On the Power and Limitations of Deception in Multi-Robot Adversarial Patrolling*

Noga Talmor and Noa Agmon
Department of Computer Science, Bar-Ilan University, Israel
nogatalmor@gmail.com, agmon@cs.biu.ac.il

Abstract

Multi-robot adversarial patrolling is a well studied problem, investigating how defenders can optimally use all given resources for maximizing the probability of detecting penetrations, that are controlled by an adversary. It is commonly assumed that the adversary in this problem is rational, thus uses the knowledge it has on the patrolling robots (namely, the number of robots, their location, characteristics and strategy) to optimize its own chances to penetrate successfully. In this paper we present a novel defending approach which manipulates the adversarial (possibly partial) knowledge on the patrolling robots, so that it will believe the robots have more power than they actually have. We describe two different ways of deceiving the adversary: *Window Deception*, in which it is assumed that the adversary has partial observability of the perimeter, and *Scarecrow Deception*, in which some of the patrolling robots only appear as real robots, though they have no ability to actually detect the adversary. We analyze the limitations of both models, and suggest a random-based approach for optimally deceiving the adversary that considers both the resources of the defenders, and the adversarial knowledge.

1 Introduction

The problem of defending a valuable asset from adversaries using patrolling robots is well studied. This problem is of great importance, as algorithms developed are used to protect airports, ports, natural reserves and train stations from criminals, thieves and terrorists [Sun *et al.*, 2011; Delle Fave *et al.*, 2015; Felemban, 2013].

There are many characteristics defining this problem, such as robots' movement and sensing abilities, communication between robots, and knowledge they have on the world and on the adversary. Researches studied the influence of each characteristic, and showed for every case how the adversary acts and what is the best defence strategy against this behavior [Machado *et al.*, 2002; Agmon *et al.*, 2008a; Paruchuri *et al.*, 2006; Elmaliach *et al.*, 2009]. The adversarial model, on the other hand, is usually based on its knowledge about patrolling robots. Specially, the adversary is commonly assumed to be rational, and acts in order to optimize its own utility given the knowledge it has on the patrolling robots.

In this research we study the problem of multi-robot *perimeter patrolling*, in which a rational adversary does not have perfect information about the patrolling robots. We examine how the information held by the adversary can be deliberately manipulated, and show how it is possible to deceive the adversary by pretending to have more power than the robots actually have, specifically: make the adversary think there are more patrolling robots, and by that decrease its chances of penetrating successfully.

We study two different mechanisms of conducting deception. The first is **Window Deception**, where the adversary can see only some portion of the entire perimeter (window). We show how along the window we can simulate patrols usually performed by more robots than we have. The second is **Scarecrow Deception**, where not all patrolling robots have the ability to detect intrusion (have no sensing abilities), and the adversary does not know they exist. In both cases, we examine what is the best way to guarantee that the adversary will not detect the fraud (if possible), and analyze the penetration detection rate achieved by using the deception mechanism. Results show that guaranteeing undiscovered deception without changing the characteristics of the robots, can be made only for random patrols. We have fully implemented the deception mechanisms, and following an empirical evaluation, report the tradeoff between deception and probability of penetration detection along the perimeter in several cases.

2 Related Work

The basic problem of multiagent or multi-robot patrol is long investigated, describing optimal placements and patrolling paths of robots in different scenarios [Elmaliach *et al.*, 2008; 2009; Agmon *et al.*, 2011; 2012]. The goal function in previous researches can be either maximizing the visiting frequency at each point in the target area, referred to as *frequency based patrolling*, or maximizing the rate of intrusion detection referred to as *adversarial patrolling*, which is the focus of this paper.

*This research was funded in part by ISF grant 1337/15.

In the problem of multi-robot adversarial patrolling, which is a part of a research topic of *security games* [Agmon *et al.*, 2008b; 2011; 2008a; Paruchuri *et al.*, 2006], a rational adversary with full knowledge of the defenders' patrol path and positions will penetrate at the weakest point of the defence - a point with minimum attendance probability. Therefore, an optimal patrol strategy is the one that maximizes the minimal penetration detection rate, achieved by calculating for every point its *ppd*- the probability that a penetration through that segment is revealed by some patrolling robot. It is proven that the best patrolling strategy when facing adversaries of different models is based on a symmetric placement of the robots along the perimeter, namely, having them maintain uniform spatio-temporal distance between them throughout the execution. If the distance between every two robots is less than penetration time of the adversary, then the robots should adopt a deterministic strategy (never turn), and otherwise they use random decision making. In those researches the adversary has full information, or partial knowledge on the entire perimeter.

