Multi-Robot Perimeter Patrol in Adversarial Settings

Noa Agmon, Sarit Kraus and Gal A. Kaminka
Department of Computer Science
Bar Ilan University, Israel
{segaln, sarit, galk}@cs.biu.ac.il

***Abstract*— This paper considers the problem of multi-robot patrol around a closed area with the existence of an adversary attempting to penetrate into the area. In case the adversary knows the patrol scheme of the robots and the robots use a *deterministic* patrol algorithm, then in many cases it is possible to penetrate with probability 1. Therefore this paper considers a *non-deterministic* patrol scheme for the robots, such that their movement is characterized by a probability p. This patrol scheme allows reducing the probability of penetration, even under an assumption of a strong opponent that knows the patrol scheme. We offer an optimal polynomial-time algorithm for finding the probability p such that the minimal probability of penetration detection throughout the perimeter is maximized. We describe three robotic motion models, defined by the movement characteristics of the robots. The algorithm described herein is suitable for all three models.**

I. Introduction

This paper discusses the problem of patrolling around a closed area by a team of robots (perimeter). The patrol problem requires to visit a target area repeatedly in order to monitor some change in the state of that area. We consider adversarial settings, in the sense that the adversary tries to enter the closed area. The problem of patrolling around an area with the existence of an adversary is applicable in many security applications. This problem can be also applicable in cases one wishes to model the worst case scenario that the system should deal with, for example toxic waste observance.

The problem of multi-robot patrol is derived from the fundamental problem of multi-robot coverage [4], and has received growing interest of its own (e.g. [1], [6]). Current patrol solutions offer deterministic algorithms for generating patrol paths for a team of robots and maintaining the patrol. Analysis of these algorithms concentrated on assuring frequency criteria in the patrolled area [6].

However, in adversarial settings the frequency criteria becomes less relevant. Consider the following scenario. We are given a cyclic fence of length 100 meters and 4 robots are required to patrol around the fence while moving in velocity $1m/sec$. Clearly, the optimal possible *frequency* of visits at each point around the fence is $1/25$, i.e., each location is visited once every 25 seconds. Assume that it takes an adversary 20 seconds to penetrate the area through the fence. If the robots move in a deterministic path, then the adversary can guarantee penetration if it simply enters through a position that was currently visited by the patrolling robot. On the other hand, if the robots move non-deterministically, then the choice of penetration position becomes less trivial.

Therefore in this paper we analyze non-deterministic patrol paths for a team of homogenous mobile robots patrolling around an area, under the assumption of an observing adversary trying to enter the area. We first divide the perimeter into segments such that each robot monitors one segment per time cycle. When the robots patrol around a closed area, a robot placed in segment i has a choice of going to segment $i-1$, $i+1$, or remaining at the same segment.

We consider three movement models of robots, characterized by different movement abilities of the robots. In the first, $\mathcal{DZCP}$, the robot has directionality associated with its movement, therefore if the robot is headed towards segment $i+1$ it will go straight to segment $i+1$ with probability p and turn backwards to segment $i-1$ with probability $1-p$. If it is directed towards segment $i-1$, then the probability of it going to segment $i-1$ is p, and to segment $i+1$ is $1-p$. The second model, $\mathcal{DCP}$, is a more realistic version of the $\mathcal{DZCP}$ model, in which the robot has cost related to turning around, i.e., if the robot turns around, then it stays in segment i. The last model is similar to a random walk, in which there is no directionality associated with the movement, i.e., for all i the robot goes to segment $i+1$ (right) with probability p and to segment $i-1$ (left) with probability $1-p$.

In the strong adversarial model we consider, in which the adversary knows the patrol scheme of the robots, it will choose to penetrate where it has lowest probability of being detected. We therefore describe a polynomial time algorithm for maximizing this probability, i.e., the maximin probability of penetration detection. We consider mainly the most realistic robotic model, $\mathcal{DCP}$, although the proposed algorithm works in all three models under minor modifications. This algorithm was implemented and we show several interesting results obtained from running the program.

II. Related work

Systems of multiple robots or agents cooperating in order to patrol in some designated area have been studied in various contexts using different approaches. Theoretical and empirical solutions were proposed in order to assure quality patrol [3], [6], [9]. The definition of quality depends on the context. Most studies concentrate on the frequency of visits throughout the designated area, therefore an efficient patrol guarantees high frequency of visits in each part of the area. In case the robots work in an adversarial environment, then an efficient patrol is one that deals efficiently with intruders.

This research was supported in part by ISF grant #1357/07 and #1685/07

Closely related to our research is the work of Paruchuri et. al., that consider the problem of placing security checkpoints [9], [8]. Similar to our assumptions, their agents work in an adversarial environment in which the adversary can exploit any predictable behavior of the patrolling agents. Their agents use policy randomization in order to maximize their rewards. However, the problem they describe, even for a single agent, is solved in exponential time, hence they provide *heuristic* algorithms for the single and multi agent case. In our work, we simplify the problem such that it is reasonable and implementable on one hand, yet we find *optimal* strategy for the robots in polynomial time. Paruchuri et. al. further study ([8]) the problem of checkpoint placement in case the adversarial behavior is unknown, and again provide heuristic algorithms for optimal strategy selection by the agents.

