2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Congress Center Hamburg
Sept 28 - Oct 2, 2015. Hamburg, Germany

Path Planning for Optimizing Survivability of Multi-Robot Formation in Adversarial Environments

Yaniv Shapira and Noa Agmon
Computer Science Department
Bar Ilan University, Israel
{shapiry9,agmon}@cs.biu.ac.il

***Abstract*— Multi robot formation is a canonical problem in robotic research. The problem has been examined in neutral environments, where the robots' goal is usually to maintain the formation despite changes in the environment. The problem of multi robot formation has been motivated by natural phenomena such as schools of fish or flocks of birds. While in the natural phenomena the team behavior is responsive to threats, in robotics research of team formation, adversarial presence has been ignored. In this paper we present the problem of *adversarial formation*, in which a team of robots travels in a connected formation through an adversarial environment that includes threats that may harm the robots. The robots' goal is, therefore, to maximize their chance of traveling through the environment unharmed, where the formation may be used as a mean to achieve this goal. We formally define the problem, present a quantitative measure for evaluating the survivability of the team, and suggest possible solutions to a variant of the problem under certain threat characteristics, optimizing different team survivability criteria. Finally, we discuss the challenges raised by transitioning the discrete representation to a continuous environment in simulation.**

I. Introduction

Multi-robot formation is a canonical task in robotic research. A team of robots is required to achieve a formation, and/or to maintain it. When achieving a formation, the goal of the robots is to distribute about the given formation, usually while minimizing time and avoiding collisions [1], [2], [3]. After the robots are organized in a formation, they are required to travel while maintaining it despite changes in the environment, such as obstacles. Formation maintenance usually aims at minimizing the deviation from the desired formation during the execution [4], [5], [6].

The problem of robot formation was initially inspired by natural phenomena, from animals (e.g., a school of fish or a flock of birds [7]) and humans (e.g., a convoy or an infantry unit [8]). In these natural phenomena, the formation of birds, fish, vehicles or humans travel in environments where the team is threatened by some adversarial existence. A school of fish may be threatened by predatory fish, a flock of birds may be targeted by a bird of prey. Similarly, a team of fire fighters need to search and rescue casualties in a wildfire, and a convoy of humanitarian aid in a disaster area may be targeted by some external parties[1].

[1] http://www.telegraph.co.uk/news/worldnews/asia/philippines/10445615/Typhoon-Haiyan-aid-convoys-come-under-fire-as-relief-operation-becomes-logistical-nightmare.html

Motivated by the adversarial presence in these natural examples of formations, and by the current use of robots in areas containing threats (from Mars rovers to UAV presence in war zones), we introduce a new problem: *robot formation in adversarial environments* (or *adversarial formation*, in short). In this problem, the team of robots travels in an adversarial environment, where possible threats exist and may harm the robots. In this novel idea the adversary and the environment influence the optimal formation that the group should be traversing in. The goal of the research is to identify possible types of threats the robots may face and the impact of the formation, in which the robots travel in, on the safety of each robot and the formation. This will give us the ability to minimize the chance of each robot to get hurt and thus maximize the formation survivability, i.e., the chances of the robots to pass through the area unharmed. To the best of our knowledge, adversarial influences have so far been *ignored* in robotic formation research.

The paper presents a broad definition of the problem, providing building blocks for possible instances of the problem, in different domains and under different optimization criteria. We define the problem of *Safe Robotic Adversarial Formation* (SRAF), which aims to find the formation that optimizes different survivability criteria, mainly the probability of the team members to pass through the environment unharmed. We show that while in general the problem is most likely intractable, under some assumptions the problem may be solved optimally in polynomial time.

We have implemented algorithms for determining a formation that optimizes survivability in ROS/Stage [2], and following extensive simulation experiments, we have adjusted the initial model in order to account for transition from a discrete representation to a continuous environment. We present simulation results showing that the adjusted model performs significantly better than the discrete one, yielding a complete, optimal, representation of realistic scenarios.

II. Background and Related work

The study of multi-robot formation is wide and consists of two main problems: getting a group of robots to create a given formation [1], [2], [3], [9] and maintaining it while traversing an environment [4], [5], [6], [10], using several methods and techniques.