An adversary having full knowledge about the defender only in some portion of the segments was studied by [Zhang and Luo, 2014; An *et al.*, 2013; 2012]. There, the adversary chooses targets he wishes to observe because learning the defenders' behavior in the entire perimeter is either too expensive or too dangerous. The defender is aware of this adversarial behavior and applies his best response. Furthermore, the adversary can also update his belief model about the defender based on his observations. In both researches, the adversary has fixed observation duration, and if the defender knows this time he can use it and maximize his detection probability. In our research the adversary does not have the ability to choose his surveillance area, and this is the base for our Window Deception model.

Another way to model the imperfect knowledge of the adversary is to assume that his sensing ability is not perfect. In [Yin *et al.*, 2011], nature-created noise in both execution and observation of adversary is presented. They developed RECON, an algorithm based on an assumption that the noise nature chooses is the one maximally reduces defenders' profit. In [Agmon *et al.*, 2011] the case of imperfect sensing of the defending robots is discussed and mathematical analysis of velocity uncertainties is suggested when patrolling along a open polygon. In all these researches, the defender does not try to change the adversarys' beliefs about his strategy or his ability, as we do in this research.

3 Window Deception Model

In this research we study a group of k homogeneous robots defending a perimeter of some closed polygon P from penetrations. We divide P into N identical time segments $S = \{s_i, 0 \leq i \leq N-1\}$, i.e., each robot passes through one segment s_i per time cycle while monitoring it. In this linear environment, at each time step a robot has two *movement-options*: *same-direction* with probability p, or *change-direction* with probability $1-p$, which takes τ time units. We say the robots are moving to the right side when they move counterclockwise, and to the left side when they move clockwise. The robots cannot jointly occupy the same segment, thus they do not change their relative order along the perimeter during the patrol.

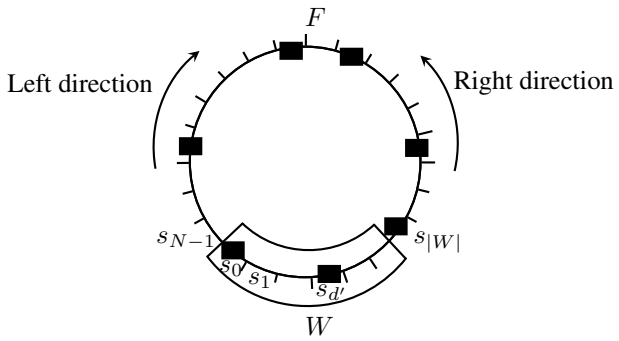


Figure 1: Perimeter including window and fence placements.

It takes the adversary $t \geq 1$ time units to penetrate the perimeter, and during this time it may be observed by the patrolling robots. We denote by ppd_i the probability that an adversary passing through segment s_i will be detected by some robot. Following the optimality proofs presented in [Agmon *et al.*, 2008b], we refer to a patrolling algorithm that maintains uniform spatio-temporal distance between every two consecutive robots throughout the execution such that if they decide to turn (with probability p) they do so simultaneously, as *basic patrol*. We refer to a basic patrol with $p = 1$ (deterministic, and never turning) as a *perfect patrol*.

The first deception model we study is the *Window Deception model*. Here, the adversary does not see the entire perimeter, but only a part of S called *window*, denoted by W (of length $|W|$), and will penetrate through it. The rest of perimeter, which is unseen to the adversary, is referred to as *fence*, denoted by F (of length $|F|$), namely $F = S \backslash W$. We define the farthest left segments in the window to be s_0, the successive window segment to be s_1, and so on (see Figure 1). We would like to preserve a perfect deception inside the window so that the adversary will believe that this is the true behavior also throughout the fence, and would not have incentive to try penetrating through segments he can't continuously observe, thus we cannot use open polygon (fence) patrolling algorithms along W (e.g., [Elmaliach *et al.*, 2008]).

Our goal is to execute a patrol algorithm by the robots along W so that the adversary will believe indefinitely that there are $k' > k$ robots patrolling around the perimeter. In this case we will say that the patrol algorithm *achieves a deception*. Namely, there are no anomalies in the patrol execution along W, which will cause the adversary to realize that there are less than k' robots patrolling around the perimeter.

The adversary observes W and expects to see a basic patrol, i.e., that all robots are fully coordinated with uniform spatio-temporal distance between every two robots, and having same momentary direction and speed. Since the number of robots determines the distance between them along the closed perimeter of N segments, the way of achieving deception is by having the robots maintain a uniform distance of some $d' = \frac{|W|}{k^W}$ between every two consecutive robots along W (k^W being the number of robots observed at one glance

to W), instead of $d = \frac{N}{k}$ (which reflects the true number of robots). Therefore, the deception algorithm allocates k^W robots to patrol W, referred to as window robots, and the rest of $k^F = k - k^W$ are used to preserve the deception and defend F, referred to as fence robots. Note that each robot is *not* limited to patrol inside a certain segment or sequence of segments, but k^W and k^F are fixed throughout the execution.

