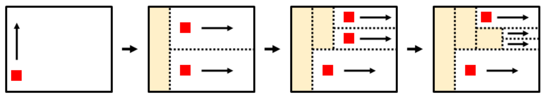


Fig. 3. Split and Cover Strategy

nodes are scattered uniformly in the environment, each split should divide the load equally between discovered robots, therefore balancing (and intuitively minimizing) the network formation time.

It turns out that Split-and-Cover is not an effective strategy for large environments for the following reason: suppose that when the split happens, one of the partitions has a very small number of robots. Even though this event has a small probability for a single split, as the number of robots (and hence the number of splits) increases, it becomes probable. When such an imbalance occurs, the algorithm can not recover from it. A small number of robots must cover a large environment making Split-and-Cover inefficient. We will further justify this argument with simulations and show that the Stripes algorithm is much more efficient than Split-and-Cover in unsaturated environments.

4 Simulations

In this section, we compare Stripes and Split-and-Cover in simulations. We also present simulation results that demonstrate the effect of the number of Stripes on the performance of the Stripes algorithm. Our simulations were performed by representing each of the individual robots as a point within the rectangular world. Each robot can be assigned to sweep along a continuous piecewise linear path. As described in the Section 3, an active robot can detect an inactive robot when it is within communication range. Each robot moves one unit distance per unit time and maintains an internal clock, to represent the time with respect to the start of the first robot. Robots that meet can synchronize clocks. We place the robots uniformly at random within the environment at the start of each trial. The simulation runs the algorithm exactly as it is stated in Section 3.4.

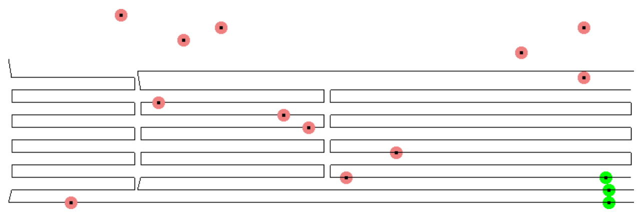
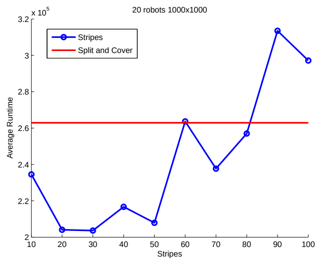


Fig. 4. This figure shows the simulation of the Stripes algorithm. Green circles are active robots, and red circles are inactive. The black lines represent the paths of active robots within the current stripe.

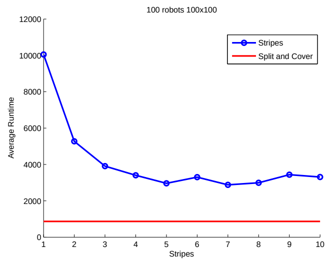
4.1 Results

In the first simulation, we compare Stripes and Split-and-Cover in environments that are sparse and dense with respect to the number of robots per area.

Figure 5(a) shows the results of the Split-and-Cover and Stripes algorithms in an environment where the concentration of robots is low. The simulations use a 1000x1000 environment and 20 robots. The plotted values are the average time to completion of 1000 trials, and the value for each number of stripes uses the same set of initial robot distributions. From these simulations, we conclude that (i) the performance of the Stripes algorithm is sensitive to the number of stripes, and (ii) in sparse environments (with a "good" stripe-number selection) Stripes outperforms Split-and-Cover. This is also justified by Figures 6(a) and 6(b), where we plot the histogram of running times of the Stripes algorithm (with 25 Stripes) and Split-and-Cover. Stripes not only outperforms Split-and-Cover on the average but also its worst case performance is considerably better.

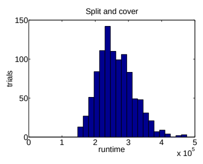


(a) Sparse environment

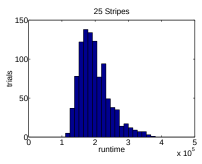


(b) Dense environment

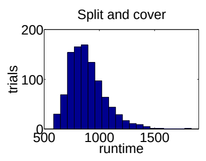
Fig. 5. Comparison of Split-and-Cover strategy with Stripes algorithm for varying number of stripes. The plotted values are the average time to completion of 1000 trials. **Left:** 1000×1000 unit world with 20 robots. **Right:** 100 × 100 unit world with 100 robots.



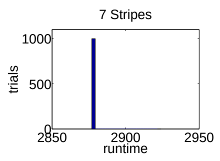
(a) Split-and-Cover Sparse



(b) Stripes Sparse



(c) Split-and-Cover Dense



(d) Stripes Dense

Fig. 6. Left: Distribution of time to completion for Split-and-Cover(6(a)), and for Stripes(6(b)) in a sparse environment.**Right:** Distribution of time to completion for Split-and-Cover(6(c)), and for Stripes(6(d)) in a dense environment.