[2] http://wiki.ros.org/

Papers on formation control and maintaining the formation configuration while moving to a goal, avoiding obstacles and collisions with other robots are extensive. In [4], [11] behaviors, or motor-schemas [12], of *move-to-goal*, *avoid-static-obstacles*, *avoid-robot* and *maintain-formation* were introduced. Those schemas are implemented by each robot in the formation and generate a behavioral response for the robot which include direction and magnitude of movement. In [5] the balance between global information and local information is being presented. These conjunctions between different levels of local and global knowledge are implemented on the task of maintaining a formation. Our study deals also with maintaining a formation while avoiding other robots but also need to include an additional behavior of minimizing the exposure to an external threat.

Desai *et al.* [13] coined and defined the phrase *control graph* as a labeled digraph whose vertices are the robots in the formation and whose edges are connections between followers and their leader. This research deals with controlling and transforming one formation to another in order to avoid obstacles. As opposed to [13] and other works which dealt with obstacle avoidance and shape transformation ([4], [5], [11], [6], [10], [14]), *threats may not considered as obstacles* since robots can, and sometimes *must*, travel through them. In this research we would like to determine the strength of the formation rather than finding methods to avoid threats.

Papers that took into account the presence of an adversary such as [15], [16], [17] present algorithms and methods for risk avoidance. These works examine the path planning problem of a single individual, in order to bypass and avoid the adversary's threats. Our study deals with multi-robot systems that traverse *through* the threats, while maintaining a connected formation. In [18], [19], [20], [21] a team of robots is considered in the problem of robot navigation under threats, but the robots carry out their traverses one at a time, sequentially, and do not travel in a formation.

In [22] the problem of adversarial coverage is defined as generating a path that visits each point in the grid world at least once by a single robot, where the environment includes threat points that may stop the robot. The environment is a grid of $n \times n$ cells (with obstacles), where each cell is associated with a probability that an existing threat will stop the robot from continuing its movement. While our work is similar in the existence of threat points, the task is inherently different: while in [22] the robot travels through the entire world, here the team must find the most efficient way to pass through the world unharmed.

III. The Problem Of Adversarial Formation

In this section we formally define the problem of multi-robot formation in adversarial environments, or adversarial formation in short, which deals with finding a formation and a path that, given a threat, maximizes the chances of the robots to travel through the adversarial environment unharmed. In order to do that, we first describe possible threat characteristics.

A. The Environment and the Threats

There are different types of threats that the adversary can execute and aim at the formation. These threats may not be posed by an actual adversary, but similarly to the existence of an adversary in distributed systems, this can model the "worst case" behavior of the system given a threat [23]. Hence modeled threats that are considered in this paper include avalanche, lightnings, tsunami and bush fire, as well as sniper fire and bomb attack.

The threats are characterized using two properties: time and space.

Time: Different threats have different duration while executed, e.g., the duration of a bomb explosion is shorter than an exposure to a radiation cloud. Threat duration is measured on a continuous time-line scale where on one end of the scale the execution time is instantaneous, and on the other end it is indefinite. In addition, time-invariant threat is defined as a threat that is not depended on time to be executed.

Space: This property is characterized by *traversability*, *concealment*, *spacial dimension* and *range of influence*. *Traversability* (in this context) indicates whether the environment is crossable after execution. *Concealment* indicates whether a robot can be concealed from the threat, either by another robot or by an object in the environment. For example, a robot that conceals its peers under a sniper threat as opposed to an earthquake where all the team is exposed to it. *Spacial dimension* of the threat can be of $1D$, $2D$ or $3D$. *Range of influence* indicates the area that the threat dominates and in which it can harm the team. It is assumed that the probability of being hit by a threat is monotonically non-increasing as the distance from the threat increases.

In this paper we consider time invariant, traversable threats. We model the environment as an obstacle-free grid, in which each cell is of the size of a robot, and the robots travel through one cell per time cycle. Each threat may influence a subset of the environment, and it is assumed that we are given a grid map of the world in which each cell is associated with a probability that a robot passing through it may be harmed (similar to the map in [22]).