3.1 Window Deception - Deterministic Patrol

Recall that a perfect patrol is a deterministic basic patrol with $p = 1$. Clearly, if $d \leq t$, then the perfect patrol maximizes the probability of adversarial detection, guaranteeing that $ppd_i = 1$ in every segment $s_i \in S$, i.e., every intrusion is detected. We therefore begin our window deception research by trying to perform deterministic patrols along W with $d' \leq t$. We assume, without loss of generality, that the window robots move to the right (counterclockwise).

Perfect patrol

Unfortunately, for perfect patrols we have the following negative result:

Theorem 3.1. *It is impossible to simulate perfect patrol of $k' > k$ robots along W for infinite time if the robots travel in uniform velocity along P.* [1]

Proof sketch. Proving this theorem is based on examining the behavior of robots passing through F: every time the rightmost window robot reaches $s_{|W|-1}$ there should be some fence robot waiting at s_{N-1} ready to enter W, otherwise the deception is revealed at the next step. As the distance between robots in W is d', once every d' time units there should be a fence robot waiting as described. Consider the first time a robot leaves W. It will take it $|F|$ time units to enter W again, duration where only k^F robots can preserve the deception and enter W when needed. During $|F|$ time units $\frac{|F|}{d'} = \frac{N}{d'} - \frac{|W|}{d'} \geq k - k^W = k^F$ robots leave W. We conclude a robot leaving W does not reach its expected position on time so the deception is revealed.

Changing Velocity

Perfect patrol by $k' > k$ robots cannot be simulated along W since the robots exiting W cannot pass through F fast enough to enter W again on time. Therefore, we now examine a way to handle this restriction, by assuming that the robots have the ability to travel in velocity v_{max}, faster than the constant velocity of one segment per time unit.

Using this new movement ability, the robots travelling through F can be ready to enter W only $\frac{|F|}{v_{max}}$ time units after leaving W (instead of $|F|$). We get that it is possible to simulate only $k' \leq \frac{Nv_{max}}{|F|+|W|v_{max}}k$ robots patrolling along W. This bound is monotonically increasing as v_{max} increases, meaning the faster the robots can move, the bigger k' value we can simulate.

[1] We omit some proofs due to lack of space. If accepted, the proofs will be added in an 8-page version of the paper.

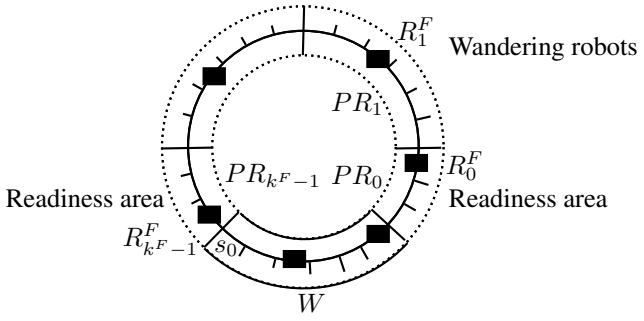


Figure 2: Readiness areas settings (adjacent to window) and two patrol regions. In this case $d' = 3, RA_c = 1$.

3.2 Window Deception - Random Patrol

We have shown that in perfect patrol the only way to simulate $k' > k$ robots patrolling W and maintaining the deception is by changing robots' movement velocity, meaning making changes to the robots' model. In this section we wish to demonstrate the power of random behavior, which allows us to implement deception based only on the patrol algorithm (without changing the robots' model). Recall that the robots continue straight with probability p at each time step (and turn with probability $1 - p$). A patrol algorithm with $p = \text{argmax}\{\min ppd_i\}$, i.e., one that maximizes the minimal ppd along the perimeter [Agmon *et al.*, 2011], is referred herein as maxmin-random patrol (mrand, in short).

We wish to perform infinite time deception that can handle every set of movement-options chosen randomly, i.e., for $p < 1$. In the very unlikely case that all chosen movement-options are same-direction option, Theorem 3.1 establishes that no algorithm can preserve the deception indefinitely. To avoid this case, we describe in the following section the basic requirements from a mrand patrol execution.

Enabling Deception in Random Patrol - Basic Assumptions

When executing a mrand patrol, movement-options are chosen based on Binomial distribution. A patrol algorithm that simulates a mrand patrol should satisfy the *distribution demand*: the algorithm must have the same mean value of movement-options distribution as a mrand patrol along W, so it seems as a mrand patrol to the adversary.