Theoretical work based on stochastic processes that is related to our work is the *cat and mouse* problem [5], also known as the *predator-prey* [7] or *pursuit evasion* [11]. In this problem, a cat is attempting to catch a mouse in a graph where both are mobile. The cat has no knowledge about the mouse's movement, therefore as far as the cat is concerned, the mouse travels similarly to a simple random walk on the graph. We, on the other hand, have worst case assumptions about the adversary. We consider a *robotic* model, in which the movement is correlated to the movement of a robot, with possible directionality of movement and possible cost of changing directions. Moreover, in our model the robots travel around a perimeter, rather than in a graph or an area.

Other theoretical work by Shieh and Calvert [10], based on computational geometry solutions, attempts to find optimal viewpoints for patrolling robots. They try to maximize the view of the robots in the area, show that the problem is $\mathcal{NP}$-Hard, and find approximation algorithms for the problem.

The first theoretical analysis of multi-robot patrol problem was given by Chevaleyre [3]. He introduces the notion of *idleness*, which is the duration each point in the patrolled area is not visited. In his work, he analyzes two types of multi-robot patrol schemes with respect to the idleness criteria: partitioning the area into subsections, each section is visited continuously by one robot, and the cyclic scheme in which a patrol path is provided along the entire area, and all robots visit all parts of the area, consecutively. He proves that in the latter approach, the frequency of visiting points in the area is considerably higher. Another survey by Almeida et. al. [2] offers an empirical comparison between different approaches towards patrolling with regards to the idleness criteria, and show show great advantage to the cycle based approach.

Elmaliach et. al. [6] offer new frequency optimization criteria for evaluating patrol algorithms. They provide an algorithm for multi-robot patrol that is proven to have optimal frequency as well as uniform frequency, i.e., each point in the area is visited with the same highest-possible frequency. Their work is based on creating one patrol cycle that visits all points in the area in minimal time, and the robots simply travel equidistant along this patrol path.

III. MODELS

In this section we provide basic definitions concerning the assumptions on the robots' behavior and coordination and the influence of these attributes on the patrol mission.

A. Robotic computational model

We consider a system consisting of k homogenous mobile robots, required to patrol around a closed area P. The robots operate in cycles, where each cycle consists of two stages.

1) **Compute**: Execute the given algorithm, resulting in a goal point p_G.
2) **Move**: Move towards the point p_G.

This model is synchronous, i.e. all robots execute each cycle simultaneously. We consider patrol in a circular path, which is similar to a one dimensional graph.

The path around P is divided into segments of length l, where l corresponds to the distance one robot travels and monitors the area in one cycle. Hence each robot travels through one segment per cycle while covering it (its velocity is 1 segment per one time cycle). This division into segments makes it possible to consider patrols in heterogenous terrains. In such areas, the difficulty of passing through terrains vary from one terrain to another, for example driving in muddy tracks vs. driving on a road. In addition, riding around corners requires a vehicle to slow down. Figure 1 demonstrates a transition from a given area to a discrete cycle. Throughout the paper, we denote the number of segments around the perimeter by N. Note that the distance between the robots is calculated with respect to the number of segments between them, i.e., the distance is in travel time. For example, if we say that the distance between R_1 and R_2 is 7, then there are 7 segments between them, and if R_1 remained still, then it would have taken R_2 7 time cycles to reach R_1.

At each cycle a robot that resides in segment i has three options as to where to go - segment $i-1$, segment $i+1$ or remain in segment i. We assume the robots are coordinated, i.e., all robots decide simultaneously to move in the same direction. We also require that the robots are initially placed uniformly around P with distance $d = N/k$ between every two consecutive robots. The motivation for these assumptions is shown in Lemma 3.

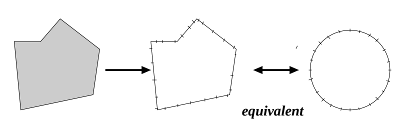

Fig. 1. An example for creating discrete segments from a circular path with the property that the robots travel through one segment per cycle.

Definition: Let s_i be a discrete segment of a perimeter P which is patrolled by one robot or more. Then the *Probability of Penetration Detection* in s_i, ppd_i, is the probability that a penetrator going through s_i during t time units is detected by some robot going through that segment during that period of time. In other words, ppd is the probability that a patrol path of some robot will pass through segment s_i during the time that a penetrator is going through that segment. We use the general acronym ppd when referring to the general term of probability of penetration detection (without reference to a certain segment).

B. Robotic movement model

The execution of the patrol differs from one model to the other in the Compute step. As mentioned previously, we consider three different patrol models, based on movement abilities of the robots.

1) Bidirectional Movement Patrol ($\mathcal{BMP}$)
2) Directional Zero-Cost Patrol ($\mathcal{DZCP}$)
3) Directional Costly-Turn Patrol ($\mathcal{DCP}$)

The $\mathcal{BMP}$ patrol is intended for robots whose movement pattern is similar to movement on train tracks or a camera going back and forth along a fixed course. Here, the robots have no movement directionality in the sense that switching directions — right to left and vice versa — does not require physically changing their direction (turning around).

In the other two models the robots' movement is directed, and turning around is a special operation that might have an attached cost in time. The $\mathcal{DZCP}$ patrol is used for robots which have directionality of movement, but turning around does not consume extra time. The $\mathcal{DCP}$ patrol model is a more realistic version of the $\mathcal{DZCP}$ model, where if the robot turns around, it remains in its current position, i.e., switching direction costs the system extra time. An example for this kind of robots are the differential drive robots commonly used in research labs. For simplicity reasons, we assume that turning around costs one time cycle.