B. Problem Definition

A team of k robots $R = \{r_1, \ldots, r_k\}$ needs to traverse through an adversarial environment. We denote the position of a robot r_i in the environment at time t by $p_i(t)$. The probability that a threat exists at a point p at time t is denoted by $P_p^E(t)$, $0 \leq P_p^E(t) \leq 1$. The probability that robot r_i is stopped by the threat at time t, denoted by $P_i^H(t)$, depends on the type of the threat, its distance from the robot and other possible factors. We say that a robot *survives* a threat, if it manages to pass through it unharmed. We assume that $P_p^E(t)$ and the threat characteristics are known.

The survivability measure is composed of two parameters: the survivability of each single robot, and the survivability of the team.

Definition 1: **Individual Survivability** (IS_i^t): the probability that a robot r_i passes a threatened point p at time t unharmed. For each $r_i \in R$, this is calculated as:

$IS_i^t = (1 - (P_p^E(t) \cdot P_i^H(t)))$

The **Team Survivability Criterion** (TSC) at a given time t may be defined either according to the most vulnerable individual, denoted by TSVI^t, or by the product of the individual survivability of all team members, denoted by TSPR^t. Formally,

$\mathsf{TSVI}^t = \min\{IS_1^t, IS_2^t, \cdots, IS_k^t\}$

$\mathsf{TSPR}^t = \prod_{i=1}^{k} IS_i^t$

Note that TSVI, TSPR are probabilities, thus resides between [0,1].

Definition 2: A **Path** is a sequence of different positions of the teammates in the environment. The path is the progression of the team from the source to the target spot, lasting T time steps, and is denoted by S. Clearly, the number of different paths depends on the environment, and is exponential in the state space.

The **Path Survivability Criterion** (PSC) is a measure that calculates the robustness of the team along the path, subject to the team survivability criterion:

- PMM: Choose a path that maximizes the step with minimal survivability, i.e.,
  $\mathsf{PMM} = argmax_S\{\min(\mathsf{TSC}^t | 1 \leq t \leq T)\}$
- PTS: Choose a path that maximizes the total survivability of the robots along the path, i.e.,
  $\mathsf{PTS} = argmax_S\{\prod_{t=1}^{T} \mathsf{TSC}^t\}$

We define the *Safe Robotic Adversarial Formation* (SRAF) problem as follows:

Definition 3: Given a team of k robots that travel through an adversarial environment $\mathbf{E}$ under given threat characteristics, find the formation and the path in which the team should travel in, subject to a given TSC and PSC.

IV. Solving The Adversarial Formation Problem

In the general case of the SRAF problem, a team of robots needs to traverse along the threat points, where in every time step each teammate (except for the global leader) needs to sense at least one teammate, meaning the control graph must be connected. The general case does not indicate the configuration and the pattern of the formation, that is the formation of the robots adjusts to the path it needs to travel in, and specifically: the formation may change along time. A variant of SRAF where the pattern is indicated before will be examined in the next two cases:

1) *Rigid Formation Problem*: A rigid formation is a layout of robots that stays fixed throughout the traversing period, without rotation of the formation (e.g., the four common formation types: column, line, wedge and diamond).
2) *Rotating Rigid Formation Problem*: A rotating rigid formation is a rigid formation with the ability to turn around and rotate while traversing. Throughout the traversing period the pattern of the formation stay the same (see Fig. 3). The resolution of the rotation defines the accuracy of the survivability.

We suggest a two step process for solving the SRAF problem:

Step 1: Constructing a graph induced from the formation placed in different parts of the environment. The graph is a weighted digraph where the weight of every incoming edge is the survivability of its vertex (see Fig. 1).

Step 2: Finding a path on the graph from the source point to the target point that solves the SRAF problem.