On the other hand, when the concentration of robots is high, Split-and-Cover outperforms Stripes (Figures 5(b),6(c),6(d)). There are two reasons that this is true. First, for Split-and-Cover, the unbalanced split described in the previous section occurs infrequently when the concentration of robots is high. Second, for Stripes the overhead cost of meeting up to evenly distribute the coverage of a stripe becomes very large compared to the discovery time. In most instances, saturation took place after the fist stripe due to the high concentration of the robots. Therefore, the running time of Stripes was often given by the time to discover the first stripe followed by a parallel scan with 7 robots. In conclusion, simulation results suggest that Stripes should be used in sparse environments with the number of stripes equal to the number of robots. In dense environments, split-and-cover algorithm is expected to yield better performance.

5 Experiments

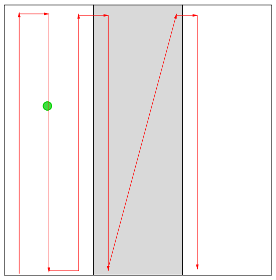
We demonstrate the feasibility of the Stripes algorithm using a small team of three Acroname Garcia robots shown in Figure 8(c). The robots are each equipped with ARM/risk PCs, and wireless network adapters. When configured to work in ad-hoc mode, the robots form a wireless sensor network, each capable of determining its own set of neighbors in the network graph.

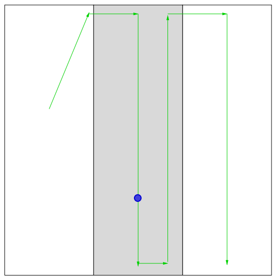
Currently, our robots do not have visual localization capabilities. Therefore, we rely on motor encoders and dead reckoning for position information. We also utilize an external stereo camera as a means of determining the robot positions off-line (to obtain ground truth). The external camera allows us to compute robot positions, but also limits the size of the work area to the field of view of the camera. Our experiments take place in a square workspace of 7.5 meters by 7.5 meters. These experiments where conducted inside of the institute gymnasium.

In practice, it would be useful to use wireless connectivity and connection strength to determine when an inactive node has been detected. However, in this setup the nodes would have to be separated by very large distances before observing a noticeable decline in signal strength. Therefore, in order to demonstrate the algorithm in our restricted space, we simulate restrictions in communication by allowing inactive nodes to be discovered only when they are within a short range. Figure 7 shows the placement of the robots and their ideal strategies when executing Stripes. Figure 8(a) shows an image of the experimental setup from the external camera. The white lines superimposed on the image outline the workspace and stripe boundaries. Figure 8(b) shows actual robot trajectories during the experiment computed from the image to ground plane homography.

5.1 Results

For this small number of robots, we use a simpler single-stripe strategy. In the experiment, each stripe is selected so that it can be covered by a single robot


(a) Robot 1


(b) Robot 2

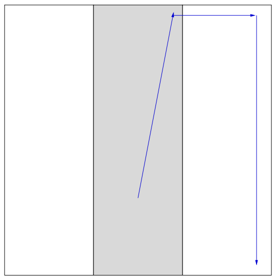
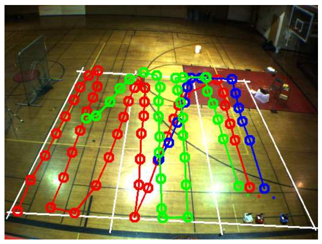
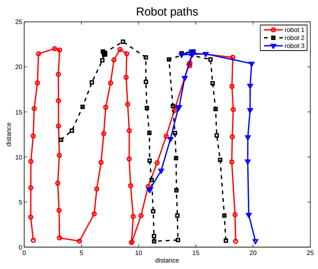

(c) Robot 3

Fig. 7. The ideal trajectories for robots 1, 2 and 3 (7(a), 7(b), and 7(c) respectively). The stripes are denoted by alternating white and gray. The first robot starts at the lower left corner. The circle in Figure 7(a) is the meeting location with robot 2. The third robot is discovered by robot 2. The meeting location is shown in Figure 7(b)


(a) paths



(b) trajectory

(c) Garcia robot platform

Fig. 8. 8(a) This image of the experimental setup shows the environment divided into three stripes. The paths traversed by each robot are shown in red, green and blue (lines superimposed). 8(b) This image shows the actual trajectory of each robot computed using the homography between the image and ground planes.

in 24 time units. This corresponds to three up and down motions (Figure 7). The initial placement of the robots is shown by the start of each of the three paths in Figure 8(a). Robots move approximately 1 meter per time unit. The total area of the workspace is approximately 56 m^2. The stripes are completed in 24, 16, and 8 time units respectively. Hence, the total completion time is 48 units. Note that the total coverage time for a single robot is 72 time units. The measured path lengths of each robot are 41.08, 25.848, and 13.26 meters respectively. In this setup, even though odometry errors resulted in deviations from the ideal trajectories in Figure 7, they were not significant enough to prevent proper execution of Stripes.