C. Adversarial model

We assume the system works with the existence of an adversary that controls the behavior of the penetrators. We assume the adversary is strong in the sense that it has full knowledge of the system. Specifically, the adversary has the following information, formally known as the *patrol scheme*:

1) Number of robots, the distance between them and their current position.
2) The movement model of the robots and any characterization of their movement.

This information can be learned by the adversary by observing the behavior of the robots for sufficiently long enough time. Note that in security applications, such strong adversary exists. In other applications, the adversary models the behavior of the system in the "worst case scenario" from the patrolling robots point of view.

The adversary, having all the information it obtained, has to decide at time 0 through what segment is wishes to penetrate. Therefore it will choose to pass where it will less likely be detected by the robots.

Note that we assume the adversary tries to penetrate *once* through some segment.Also, the robots are responsible only for *detecting* penetrations and *not* handling the penetration (which requires task-allocation methods). Therefore the case in which the adversary issues multiple penetrations is similar to handling a single penetration, as the robots detect, report and continue their monitoring through the rest of the path, according to their algorithm.

D. Problem definition

Since we assume the existence of a strong adversarial model, we assume the adversary will choose to penetrate through the weakest spot in the cycle. Therefore we wish to find a patrol algorithm that maximizes the probability of penetration detection in that weakest spot. This algorithm is characterized by a probability value p, according to which the robots switch their direction through their patrol. Note that p could be 1, and then the algorithm is deterministic. First, we define how p characterizes the movement of the robots in the different movement models. We then provide a formal definition of the generic problem.

Assume a robot is currently located in segment i. In the $\mathcal{BMP}$ model, it moves one step to the right (segment $i+1$) with probability p and one step to the left (segment $i-1$) with probability $q = 1-p$. This model is similar to a random walk. See Figure 2a for an illustration. In both the $\mathcal{DZCP}$ and $\mathcal{DCP}$ models, we assume directionality of movement, hence the robot continues its movement in its current direction with probability p, and turns around (rewinds) with probability $q = 1 - p$. Therefore in the $\mathcal{DZCP}$ model, if the robot is facing segment $i+1$, then it has probability p of going straight to it and probability $1-p$ for turning around and reaching cell $i-1$. Similarly, if it faces segment $i-1$, then it has probability p or reaching $i-1$ and probability $1-p$ of reaching segment $i+1$. The $\mathcal{DCP}$ model is similar, only that if the robot turns around it remains in segment i. See Figures 2b. and 2c. for illustration of the $\mathcal{DZCP}$ and $\mathcal{DCP}$ models, respectively.

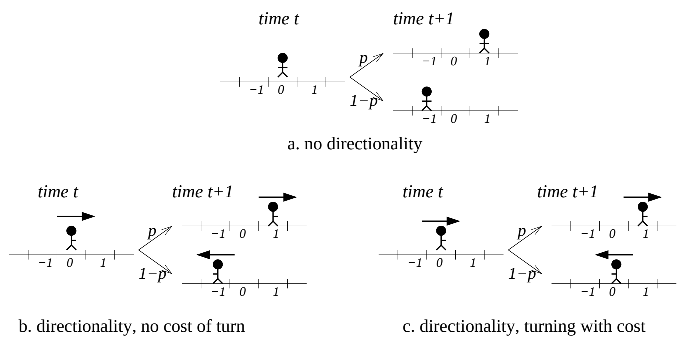

Fig. 2. Illustration of p's characterization of the three models of movement.

Border penetration detection ($\mathcal{BPD}$) problem: Given a circular fence (perimeter) of length l divided into N segments, and k robots uniformly distributed around this perimeter with distance $d = N/k$ (in time) between every two consecutive robots. Assume that it takes t time units for the adversary to penetrate. Let $f_i(p) = \mathsf{ppd}_i$, $1 \le i \le d-1$ (describing ppd_i as a function of p). Find the optimal p, p_{opt}, such that the minimal ppd throughout the perimeter is maximized. Formally,

$$p_{opt} = \max \min_{1 \le i \le d-1,\ 0 \le p \le 1} f_i(p)$$

IV. Preliminaries

Following, we justify our motivation for considering models in which robots are placed uniformly around the perimeter and are coordinated in the sense that they all move together in the same direction and switch directions simultaneously.

Recall that we assume the number of time units it takes the penetrator to enter is t, and that the time between every two consecutive robots around the perimeter is d. Therefore we consider t values between the boundaries $\lfloor\frac{d}{2}\rfloor \leq t \leq d-2$. The reason for this is that in case $t < \lfloor\frac{d}{2}\rfloor$, then there is at least one segment s_i with $\mathsf{ppd}_i = 0$, therefore a strong adversary will *always* manage to penetrate successfully regardless of the actions taken by the patrolling robots. On the other hand, if $t > d-2$ then *all* segments s_i can have $\mathsf{ppd}_i = 1$ simply by using a deterministic algorithm.