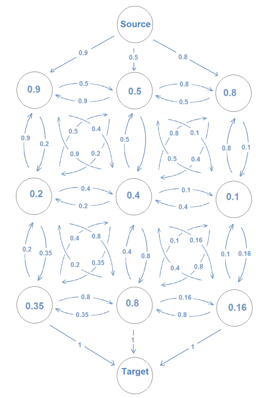

Fig. 1. The constructed graph used for path finding. The weights on every incoming edge is the survivability of its vertex. The weights of all incoming edges to the target vertex equal 1

A. Rigid Formation

In the *Rigid Formation Problem*, given a formation and the threat characteristics, we would like to find the path in which the team should travel in, in order to fulfill its TSC and PSC. This special case can be solved in polynomial time. The first step in the algorithm is constructing a graph that its vertices are all possible positions of the formation pattern in the environment, and the vertices utility are the team survivability when placed at a specific position. The edges of the graph are the possible moves from one position to another, while maintaining the rigid configuration. The graph represents all possible positions the formation can be located in the environment, and the edges are the transition from any adjacent positions at one time step (see Fig. 2). The second stage of the algorithm is to find the best path from the starting position, through the environment, towards the end position that meets the desired PSC (see Algorithm 1). The input to the algorithm is the designated formation and the environment map (grid).

B. Rotating Rigid Formation

In the *Rotating Rigid Formation Problem*, given a formation pattern and threat characteristics, we would like to find the path in which the team should travel while rotating and manipulating the formation (but preserving the given pattern) in order to fulfill the TSC and PSC. This special case can also be solved optimally in polynomial time. The first step in the algorithm is to obtain all possible formations that can be created from the pattern by rotating it, where the number of possibilities depends on the resolution of the rotation that we would like to use: higher resolution will yield a large number of possibilities and a lower resolution -

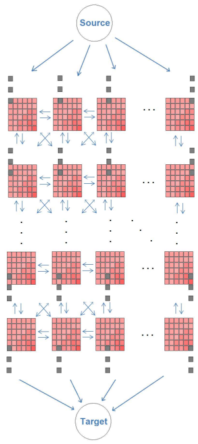

Fig. 2. A graph construction of the *Rigid Formation Problem*. This example contains an environment of 6×6 grid cells, and a team of three robots in a column formation. Each vertex in the graph is one of the possible positions of the column formation in the environment and the edges are the transitions from one position to its adjacent positions

Algorithm 1 *Rigid Formation Path Finding*

Input:RF The configuration of the rigid formation
Input:Env The environment with threat distribution (size $m \times n$)
Let $Graph \leftarrow \emptyset$
Step 1:
Place RF on the Upper-Left corner of Env
for $i \leftarrow 1$ to $n + FormationVerticalSize$ **do**
  for $j \leftarrow 1$ to $m + FormationHorizontalSize$ **do**
    $Vertex(j, i) \leftarrow$ Survivability of RF on the current Env position
    $Graph \leftarrow Vertex(j, i)$
    Shift RF one step to the right
  end for
  Place RF on the left side of the next line
end for
Link every vertex in $Graph$ with its adjacent vertices. Set the weight of every outgoing edge to be the adjacent vertex's TSC

Step 2:
if PSC $==$ PMM **then**
  Perform *Widest Path Algorithm* on $Graph$
else if PSC $==$ PTS **then**
  Perform *Shortest Path Algorithm* on $Graph$
end if

a small number of possibilities. For each possibility, a graph is constructed in a similar way that it is done in the *Rigid Formation* Algorithm. In the next step we would like to link vertices from different graphs that contains a joint agent, meaning an agent that is located at the same position on both graphs and is acting as a pivot of the rotation. Linking all possible vertices will produce a multi-layer graph and on that graph we will find the best path from the starting position, through the environment, towards the end position that meets the desired PSC. (see Algorithm 2).

Constructing a graph for the Rigid Formation is done in polynomial time. Expanding it to contain the Rotating Rigid formation, it is still done in polynomial time in the resolution of rotation.