As stated, no algorithm can handle infinite same-direction movement-options sequentially. We therefore define a *continuous demand*: an adversary observing the algorithm can see movement in the same direction that involves insertion of up to RA_c robots one after the other into W, where RA_c is chosen by the defender (determining RA_c's value is established later in this section), but the algorithm cannot handle more than $RA_c \times d$ consecutive same-direction movement-options.

To fulfill the continuous demand we place RA_c robots at each side of W (outside of W) with distance d' from one another and from all window robots, and they perform exactly like the window robots. Each sequence of segments where those robots patrol, $\{s_{|W|}, ..., s_{|W|+d'RA_c-1}\}$ and $\{s_{N-d'RA_c}, ..., s_{N-1}\}$, is called *readiness area*. We refer

to fence robots not populating readiness areas as *wandering robots*. See illustration in Figure 2.

The patrolling robots can leave the readiness areas, so the wandering robots are needed to repopulate them. We therefore demand there exist at least one wandering robot, meaning $2RA_c \leq k^F - 1$. As wandering robots can be needed in both readiness areas and therefore cannot be too far away from both of them, we define for each of them a set of segments they should patrol in, referred to as *patrol regions*. We also relate to each readiness area as composed of RA_c patrol regions, each of size d'. We conclude there is total number of k^F regions marked by $RA_i, 0 \leq i < k^F$ (readiness areas are represented by regions $PR_0, ..., PR_{RA_c-1}, PR_{k^F-RA_c}, ..., PR_{k^F-1}$), and the size of each region should be at least d' (equal to or greater than the distance between window robots). All region sizes are fixed through the entire patrol, and at any time there is exactly one robot allocated to patrol at each region, elected according to fence robots' relative distance to the window.

The *patrol purpose* of all fence robots is the same: each robot should be at its currently allocated region facing the same direction as all window robots (if the region is part of some readiness area, the robot should also be at the correct segment forming distance d' from window robots), and at any time to use the same movement decision as the window robots use, only without leaving the region (so if needed it can choose to stay at the same segment). When all fence robots fulfill their patrol purpose we say the system is in a *steady state*, and the deception is preserved.

Sometimes fence robots are not in a steady state, so the prime task of each robot is to move into its correct region. This process can take a while, and sometimes eventually deception is revealed during that time. We therefore define the terms that require we interfere in the random decision making and return back into a steady state, a process called *stabilizing phase*. In order to satisfy the continuous demand, there should always be at least RA_c robots in both readiness areas that satisfy their patrol purpose, all adjacent to the window robots and to one another. We denote the minimal time it would take all fence robots to return into a steady state by θ and require that $\theta \leq 2d'RA_c$, to guarantee that during stabilizing phase the RA_c robots are able to preserve the deception. As a result, the number k' of robots is bounded by $k' \leq \frac{2N}{|F|+2|W|}k < 2k$.

If a movement-option chosen at random results in a violation of those conditions, stabilizing phase begins. In order to minimize the length of the stabilizing phase, regions $PR_{RA_c}, ..., PR_{k^F-RA_c-1}$ should be of the same size[2].

Deception by Random Patrol (Seemingly)

We now describe how the stabilizing phase is implemented, satisfying all demands described. At the end of the stabilizing phase, each fence robot should be in its current segment, satisfying his patrol purpose, and facing the same direction as it was at the beginning of the phase. We, in fact, demand that each window robot will start and finish stabilizing phase at the same segment facing the same direction. We refer to a sequence of movement-options as a *path*, and a path in which each window robot starts and finishes at the same segment facing the same direction as a *closed path*. Closed paths must contain an even number of movement-options and an even number of change-direction movement-options.

During the stabilizing phase we also wish to satisfy the distribution demand, so the path we use is based on the original Binomial distribution of p in the following manner: we choose values out of the Binomial random variable until the time it takes to perform the resulting path is equal to or greater than θ. The number of movements chosen should be even, otherwise the path could not be closed, so if needed we choose one more random value. The time to perform the resulting path is bounded by $2d'RA_c + 2\tau$. Algorithm 1 describes how to convert the path described into a set of movement-option returning all robots into steady state:

Algorithm 1 Seemingly Random Patrol

1: **Closed path stage**: edit the path to be a closed path.
2: **Validation stage**: make changes to the path so deception is not revealed.