In order to find the probability of penetration detection in some segment s_i, ppd_i, we need to find the probability that s_i is visited during t time units. ppd_i is determined only by the *first* visit to s_i, since once the intruder is detected then the detection mission was successful. Therefore ppd_i is actually the probability that a segment will be visited *at least once* during t time units. Denote the probability of detecting a penetrator by robot R_a in segment s_j after t time units by $\mathsf{ppd}_j(R_a)$. Note that ppd_i is the sum of probabilities that $R_1, \ldots, R_k$ will visit that segment during this time, i.e., $\mathsf{ppd}_i = \sum_{j=1}^{k} \mathsf{ppd}_i(R_j)$. Also, the value of the ppd is calculated *regardless of the actions of the adversary*.

Lemma 1: For a given p, the function $\mathsf{ppd}_i(R_a) : \mathbb{N} \Rightarrow [0,1]$ for constant t and R_a residing in segment s_0 is a monotonic decreasing function, i.e., as the distance between a robot and a segment increases, the probability of arriving in it during t time units decreases.

Proof: We need to show that for all i, $i \in \mathbb{N}$, $\mathsf{ppd}_i(R_a) \geq \mathsf{ppd}_{i+1}(R_a)$. The movement of the robots is coherent, i.e., in order to move from segment i to segment $i+2$, it has to move through segment $i+1$. Therefore the probability of arriving at segment i given that we have arrived at segment $i+1$ is 1, i.e., $\mathsf{ppd}_{i|i+1}(R_a) = 1$. By conditional probability law, if $\mathsf{ppd}_{i+1}(R_a) > 0$ then

$$\mathsf{ppd}_{i|i+1}(R_a) = \frac{\mathsf{ppd}_{i+1\cap i}(R_a) \cdot \mathsf{ppd}_i(R_a)}{\mathsf{ppd}_{i+1}(R_a)} = 1$$

$$\Rightarrow \mathsf{ppd}_{i+1}(R_a) = \mathsf{ppd}_{i+1\cap i}(R_a) \cdot \mathsf{ppd}_i(R_a) \leq \mathsf{ppd}_i(R_a)$$

If $\mathsf{ppd}_{i+1}(R_a) = 0$, then since ppd's value can not be lower than 0, then necessarily $\mathsf{ppd}_i(R_a) \geq \mathsf{ppd}_{i+1}(R_a)$. ■

Lemma 2: As the distance between two consecutive robots along a cyclic patrol path is smaller, the ppd in each segment is higher and vice versa.

Proof: Consider a sequence S_1 of segments $s_1, \ldots, s_w$ between two adjacent robots, R_a and R_b, where s_1 is adjacent to the current location of R_a and s_w is adjacent to the current location of R_b. Let S_2 be a similar sequence, but with $w-1$ segments, i.e., the distance between R_a and R_b decreases by one segment. Assume that other robots are in distance greater than or equal to $w-1$ away from R_a and R_b, and that $w-1 < t$. Since a robot may influence the ppd in segments that are up to distance t from it (as it has probability 0 of arriving at any segment with greater distance within t time units), the ppd in these sequences is influenced only by possible visits of R_a and R_b.

Denote the probability of penetration detection in segment $s_i \in S_j$ by ppd_i^k, $1 \leq i \leq w$, $j \in \{1,2\}$, and the probability that the penetrator is detected by robot R_l by $\mathsf{ppd}_i^k(R_l)$. Therefore, for any segment $s_i \in S_j$, $\mathsf{ppd}_i^j = \mathsf{ppd}_i^j(R_a) + \mathsf{ppd}_i^j(R_b)$. Note that either $\mathsf{ppd}_i^j(R_a)$, $\mathsf{ppd}_i^j(R_b)$ or both can be equal to 0. It is required to show that $\mathsf{ppd}_i^2 \geq \mathsf{ppd}_i^1$, for all $1 \leq i \leq w$, and for at least one segment s_m, $\mathsf{ppd}_m^2 > \mathsf{ppd}_m^1$.

First, for s_w, $\mathsf{ppd}_w^2 = 1$ as R_j is presently on segment s_w in S_2. Therefore $\mathsf{ppd}_w^2 = 1 \geq \mathsf{ppd}_w^1$.

For every other segment s_i, $\mathsf{ppd}_i^j(R_a)$ remains the same (there is no change in its relative location), hence we need to examine the change in $\mathsf{ppd}_i^j(R_b)$. From Lemma 1 we know that $\mathsf{ppd}_i^j(R_b)$ is a monotonic decreasing function. Therefore for each i, $\mathsf{ppd}_i^2(R_b) \geq \mathsf{ppd}_i^1(R_b)$. We need to show that for at least one segment $\mathsf{ppd}_i^2(R_b) > \mathsf{ppd}_i^1(R_b)$. A robot may influence the ppd to both of his sides - segments located left and right to its current position. Denote the number of influenced segments to its right by s (s may be equal to 0). If $s > 0$, then $\mathsf{ppd}_{w-s+1}^2(R_b) > \mathsf{ppd}_{w-s}^1(R_b)$. In other words, R_b has probability 0 to reach the segment with distance $s+1$ from it in S_1, but in S_2 it is s segments away from it, therefore R_b has probability greater than 0 to reach it. If $s = 0$, then $\mathsf{ppd}_w^2 = 1 > \mathsf{ppd}_w^1$, as R_b lies exactly on segment s_w in S_2, and $\mathsf{ppd}_k^1(R_b) = 0$. ■

Lemma 3: A team of k mobile robots engaged in a patrol mission maximizes minimal ppd if the following conditions are satisfied. **a.** The time distance between every two consecutive robots is equal **b.** The robots are coordinated.