C. Path Survivability Criterion - Complexity Analysis

Given a graph that its vertices are all the possible positions of the rigid or rotating rigid formation, finding a path that

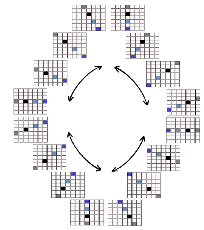

Fig. 3. The possible rotating options of the column formation in a given resolution. With each possibility a graph is constructed using the Rigid Formation algorithm

Algorithm 2 *Rotating Rigid Formation Path Finding*

Input:FP The formation pattern
Input:Env The environment with threat distribution (size $m \times n$)
Let $MultiGraph \leftarrow \emptyset$

Step 1:
Let $Vector_{rf} \leftarrow$ All possible Rigid Formations created by rotating FP
for each RF in $Vector_{rf}$ **do**
  Let $Graph_{rf} \leftarrow \emptyset$
  Place RF on the Upper-Left corner of Env
  for $i \leftarrow 1$ to $n + FormationVerticalSize$ **do**
    for $j \leftarrow 1$ to $m + FormationHorizontalSize$ **do**
      $Vertex(j, i) \leftarrow$ Survivability of RF on the current Env position
      $Graph \leftarrow Vertex(j, i)$
      Shift RF one step to the right
    end for
    Place RF on the left side of the next line
  end for
  Link every vertex in $Graph_{rf}$ with its adjacent vertices and set the weight of every outgoing edge to be the adjacent vertex's TSC
  $MultiGraph \leftarrow Graph_{rf}$
end for

for each $Graph_{rf_a}$ and $Graph_{rf_{a+1}}$ in $MultiGraph$ **do**
  Mutually link vertices from $Graph_{rf_a}$ containing joint agents that are also placed on the same positions in vertices from $Graph_{rf_{a+1}}$
end for

Step 2:
if PSC $==$ PMM **then**
  Perform *Widest Path Algorithm* on $MultiGraph$
else if PSC $==$ PTS **then**
  Perform *Shortest Path Algorithm* on $MultiGraph$
end if

meets the desired PSC is done by using graph theory method, with polynomial time complexity:

1) The PMM criterion: finding a path that its minimal edge is the maximal one among all other possible paths, is equal to the *Widest Path* problem, which can be solved in a polynomial time with some modification to shortest path algorithms [24].
2) The PTS criterion: finding a path that maximizes $\Pi_{t=1}^{T}\mathsf{TSC}^t$, is equal to running the shortest path algorithm (runs in polynomial time) [25]. In the shortest path problem the goal is to find a path that minimizes $\Sigma_{t=1}^{T} W^t$, where W^t is the weight of the visited edges (t). The equivalence between $\min \Sigma$ and $\max \Pi$ is obtained by using $-\log()$, which gives:

$$\max \Pi_{t=1}^{T}\mathsf{TSC}^t \equiv \min \Sigma_{t=1}^{T} - \log(\mathsf{TSC}^t)$$

D. From Discrete To Continuous

The costs of the edges that were set in the above algorithms consider a discrete model of the world. In order to use the algorithm in a more realistic manner, the edges' costs should be altered to depict a continuous model of the world. The distance that a robot is passing in a horizontal or a vertical path is shorter from the distance it is traveling in a diagonal path. This distance is not taken into account when setting the edges' costs. Furthermore, the threats that the robot is exposed to when passing through a diagonal path are not considered in the edge's cost (see Fig. 4).

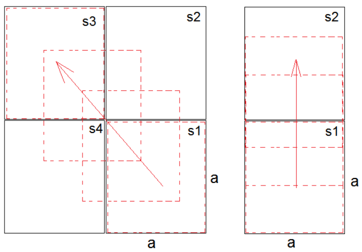

Fig. 4. The two possible paths (without loss of generality) a robot can take in order to move between cells. Left: a diagonal path where the robot is moving from a cell with survivability s_1 to a cell with survivability s_3, and passing along the way the cells with survivability s_2 and s_4. Right:a vertical way where the robot is moving from the cell with s_1 survivability to the cell with s_2 survivability.

In a more realistic model of the world, at every time t different parts of the robot are located on different cells. To find the survivability of the movement we should calculate the survivability at each time step t until the robot reaches either an adjacent (up/down/right/left) or diagonal cell.