Closed path stage

We first wish to take the given path and create a closed path that satisfies the demands described above. The following lemma shows that the required number of changes to the original path is minimal:

Lemma 3.2. *Closed path stage takes the described path and makes up to one order change and one movement-option change to create a closed path out of it.*

Proof sketch. The number of change-direction movement-options must be even in a closed path, so if needed we must convert one movement-option in the path, while still fulfils the stabilizing phase demands. The converted decision chosen is the one that optimally fulfilling distribution demand. We randomly replace one change-direction movement-option into same-direction movement-option, and using intermediate value theorem, there exist some same-direction movement-option that if converted the resulting path is a closed path.

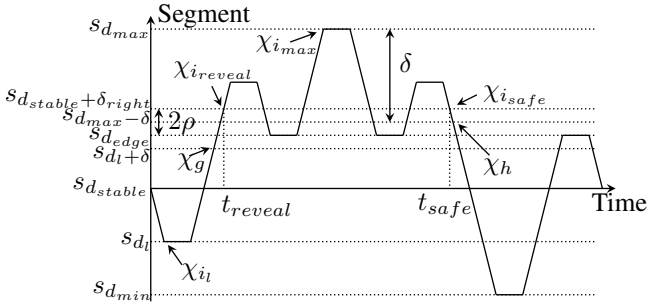
We denote by χ the closed path received after performing this change. All changes done are required and minimal, and recalling that changing the order of decisions that were picked randomly does not change expected value and variance of the random variable, thus we satisfy distribution demand.

Validation stage

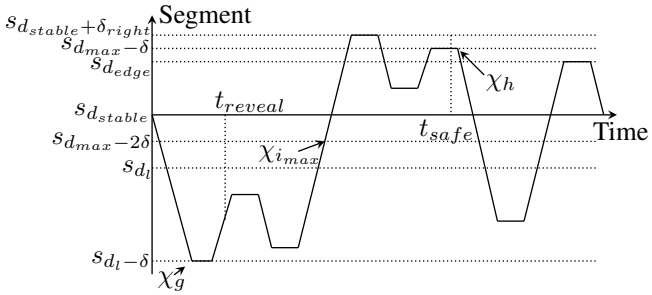
The validation stage takes χ and converts it into a path preserving the deception, as some paths can cause all window robots to move too much into one direction until there are no robots ready to enter W when needed. We wish to choose up to two movement-options that need a different placement in χ.

At the beginning of the stabilizing phase, each window robot can move up to $\delta_{right} \geq 0$ segments to the right and up to $\delta_{left} \geq 0$ segments to the left without revealing the deception. Looking at one window robot (as they perform coordinated patrol) starting at segment $s_{d_{stable}}$, we mark by $s_{d_{max}}$

[2]Full proof exists but was omitted due to paper size restriction



(a) closed path stage result



(b) validation stage result

Figure 3: Validation stage demonstration. At 3a deception is revealed while at 3b not.

and $s_{d_{min}}$ the maximal and minimal segments he would reach executing χ, by t_{reveal} the first time he moves pass segments $s_{d_{stable}+\delta_{right}}$ or $s_{d_{stable}-\delta_{left}}$, and by $t_{safe} \geq t_{reveal}$ the first time his current and all next positions are in the range between $s_{d_{stable}-\delta_{left}}$ and $s_{d_{stable}+\delta_{right}}$ (see Figure 3).

We assume, without loss of generality, that χ reveals the deception at the right side. We mark several movement decisions at $\chi = \chi_0, ..., \chi_{i_l}, ..., \chi_{i_{reveal}}, ..., \chi_{i_{max}}, ..., \chi_{i_{safe}}, ..., \chi_{2n-1}$:

1. $\chi_{i_{reveal}}$: Movement-decision finished at time t_{reveal}, causing the first movement to $s_{d_{stable}+\delta_{right}+1}$.
2. χ_{i_l}: We denote by $s_{d_l}, d_l \leq d_{stable}$ the leftmost segment the robot reaches before time t_{reveal}. χ_{i_l} is the last change-direction movement-option from left to right that happens before $\chi_{i_{reveal}}$ at s_{d_l}.
3. $\chi_{i_{max}}$: Change-direction movement-option from right to left after reaching $s_{d_{max}}$ for the last time.
4. $\chi_{i_{safe}}$: Movement-option finished at time t_{safe}, causing the last movement to $s_{d_{stable}+\delta_{right}}$.

Consider the movements happening between $\chi_{i_{max}}$ and $\chi_{i_{safe}}$. We marked by $s_{d_{edge}}, d_{edge} \leq d_{stable}+\delta_{right}$ the leftmost segment the robot reaches during this time and let $\rho = \frac{d_{stable}+\delta_{right}-d_{edge}}{2}$. We define $\delta = d_{max} - d_{stable} - \delta_{right} + \rho$. We denote by χ_h the first movement from $s_{d_{max}-\delta+1}$ to $s_{d_{max}-\delta}$ after t_{safe}.