Note that condition **b** means that all robots move together in the same direction, i.e., if they change direction, then all k robots change their direction simultaneously.

Proof: Following Lemma 2, it is sufficient to show that the combination of conditions **a** and **b** yield the minimal distance between two consecutive robots along the cyclic path. Since we have N segments and k robots, there are $\binom{N}{k}$ possibilities of initial placing of robots along the cycle (robots are homogenous, so this is regardless of their order). If the robots are placed uniformly along the cycle, then the time distance between each pair of consecutive robots is N/k. This is the minimal value that can be reached. Therefore, clearly, condition $\mathbf{a}$ guarantees this minimality.

If the robots are not coordinated, then it is possible that for two consecutive robots along the cycle, R_i and R_{i+1}, to move in opposite directions. Therefore the distance between them increases from $\frac{N}{k}$ to $\frac{N}{k}+2$, and by Lemma 2 the ppd in the segments between them is smaller. If R_i and R_{i+1} move towards one another, then the distance between them is $\frac{N}{k}-2$ and the ppd in the segments between them becomes higher. On the other hand, there exists some pair R_j and R_{j+1} where the distance between them increases, as the total sum of distances between consecutive robots remains N, hence the minimal ppd around the cycle becomes smaller.

Therefore the only way of achieving minimal distance (maximal ppd) is by assuring that condition $\mathbf{a}$ is satisfied, and maintaining it is achieved by satisfying condition $\mathbf{b}$. ■

Following Lemma 3, we assume that the robots are coordinated, and placed uniformly along the patrol path.

V. Algorithm for finding optimal p

After establishing the preliminary assumptions, we wish to find a solution to the $\mathcal{BPD}$ problem. The solution to the problem is twofold. First, it is necessary to find equations representing the detection probability in each segment along the patrol path. At the second stage, the equations are manipulated in order to find the required probability (here the maximin point). In this section we describe a polynomial time algorithm for solving the $\mathcal{BPD}$ problem *optimally*.

A. Finding the equations

In order to analyze the ppd achieved by a patrol algorithm, it is enough to consider only one segment of the path that lies between two consecutive robots, without loss of generality R_1 and R_2. This segment has two extreme robots, and is of length d. We use a Markov chain in order to model the states the system can be in. We describe herein the modeling under the $\mathcal{DCP}$ movement model (the cases of $\mathcal{BMP}$ and $\mathcal{DZCP}$ are similar, and are illustrated in Figure 3).

In order to calculate the probability of detection in each segment along t time cycles, we use the graphic model G illustrated in Figure 3. For each segment s_i in the original path, $1 \le i \le d-1$, we create two states in G: One for going in clockwise direction (s_i^{cw}), and the other going in counterclockwise direction (s_i^{cc}). As mentioned previously, if R_1 or R_2 reach one of the segments s_i within t time units, then the intruder is discovered, i.e., it does not matter if the segment is visited more than once during these t time units. Therefore we consider only the probability of the first visit to each segment, and this is done by defining the states s_0 and s_0' as absorbing states. The edges of G are as follows. There exists one outgoing edge from s_i^{cw} to s_i^{cc} with probability q for turning around, and one outgoing edge to s_{i-1}^{cw} with probability p for continuing straightforward. Similarly, there exists one outgoing edge from s_i^{cc} to s_i^{cw} with probability q for turning around, and one outgoing edge to s_{i+1}^{cc} with probability p for continuing straightforward.

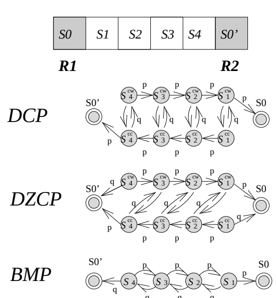

Fig. 3. Converting the initial segments and robot locations to a graphical model for the three possible robotic models: $\mathcal{DZCP}$, $\mathcal{DCP}$ and $\mathcal{BMP}$.

The straightforward way of finding the probability of arriving at an absorbing vertex is using a stochastic matrix M, which represents the probability of transition between states. In order to find the probability of absorption after t cycles starting from each state s_i, $1 \le i \le d-1$, M^t should be computed. However, as t reaches $d = \frac{N}{k}$, it leads to a computational complexity exponential in the input size.

Therefore we use the following dynamic-programming inspired rule in order to find the *optimal* solution, yet in polynomial time. We determine the probability of reaching a certain state in time r by the sum of probabilities of reaching s_i from any other state s_j multiplied by the probability of being in state s_j at time $r-1$. Hence in order to compute the probability of reaching absorption state in t time cycles starting from state s_{init}, we initialize s_{init} with the value 1 at $r = 0$, compute the values for $r = 1, \ldots, t$, and extract the probability at the absorption state, s_{abs}. See Procedure FindFunc(d,t) for a detailed description of the method.

Procedure FindFunc(d,t)
For each $s_{init} = s_i \in \{s_1, \ldots, s_{d-1}\}$ do:
Create matrix M of size $(2d+2) \times (t+1)$, initialized with 0s.
Set $M_0(s_{init}) \leftarrow 1$.
Complete M gradually using the following rules.