1) Moving to a diagonal cell: If a cell size (and robot size) is aXa, then the robot needs to pass a distance of $\sqrt{2}a$ for reaching the center of a diagonal cell. While leaving the *origin cell*, the area of the robot that is exposed as a function of x (distance) is: $f(x) = (a - \frac{x}{\sqrt{2}})^2$. While reaching the *target cell*, the area of the robot that is exposed as a function of x (distance) is: $f(x) = (\frac{x}{\sqrt{2}})^2$, and the area of the robot that is passing on an *adjacent cell* is : $f(x) = (\frac{x}{\sqrt{2}})\cdot(a-\frac{x}{\sqrt{2}})$

The survivability of a robot in continuous mode, (denoted as $Surv$), while crossing a diagonal path is $\ln(Surv) = \int_{x=0}^{\sqrt{2}a} ln(s_3{}^{(\frac{x}{\sqrt{2}})^2} \cdot s_2{}^{(\frac{x}{\sqrt{2}})\cdot(a-\frac{x}{\sqrt{2}})} \cdot s_4{}^{(\frac{x}{\sqrt{2}})\cdot(a-\frac{x}{\sqrt{2}})} \cdot s_1{}^{(a-\frac{x}{\sqrt{2}})^2})$. (Where s_1,s_2,s_4 and $_3$ are the survivability of the origin, adjacent and the target cells respectively, see figure 4).

This equation is inspired from the survivability in the discrete model: $Surv = \prod^{length\ of\ path}$ which is equivalent to $\ln(Surv) = ln(\prod^{length\ of\ path}) = \sum^{length\ of\ path} ln()$

Therefore, we will get: $\ln(Surv) = \int_{x=0}^{\sqrt{2}a} ln(s_3{}^{(\frac{x}{\sqrt{2}})^2} \cdot s_2{}^{(\frac{x}{\sqrt{2}})\cdot(a-\frac{x}{\sqrt{2}})} \cdot s_4{}^{(\frac{x}{\sqrt{2}})\cdot(a-\frac{x}{\sqrt{2}})} \cdot s_1{}^{(a-\frac{x}{\sqrt{2}})^2}) = 0.4714 \cdot a^3 \cdot ln(s_3) + 0.2357 \cdot a^3 \cdot ln(s_2) + 0.2357 \cdot a^3 \cdot ln(s_4) + 0.4714 \cdot a^3 \cdot ln(s_1)$.

$Surv = s_3{}^{0.4714 \cdot a^3} \cdot s_2{}^{0.2357 \cdot a^3} \cdot s_4{}^{0.2357 \cdot a^3} \cdot s_1{}^{0.4714 \cdot a^3}$.

2) Moving to a vertical or horizontal cell: If the cell size (and robot size) is aXa then the robot needs to pass a distance of a before reaching the center of an adjacent cell. While leaving the *origin cell*, the area of the robot that is exposed as a function of x (distance) is: $f(x) = a \cdot x$. While reaching the *target cell* the area of the robot that is exposed as a function of x (distance) is: $f(x) = a^2 - a \cdot x$.

The survivability of a robot in continuous mode, (denoted as $Surv$), while crossing a vertical or horizontal path is $\ln(Surv) = \int_{x=0}^{a} ln(s_2{}^{a \cdot x} \cdot s_1{}^{a^2 - a \cdot x})$. (Where s_1,s_2 are the survivability of the origin cell and the target cell respectively, see figure 4).

This equation is inspired from the survivability in the discrete model which was shown in the previous section. Therefore, we will get: $\ln(Surv) = \int_{x=0}^{a} ln(s_2{}^{a \cdot x} \cdot s_1{}^{a^2 - a \cdot x}) = 0.5 \cdot a^3 \cdot ln(s_1) + 0.5 \cdot a^3 \cdot ln(s_2)$.

$Surv = s_1{}^{0.5 \cdot a^3} \cdot s_2{}^{0.5 \cdot a^3}$.

3) Extending IS_i^t *definition:* The survivability that was calculated here does not change the main outlines of the algorithms above, but rather extends the definition of *Individual Survivability* (IS_i^t). The *Individual Survivability* with its extended definition takes into account also the direction from which the robot arrived (the previous immediate cell) and not merely the survivability imprinted in each cell.