There is at least one movement-option before t_{reveal} that is changing-direction movement-option from left to right, so χ_{i_l} must exist. We denote by χ_g the first movement from $s_{d_l+\delta-1}$ to $s_{d_l+\delta}$ after χ_{i_l}. Furthermore, χ_g must happen before $\chi_{i_{reveal}}$ as $d_{stable} + \delta_{right} \geq d_l + \delta$. We look at $\bar{\chi} = \chi_0, .., \chi_{i_l-1}, \chi_{i_l+1}, .., \chi_g, \chi_{i_l}, \chi_{g+1}, .., \chi_{i_{max}-1}, \chi_{i_{max}+1}, .., \chi_h, \chi_{i_{max}}, \chi_{h+1}, .., \chi_{2n-1}$.

Theorem 3.3. *It is possible to simulate* mrand *patrol of* $k < k' \leq \frac{2N}{|F|+2|W|}k$ *robots along* W *for infinite time.*

Proof sketch. $\bar{\chi}$ is a closed path, based on summing the total distance the robot passed. During movement, the robot does not pass segment $s_{d_{stable}-\delta_{right}}$, and only reaches to it, so the deception is not revealed at any stage of movement. The expected value of p the adversary observes is based on the random decision of Binomial variable.

4 The Scarecrows Deception Model

In this section we wish to study another way a defender can deceive the adversary, this time using *scarecrow robots*-robots that look like real robots in the eye of the adversary, but actually have no sensing abilities, and therefore are much cheaper, require less communication resources and easier to maintain compared to real robots, and we ignore them when calculating the probability of penetration detection (they contribute 0 to the calculation). We denote the number of scarecrow robots by k^s and the number of regular robots by k. We study the case where $k < \frac{N}{t}$ (otherwise we could have performed a perfect patrol), and show how using scarecrows we can successfully simulate perfect patrol.

We assume that the adversary does not know we use scarecrows against him. The adversary sees $\Omega = k + k^s$ robots patrolling with distance $d = \frac{N}{k+k^s}$ between them, and since he is rational, he will try penetrating through one of the segments where $\min ppd_i$ is obtained: in perfect patrol where all segments have $ppd_i = 1$ in the eye of adversary, penetration occurs at random through any segment with uniform distribution, and in mrand patrol when using $k + k^s$ robots and optimal p, penetration occurs through segments with $\min ppd_i$. We wish to find the placement of scarecrows resulting in maximal expected ppd of the above segments (all or ones with minimal ppd, in perfect of mrand patrol, respectively).

4.1 Combining Scarecrows in Perfect Patrol

Whenever $d \leq t$ the adversary expects to see a perfect patrol. Allocation of scarecrows in this case is equivalent to dividing all k^s scarecrows into $\min\{k, k^s\}$ groups, and placing each group between two real robots along the perimeter. We wish to find the optimal group sizes resulting in maximal expected probability of penetration detection over all segments.

Let $m = \lfloor \frac{t}{d} \rfloor - 1$, m represent the maximal group size we can place between two real robots and still have $ppd_i = 1$ at all segments. If all scarecrows can be divided into groups of at most m scarecrows we would get $ppd_i = 1$ at all segments, so the placement is optimal. Otherwise, the difference between every two group sizes should be at most 1, or by mediating those group sizes we get bigger expected probability of penetration detection in contradiction to division optimality. We get that the optimal division into groups in this case is: $k - b$ groups of size $\lfloor \frac{k^s}{k} \rfloor \geq m$ and b groups of size $\lfloor \frac{k^s}{k} \rfloor + 1 > m$. Calculating the resulting expected penetration detection rate, Theorem 4.1 follows.

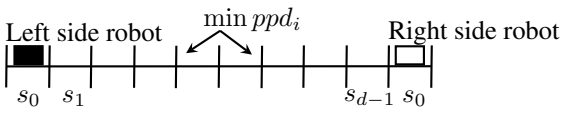


Figure 4: A section of d segment with robots at s_o. The robot on the left is a real robot, while the robot on the right is a scarecrow.