1) **For** each entry $M_r(s_i^{cw})$ set value to $p \cdot M_{r-1}(s_{i+1}^{cw}) + q \cdot M_{r-1}(s_i^{cc})$.
2) **For** each entry $M_r(s_i^{cc})$ set value to $p \cdot M_{r-1}(s_{i-1}^{cc}) + q \cdot M_{r-1}(s_i^{cw})$.
3) **For** absorbing states, set entry $M_r(s_{abs}) = M_{r-1}(s_{abs}) + p \cdot [M_{r-1}(s_1^{cw}) + M_{r-1}(s_d^{cc})]$.

Report row t of M.

Fig. 4. Description of FindFunc algorithm.

The time complexity of Procedure FindFunc is $d \cdot (2d+2) \cdot (t+1)$. Since t is bounded by $d-1$ and $d = N/k$, then the complexity is $\mathcal{O}((\frac{N}{k})^3)$.

B. Finding the maximin point

After establishing the $d-1$ equations representing the probability of detection in each segment, it is left to find the value p that maximized the minimal possible value in each segment, where $p \in [0,1]$. Denote these equations by $f_i(p)$, $1 \le i \le d-1$. The maximal minimal value is the maximal value that lies inside the intersection of all integrals of f_i.

Observing the problem geometrically, consider a vertical sweep line that sweeps the section $[0,1]$ and intersects with all $d-1$ curves. It seeks the point p in which the minimal intersection point between the sweep line and the curves, $f^*(p)$, is maximal. This p is the maximin point. Since the segment $[0,1]$ and the functions $f_1, \ldots, f_{d-1}$ are continuous, this sweep line solution cannot be implemented. We observe that a maximin point is actually the maximal point that lies inside the integral of all curves. We prove in the following lemma that this point is either an intersection point of two curves, or a local maxima of one curve (see Figure 5). See Figure 6 for the formal description of Algorithm FindP.

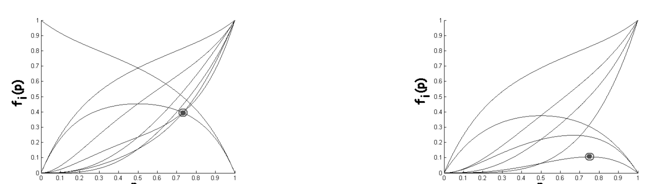

Fig. 5. An illustration of two possible maximin points. On the left, the point is created by the intersection of two curves, and on in the right it is the local maxima of the lowest curve.

Following, we prove that Algorithm FindP finds the point p such that maximin property is satisfied.

Lemma 4: A point p yields a maximin value $f^*(p)$ if the following two properties are satisfied.
a. $f^*(p) \leq f_i(p)\ \forall 1 \leq i \leq d-1$.
b. One of the two following conditions holds: $f^*(p)$ is an intersection of two curves (or more), $f_i(p)$ and $f_j(p)$ *or* a local maxima of curve $f_k(p)$.

Proof: Property **a.** is derived from the definition of a maximin point. Therefore we are looking for the maximal point that satisfies property a. It is left to show that this point, $f^*(p)$, is obtained by either an intersection of two or more curves or is a local maxima. Clearly, a maximal point of an integral is found on the border of the integral (the curve itself). The area which is in the intersections of all curves lies beneath parts of curves, $f_{i_1}, \ldots, f_{i_m}$, such that f_{i_j} is the minimal curve in the section $[l^j, r^j]$ and $\bigcup_{j=1}^{m}[l^j, r^j] = [0,1]$. By finding the maximal point in each section $f^j_{max} = \max\{f(x), x \in [l^j, r^j]\}$, and choosing the maximal between them, i.e., $\max\{f^j_{max}, 1 \leq j \leq m\}$, we obtain $f^*(p)$. In each section $[l^j, r^j]$ the maximal point can be either inside the section or on the borders of the section. The former case is exactly a local maxima of f_{i_j}. The latter is the intersection point of two curves $f_{i_{j-1}}, f_{i_j}$ or $f_{i_j}, f_{i_{j+1}}$. ∎

Lemma 5: There exists a point p yielding a maximin value $f^*(p) > 0$.

Proof: In order to prove the lemma, we need to show that the intersection of all integrals $f_1, \ldots, f_{d-1}$ in the x section $[0,1]$, and the y section $(0,1]$ is not empty. It suffices to show that for every f_i, $f_i(x) > 0, 0 < x < 1$.

Each function $f_i, 1 \leq i \leq d-1$ represents the ppd in a segment s_i between two robots. From our requirement that $t \geq \lfloor \frac{d}{2} \rfloor$, it follows that in all models we consider, for $0 < p < 1$ the ppd $\neq 0$. Note that if $p = 0$ or $p = 1$, then ppd is either 0 or 1, but this does not contradict the fact that we have a point guaranteeing $f^*(p) > 0$. ∎

Algorithm FindP finds this point by scanning all possible points satisfying the conditions given in Lemma 4, and reporting the x-value (corresponding to the p value) that its y-value is dominated by all f_i. The input to the procedure is a vector of functions f_i, $1 \leq i \leq d-1$ and the value t. The time complexity of Algorithm FindP is the complexity of Procedure FindFunc, $\mathcal{O}((\frac{N}{k})^3)$ plus $\mathcal{O}(d^3) = \mathcal{O}((\frac{N}{k})^3)$ (the algorithm itself), i.e., together $\mathcal{O}((\frac{N}{k})^3)$.