E. Experiments and Results

We performed a total of thirty experiments on three environments. Each environment had a different set of random threat probabilities scattered around in it. One set of experiments tested the algorithm under the continuous model of the world (see Fig. 5), while the second set tested the algorithm under the discrete model of the world (see Fig. 6). The results show the improvement of the team survivability when moving from discrete to continuous calculation of the survivability criterion.

1) The environment: Figures 5 and 6 show an example of an experimental environment, which are an obstacle-free grid of 20×20. Each cell is associated with a probability for being stopped, ranging from light blue (for 0) to black (for 1), i.e., a darker color depict a higher probability that a robot passing through it may be harmed.

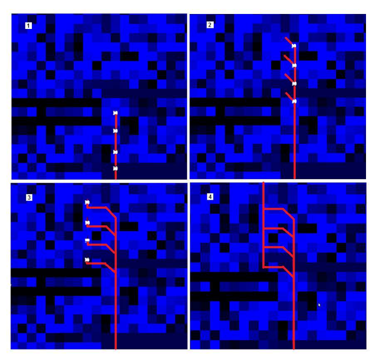

Fig. 5. One of the experimental environments (simulated with ROS/Stage). The red lines are an optimal paths followed by team of robots in a column formation while using the continuous representation algorithm. The team is progressing from the bottom (figure 1) to the top (figure 4).

2) The results: The experiments simulated (using ROS/Stage) a group of robots in a column formation with team survivability criterion TSPR, and path survivability

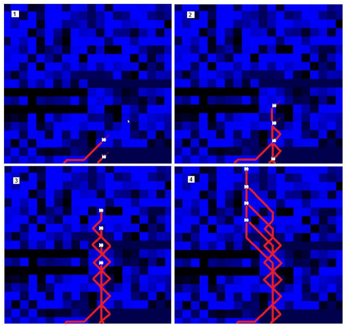

Fig. 6. The same environment as presented in 5, but showing the optimal paths selected by the algorithm using the original discrete representation. Again, the red lines are the paths that a group of robots in a column formation is following, where the group is progressing from the bottom (figure 1) till the top (figure 4).

criterion PTS. In every time step a sample of several points on each robot is taken. The results are a summation of all the sampled points along the path the robots traveled, produced from the algorithms. Each point is the $-\log(survivability)$ of the cell it resides in. The calculated summation is then being divided by the number of samples. Due to the nature of $\frac{\Sigma_{t=1}^{T} -\log(survivability)}{\#samples}$, the lower the result are, the better (see Fig. 7). In all three environments, continuous modeling is significantly more effective compared to discrete modeling (using t-test, $p-value < 0.1$ in Environment 1 and 2, and $p-value < 0.5$ in Environment 3).

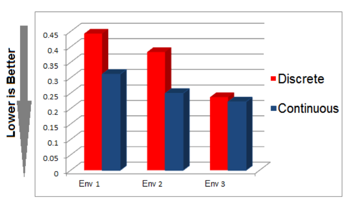

Fig. 7. The results of the experiments of all three environments. The red bars are of the discrete world modeling and the blue bars are of the continuous world modeling. The lower the result is the better.

V. Conclusions and Future Work

In this paper we introduced the problem of *Adversarial Formation*, where a team of robots is required to travel in a formation through an area that poses threats on the robots, while maximizing the chances of the robots to pass through this area safely. We have formally defined the problem, including possible threats posed by the adversary and examined the time complexity of finding a formation that maximizes the probability that the robots will not be harmed by the threat. This work sets the building blocks for a new problem, leaving numerous exciting directions for future work. One such direction includes introducing dependencies between the robots' locations. An important advantage of traveling in a formation is the ability of the robots to shield one another from danger. Adding such dependencies increases the complexity of finding an optimal solution on one hand, but significantly improves the chances of survivability on the other. Other team survivability measures can be considered, for example reorganization time after leaving the adversarial environment and cost of creating disconnected components in the control graph along the way.

References

[1] I. Suzuki and M. Yamashita, "Distributed anonymous mobile robots: Formation of geometric patterns," *SIAM Journal on Computing*, vol. 28, no. 4, pp. 1347–1363, 1999.