Theorem 4.1. *Let $m = \lfloor \frac{t}{d} \rfloor - 1, b = (k^s) \mod k$. When performing perfect patrol of k real robots and k^s scarecrows placed optimally as described, the resulting expected probability of penetration detection along P is:*

$$Exp\left[PPD^{k^s}\right] = \begin{cases} \frac{k^s t}{N} + \frac{k-k^s}{k+k^s} & 0 = \lfloor \frac{k^s}{k} \rfloor, m = 0 \\ \frac{kt}{N} & 0 \leq m < \lfloor \frac{k^s}{k} \rfloor \\ \frac{(k-b)(m+1)d+bt}{N} & 1 \leq \lfloor \frac{k^s}{k} \rfloor = m < \lceil \frac{k^s}{k} \rceil \\ 1 & \lceil \frac{k^s}{k} \rceil \leq m, m \geq 1 \end{cases}$$

4.2 Combining Scarecrows in mrand Patrol

We now combine scarecrows in mrand patrol, meaning $t < d$. The optimal p used by the mrand patrol algorithm depends on Ω. Furthermore, the more robots patrolling, the bigger $\min_i ppd_i$ is. We wish to calculate the expected penetration detection rate through segments with $\min ppd_i$.

Consider a single section of d segments (due to the symmetry of the perimeter, all such segments are identical throughout the perimeter). During a period of t time units, each segment is visited by at most one robot, since $t < d$. Therefore, when calculating ppd_i of some segment, we sum the probabilities that the robot on the right and the robot on the left will reach that segment (see Figure 4), and only them (no other robot can reach those segments within t time units). We mark by $Exp\left[PPD^{k^s}\right]_i$ the expected probability of penetration detection rate at all s_i along P.

Lemma 4.2. *Regardless of scarecrows allocations, $Exp\left[PPD^{k^s}\right]_i = \frac{k}{k+k^s} ppd_i$ for all i.*

As conclusion, $\min\{Exp\left[PPD_i^{k^s}\right]_i\} = \frac{k}{k+k^s} \min ppd_i$.

Theorem 4.3. *When combining scarecrows in random patrol using any placement of scarecrows, $Exp\left[PPD_i^{k^s}\right] = \frac{k}{k+k^s} \min ppd_i$.*

Proof. We know the penetration occurs in segments where $ppd_i = \min ppd_i$ because the adversary does not know there are scarecrows participating. Hence, the expected penetration detection rate at those segments is $Exp\left[PPD_i^{k^s}\right]_i = \frac{k}{k+k^s} \min ppd_i$. The average expected value of all those segments is $Exp\left[PPD_i^{k^s}\right] = \frac{k}{k+k^s} \min ppd_i$, as required. □

Corollary 4.3.1. *Using scarecrows does not change the location of segments where $\min ppd_i$ is obtained.*

Corollary 4.3.1 suggests that even if the opponent would have known there are scarecrows participating, he would not have changed his penetration strategy.

We now fix the total number of robots to be Ω, as this is what the adversary sees and acts according to, and place the robots optimally as described. Analyzing the results of Theorems 4.1, 4.3 we conclude that the function

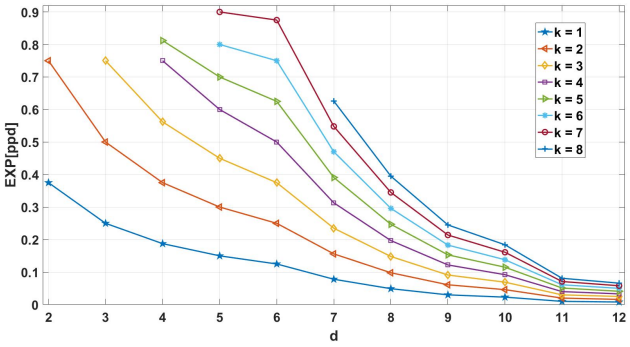


Figure 5: Demonstration of scarecrows influence on $Exp\,[ppd]$, using $t = 6, \tau = 1, \Omega = 8$. For $2 \leq d \leq 6$ the robots performed perfect patrol, and for $7 \leq d \leq 12$ the robots performed mrand patrol.

$Exp\left[PPD^{k^s}\right](k)$ is monotonically increasing at k when using fixed Ω, hence as the number of real robots in Ω increases, $Exp\left[PPD_i^{k^s}\right] = \frac{k}{\Omega} \min ppd_i$ increases.

We have fully implemented our scarecrow-based deception model. We used fixed values of t, τ an Ω, and calculated the influence of different values of $k \leq \Omega$ and d on the resulting $Exp\,[ppd]$ (recall that $k < \frac{N}{t}$ and $d \leq 2t - \tau + 1$). In Figure 5 a partial set of results is demonstrated. The case when $k = \Omega$ represents the belief of the opponent, and for all values of d result in higher penetration detection rate compared to all other k values. Using this analysis, the defender can choose the tradeoff between expected probability of penetration detection the adversary assumes, and the *actual* expected probability of penetration detection achieved, based on his resources.