Algorithm FindP(d,t)
1) $F \leftarrow$ Procedure FindFunc(d,t).
2) Set $p_{opt} \leftarrow 0$.
3) **For** $F_{pivot} \leftarrow F_{1,\ldots,d-1}$ **do**:
   a) Compute local maxima $(p_{max}, F_{pivot}(p_{max}))$ of F_{pivot} in the range $(0,1)$.
   b) **For** each F_i, $1 \leq i \leq d-1$, compute intersection point p_i of F_i and F_{pivot} in the range $(0,1)$.
   c) If $F_{pivot}(p_i) > F_{pivot}(p_{max})$ and $F_{pivot}(p_i) \leq F_k(p_i) \forall k$, then $p_{opt} \leftarrow p_i$.
   d) If $F_{pivot}(p_{max}) > F_{pivot}(p_i)$ and $F_{pivot}(p_i) \leq F_k(p_i) \forall k$, then set $p_{opt} \leftarrow p_{max}$.
4) Return $(p_{max}, F_{pivot}(p_{max}))$.

Fig. 6. Description of FindP algorithm.

Theorem 6: Algorithm FindP(F,t) finds the point p yielding maximin value of ppd.

Proof: Algorithm FindP checks both all intersection points between pair of curves, and points of local maxima of curves. It then checks the dominance of these points,and picks maximal between them. Therefore, if such a point is found, by Lemma 4, this point is exactly the maximin point. Moreover, by Lemma 5 this point exists. ∎

VI. RESULTS

We have fully implemented Algorithm FindP in order to find the optimal maximin p for pairs of d's and t's. In the following section, we describe a few interesting results that we got when running the program. Recall that when running a deterministic patrol algorithm in all scenarios we handle, the minimal ppd is 0. If directionality is considered, we assume the robots are initially heading to the clockwise direction. We first show results of the $\mathcal{DCP}$ model, then an example of the difference between the three models.

First of all, we have seen that the minimal ppd achieved after running FindP was always more than 0. As $t/d \rightarrow 1$, i.e., t increases, then the value of the maximin ppd increases, and vice versa, i.e., as $t/d \rightarrow 1/2$, then the value of the maximin ppd decreases. This can be seen clearly in Figure 7. In this case, we have fixed the value of t to 8 and checked the maximin ppd for $9 \leq d \leq 15$. When t/d is close to 1 ($d = 9, t = 8$) the maximin ppd $= 0.423$, and the value decreases to 0.05 when t/d is close to $1/2$ ($d = 15, t = 8$). Similar results are seen if we fix the value of d and check for different values of t.

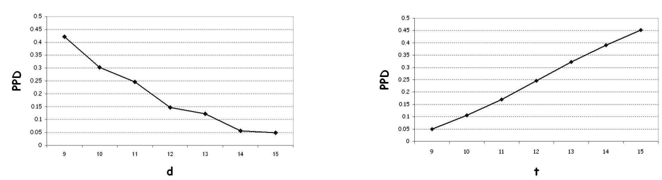

Fig. 7. On the left, results of maximin ppd for fixed $t = 8$ and different values of d: the possible maximin ppd decreases as d increases. On the right, results of maximin ppd for fixed $d = 16$ and different values of t: the possible maximin ppd increases as t increases.

In Figure 8, we bring the values of the ppd in all 16 segments, for all different possible values of t ($9 \leq d \leq 15$). It is seen clearly, that the value of ppd usually decreases as the distance from the left robot increases, until it reaches the segment with maximin ppd, then the value rises again until reaching the current location of the robot to the right. The reason lies in the fact that the segments to the left of the segment with the maximin ppd are influenced mostly by the robot on the left, and the segments to the right of that point are mostly influenced by the robot to the right. Since the p's yielding the maximin point in this example have value of greater than 0.8 for all t's, the segment having the maximin value is to the right of the midpoint.

Last, we bring an illustration of the difference between the values of the ppd obtained by all three models: $\mathcal{DZCP}$, $\mathcal{DCP}$ and $\mathcal{BMP}$ in all 16 segments, in case $t = 12$. It is clearly noticeable that the $\mathcal{DCP}$ model yields less or equal values of ppd compared to $\mathcal{DZCP}$ model throughout

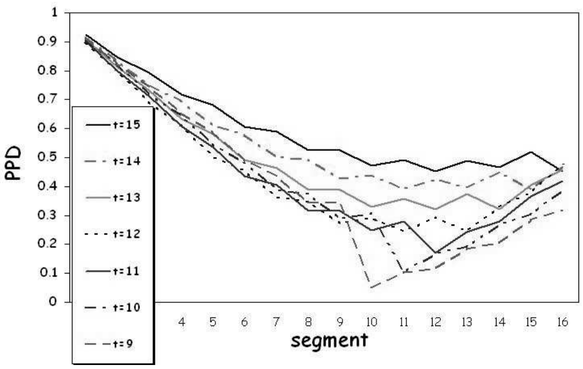

Fig. 8. ppd values in all 16 segments for all t values (9 to 15)

the segments. The reason is because when turning around, in the $\mathcal{DCP}$ model, the operation costs an extra cycle, therefore the probability of arriving at a segment decreases, compared to the case in which turning around is not costly. Another interesting phenomena is that the ppd values of the $\mathcal{BMP}$ are considerably higher (and close to 1) than the values obtained by other models for segments closer to the location of the righthand side robot. The value then decreases dramatically around the value of t and then increases back again. Recall that here there is no directionality of movement, therefore the probability of going right is 0.707 and going left is $1 - 0.707 = 0.293$, which explains this phenomena.