[2] X. Défago and A. Konagaya, "Circle formation for oblivious anonymous mobile robots with no common sense of orientation," in *Proceedings of the second ACM international workshop on Principles of mobile computing*. ACM, 2002, pp. 97–104.

[3] L. L. Whitcomb, D. E. Koditschek, and J. B. Cabrera, "Toward the automatic control of robot assembly tasks via potential functions: The case of 2-d sphere assemblies," in *Proc. of ICRA*. IEEE, 1992, pp. 2186–2191.

[4] T. R. Balch and R. C. Arkin, "Motor schema-based formation control for multiagent robot teams." in *Proceedings of the International Conference on Multi Agent Systems*. AAAI Press, 1995, pp. 10–16.

[5] L. E. Parker, "Designing control laws for cooperative agent teams," in *Proc. of ICRA*. IEEE, 1993, pp. 582–587.

[6] P. Ogren and N. E. Leonard, "Obstacle avoidance in formation," in *Proc. of ICRA*, vol. 2. IEEE, 2003, pp. 2492–2497.

[7] J. K. Parrish, S. V. Viscido, and D. Grünbaum, "Self-organized fish schools: an examination of emergent properties," *The Biological Bulletin*, vol. 202, no. 3, pp. 296–305, 2002.

[8] U. S. Army, "Fm 3-21.71 mechanized infantry platoon and squad (bradley)," 2010.

[9] S. Murata, H. Kurokawa, and S. Kokaji, "Self-assembling machine," in *Proc. of ICRA*. IEEE, 1994, pp. 441–448.

[10] J. Fredslund and M. J. Mataric, "A general algorithm for robot formations using local sensing and minimal communication," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 837–846, 2002.

[11] T. Balch and M. Hybinette, "Social potentials for scalable multi-robot formations," in *Proc. of ICRA*, vol. 1. IEEE, 2000, pp. 73–80.

[12] R. C. Arkin, "Motor schemabased mobile robot navigation," *IJRR*, vol. 8, no. 4, pp. 92–112, 1989.

[13] J. P. Desai, J. P. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 905–908, 2001.

[14] R. Fierro, A. K. Das, V. Kumar, and J. P. Ostrowski, "Hybrid control of formations of robots," in *Proc. of ICRA*, vol. 1. IEEE, 2001, pp. 157–162.

[15] M. Zabarankin, S. Uryasev, and P. Pardalos, "Optimal risk path algorithms," *Applied Optimization*, vol. 66, pp. 273–296, 2002.

[16] M. Likhachev and A. Stentz, "Goal directed navigation with uncertainty in adversary locations," in *Proc. of IROS*. IEEE, 2007, pp. 4127–4134.

[17] S. A. Bortoff, "Path planning for uavs," in *Proceedings of the American Control Conference*, vol. 1, no. 6. IEEE, 2000, pp. 364–368.

[18] M. Al Marzouqi and R. A. Jarvis, "Robotic covert path planning: A survey," in *IEEE Conference on Robotics, Automation and Mechatronics*. IEEE, 2011, pp. 77–82.

[19] A. Tews, M. J. Mataric, and G. S. Sukhatme, "Avoiding detection in a dynamic environment," in *Proc. of IROS*, vol. 4. IEEE, 2004, pp. 3773–3778.

[20] A. D. Tews, G. S. Sukhatme, and M. J. Mataric, "A multi-robot approach to stealthy navigation in the presence of an observer," in *Proc. of ICRA*, vol. 3. IEEE, 2004, pp. 2379–2385.

[21] Y. A. Teng, D. DeMenthon, and L. S. Davis, "Stealth terrain navigation," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, no. 1, pp. 96–110, 1993.

[22] R. Yehoshua, N. Agmon, and G. A. Kaminka, "Robotic adversarial coverage: Introduction and preliminary results," in *Proc. of IROS*, 2013.

[23] N. A. Lynch, *Distributed Algorithms*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1996.

[24] M. Pollack, "The maximum capacity through a network," in *Operations Research*, vol. 8, no. 5. INFORMS, 1960, pp. 733–736.

[25] E. W. Dijkstra, "A note on two problems in connexion with graphs," vol. 1, no. 1. Springer, 1959, pp. 269–271.