6 Extension to General Convex Environments

As mentioned earlier, the time to cover the environment is a trivial upper bound on the network formation time. This is because the undiscovered nodes are stationary, and by covering the environment, the first robot can guarantee that all nodes are discovered. In a convex environment, this coverage time is proportional to the area of the environment A. In the case of an unsaturated rectangular environment, we showed that the Stripes algorithm performs better than this upper bound.

In general environments, even when the environment is convex, the upper bound would be achieved. To see this, consider a long $1 \times A$ environment. In this environment, even when the robot locations are chosen randomly, the network formation time would be roughly A because the information is propagated sequentially and one of the robots is expected to be close to the "other" end of the environment. Therefore, the upper bound of A can not be beaten in some general convex environments.

7 Conclusion and Future Work

In this paper, we introduced a novel network formation problem with ties to freeze-tag and coverage problems. In the network formation problem, a robot tries to propagate a piece of information to other robots with unknown locations as quickly as possible. Once a robot is discovered, it joins in the information propagation process by searching for other robots.

We presented an algorithm for rectangular environments and analytically proved that its performance is within a logarithmic factor of the optimal performance. We demonstrated the utility of the algorithm further with simulations and a proof-of-concept implementation.

In our future work, we will address network formation in general environments. We will start with general convex environments. We believe that the Stripes algorithm can be modified to achieve good performance in such scenarios (compared to the optimal performance). Arbitrary environments, represented as polygons with holes, seem to be more challenging due to the lack of a natural way of partitioning the environment. We plan to study this interesting and equally challenging problem in our future work.

References

1. S. Alpern. The rendezvous search problem. *SIAM J. Control Optim.*, 33(3):673–683, 1995.
2. S. Alpern and S. Gal. *The Theory of Search Games and Rendezvous.* Springer, 2002.

3. E. J. Anderson and S. P. Fekete. Asymmetric rendezvous on the plane. In *SCG '98: Proceedings of the fourteenth annual symposium on Computational geometry*, pages 365–373, New York, NY, USA, 1998. ACM Press.
4. E. Arkin, M. Bender, S. Fekete, J. Mitchell, and M. Skutella. The freeze-tag problem: How to wake up a swarm of robots, 2002.
5. E. M. Arkin, S. P. Fekete, and J. S. B. Mitchell. Approximation algorithms for lawn mowing and milling. *Computational Geometry*, 17(1-2):25–50, 2000.
6. H. Choset. Coverage of known spaces: The boustrophedon cellular decomposition. *Auton. Robots*, 9(3):247–253, 2000.
7. S. Deb, M. Médard, and C. Choute. Algebraic gossip: a network coding approach to optimal multiple rumor mongering. *IEEE/ACM Trans. Netw.*, 14(SI):2486–2507, 2006.
8. A. Ganguli, J. Cortes, and F. Bullo. Multirobot rendezvous with visibility sensors in nonconvex environments. *IEEE Transactions on Robotics*, Nov. 2006. To appear.
9. S. Martinez, F. Bullo, J. Cortes, and E. Frazzoli. On synchronous robotic networks - Part II: Time complexity of rendezvous and deployment algorithms. *"IEEE Transactions on Automatic Control"*, 2007. To appear.
10. R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge Press, 2000.
11. S. Poduri and G. S. Sukhatme. Achieving connectivity through coalescence in mobile robot networks. In *International Conference on Robot Communication and Coordination*, Oct 2007.
12. S. Poduri and G. S. Sukhatme. Latency analysis of coalescence in robot groups. In *IEEE International Conference on Robotics and Automation*, 2007.
13. N. Roy and G. Dudek. Collaborative robot exploration and rendezvous: Algorithms, performance bounds and observations. *Auton. Robots*, 11(2):117–136, 2001.
14. L. Thomas and M. Pikounis. Many-player rendevous search: stick together or split and meet? *Naval Research Logistics*, 48:710–721, 2001.

Appendix

A Saturated Environments

In a saturated environment where $k \geq h$, the performance of Stripes can be improved with a simple modification after the number of active robots reach the height of the environment. The main difference between the coverage time for a saturated environment and an unsaturated environment is how the $k \geq h$ constraint affects the coverage of a single stripe.

In a saturated environment, after the number of active robots in a stripe, r, exceeds the height of a stripe, it is no longer possible to divide a stripe into r equal, non-overlapping regions. However, it is possible to line up the active robots along the height of the environment. Afterwards, they can cover the remaining portion of the environment without meeting again. Therefore, when $r \geq h$, the coverage time for a stripe becomes $\frac{w}{n}$ where n is the number of stripes. We will call any stripe where this occurs as saturated.

This changes the original upper bound equation (Equation 1) to contain an additional term. For the saturated environment, the term in the logarithm will instead be the number of non-empty stripes until stripes become saturated or simply h. The remaining area $A' = A(1 - \frac{h}{n})$ can be covered in time $\frac{A'}{h}$. Therefore, the coverage time of an $A = h \times w$ environment is bounded by:

$$\left(\frac{A}{n} + \frac{A}{n}\log h\right) + w - \frac{A}{n} = \frac{A}{n}\log h + w \tag{10}$$