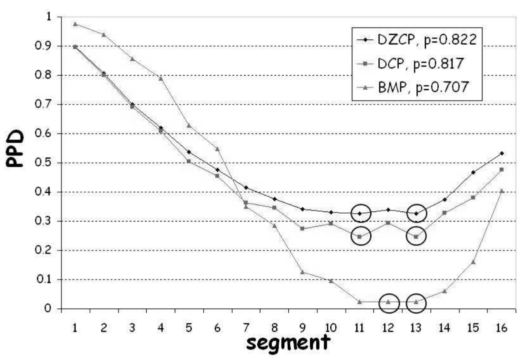

Fig. 9. Results of maximin ppd values for $d = 16$ and $t = 12$ for all three models: $\mathcal{DZCP}$, $\mathcal{DCP}$ and $\mathcal{BMP}$. The maximin ppd values are circled.

VII. Working under other adversarial models

Until now we have discussed the case in which the adversary has full knowledge of the robots' policy, and decides to penetrate through the least protected segment. If the adversary has partial knowledge, finding p_{opt} has to be done differently. This case is interesting, since we might obtain higher values of ppd considering weaker adversaries. The advantage of our method is that the first part of finding the equations is similar for all methods. Finding the optimal p is done by replacing the second stage with a more suitable function to the given scenario. This scenario depends, for instance, on the knowledge of the adversary and/or its preferences and on the preferences of the robots. For example, if the adversary has no knowledge on the robots' behavior, then it might choose to enter through any point currently unoccupied by a robot with uniform probability. In this case the p we would choose would correspond to the p that maximizes the expected ppd over all segments. One might want to minimize the standard deviation of the ppd through all segments, or create some weighted function taking into consideration multiple requirements.

VIII. Conclusions and future work

This paper discusses the problem of multi-robot perimeter patrol around a closed area in adversarial settings. We assume a strong adversarial model, in which the adversary knows the location of the robots and the patrol scheme. We show that in this case, if the time it takes the adversary to penetrate is less than the minimal duration between two visits of some robot, then it can penetrate with probability 1 even through an optimal *deterministic* patrol algorithm. Therefore we consider a non-deterministic patrol algorithm, with probability p characterizing the robots' movement. We assume *strong* adversary, that has full knowledge of the patrol scheme. It will therefore decide to penetrate through the point in which it has minimal probability of being detected. We offer a polynomial-time algorithm for finding the probability p of the robots, such that the minimal probability of penetration detection is maximized. We have implemented this algorithm, and showed that this probability is always greater than 0 for various values of penetration time of the adversary we consider and for three robotic models we propose.

There are various points we wish to address as future work. First, we would like to find a solution for the continuous case, rather than the discrete model we consider here. We are interested in more realistic movement models, mainly ones with arbitrary turning time. We would also like to consider in depth other adversarial models, and also the case of unknown adversary similar to Bayesian games. Last, we would like to see how this algorithm can be adapted to patrol in other domains (area patrol, for example).

Acknowledgement: We thank Efrat Manisterski for her highly useful comments.

References

[1] M. Ahmadi and P. Stone. A multi-robot system for continuous area sweeping tasks. In *ICRA*, 2006.

[2] A. Almeida, G. Ramalho, H. Santana, P. Tedesco, T. Menezes, V. Corruble, and Y. Chevaleyr. Recent advances on multi-agent patrolling. *Lecture Notes in Computer Science*, 3171:474–483, 2004.

[3] Y. Chevaleyre. Theoretical analysis of the multi-agent patrolling problem. In *Proceedings of Intelligent Agent Technology (IAT)*, 2004.

[4] H. Choset. Coverage for robotics—a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31:113–126, 2001.

[5] D. Coppersmith, P. Doyle, P. Raghavan, and M. Snir. Random walks on weighted graphs and applications to on-line algorithms. *J. ACM*, 40(3), 1993.

[6] Y. Elmaliach, N. Agmon, and G. A. Kaminka. Multi-robot area patrol under frequency constraints. In *ICRA*, 2007.

[7] T. Haynes and S. Sen. Evolving behavioral strategies in predators and prey. In *IJCAI-95 Workshop on Adaptation and Learning in Multiagent Systems*, pages 32–37, 1995.

[8] P. Paruchuri, J. P. Pearce, M. Tambe, F. Ordonez, and S. Kraus. An efficient heuristic approach for security against multiple adversaries. In *AAMAS*, 2007.

[9] P. Paruchuri, M. Tambe, F. Ordonez, and S. Kraus. Security in multiagent systems by policy randomization. In *AAMAS*, 2007.

[10] J. S. Shieh and T. W. Calvert. View and route planning for patrol and exploring robots. *Advanced Robotics*, 6(4):399–430, 1992.

[11] R. Vidal, O. Shakernia, H. J. Kim, D. H. Shim, and S. Sastry. Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation. *Robotics and Automation, IEEE Transactions on*, 18(5):662–669, 2002.