5 Conclusions and Future Work

This paper examines the problem of adversarial deception in the multi-robot adversarial patrolling problem, along a perimeter. We presented two different deception models. The first model is based on the inability of the adversary to view and act along the entire perimeter, thus the robots deceive the adversary to think there are more robots than there actually are. This is achieved either by adjusting the robots' model (velocity) if using a deterministic patrol strategy, or by adjusting the behavior of the robots, if using a random strategy. In the second model, the team of robots consists of both real robots and *scarecrows*, that just appear as real robots, but actually have no detection capabilities. In this case we have showed that integration of scarecrows in patrol enables optimizing defenders' utilization of resources.

There are many directions we wish to pursue in the future, including combining the two deception models, analyzing the influence of deception on penetration detection rate along F, and adding new deception models (for example: different window outlines and creating honeypots).

References

[Agmon *et al.*, 2008a] Noa Agmon, Sarit Kraus, and Gal A Kaminka. Multi-robot perimeter patrol in adversarial settings. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 2339–2345. IEEE, 2008.

[Agmon *et al.*, 2008b] Noa Agmon, Vladimir Sadov, Gal A Kaminka, and Sarit Kraus. The impact of adversarial knowledge on adversarial planning in perimeter patrol. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems*, pages 55–62. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[Agmon *et al.*, 2011] Noa Agmon, Gal A Kaminka, and Sarit Kraus. Multi-robot adversarial patrolling: facing a full-knowledge opponent. *Journal of Artificial Intelligence Research (JAIR)*, pages 887–916, 2011.

[Agmon *et al.*, 2012] Noa Agmon, Chien-Liang Fok, Yehuda Emaliah, Peter Stone, Christine Julien, and Sriram Vishwanath. On coordination in practical multi-robot patrol. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[An *et al.*, 2012] Bo An, David Kempe, Christopher Kiekintveld, Eric Shieh, Satinder Singh, Milind Tambe, and Yevgeniy Vorobeychik. Security games with limited surveillance. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 1241–1248, 2012.

[An *et al.*, 2013] Bo An, Matthew Brown, Yevgeniy Vorobeychik, and Milind Tambe. Security games with surveillance cost and optimal timing of attack execution. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems*, pages 223–230, 2013.

[Delle Fave *et al.*, 2015] Francesco Maria Delle Fave, Eric Shieh, Manish Jain, Albert Xin Jiang, Heather Rosoff, Milind Tambe, and John P Sullivan. Efficient solutions for joint activity based security games: fast algorithms, results and a field experiment on a transit system. *Autonomous Agents and Multi-Agent Systems*, 29(5):787–820, 2015.

[Elmaliach *et al.*, 2008] Yehuda Elmaliach, Asaf Shiloni, and Gal A Kaminka. A realistic model of frequency-based multi-robot fence patrolling. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems*, pages 63–70, 2008.

[Elmaliach *et al.*, 2009] Yehuda Elmaliach, Noa Agmon, and Gal A Kaminka. Multi-robot area patrol under frequency constraints. *Annals of Mathematics and Artificial Intelligence*, 57(3-4):293–320, 2009.

[Felemban, 2013] Emad Felemban. Advanced border intrusion detection and surveillance using wireless sensor network technology. *International Journal of Communications, Network and System Sciences*, 6(5):251, 2013.

[Machado *et al.*, 2002] Aydano Machado, Geber Ramalho, Jean-Daniel Zucker, and Alexis Drogoul. Multi-agent patrolling: An empirical analysis of alternative architectures. In *Proceedings of the International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 155–170. Springer, 2002.

[Paruchuri *et al.*, 2006] Praveen Paruchuri, Milind Tambe, Fernando Ordóñez, and Sarit Kraus. Security in multiagent systems by policy randomization. In *Proceedings of the 5th International Conference on Autonomous Agents and Multiagent Systems*, pages 273–280. ACM, 2006.

[Sun *et al.*, 2011] Zhi Sun, Pu Wang, Mehmet C Vuran, Mznah A Al-Rodhaan, Abdullah M Al-Dhelaan, and Ian F Akyildiz. Bordersense: Border patrol through advanced wireless sensor networks. *Ad Hoc Networks*, 9(3):468–477, 2011.

[Yin *et al.*, 2011] Zhengyu Yin, Manish Jain, Milind Tambe, and Fernando Ordonez. Risk-averse strategies for security games with execution and observational uncertainty. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[Zhang and Luo, 2014] Youzhi Zhang and Xudong Luo. Security games with partial surveillance. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems*, pages 1527–1528, 2014.