# On Events in Multi-Robot Patrol in Adversarial Environments

Noa Agmon
Department of Computer Science and Applied Mathematics
Weizmann Institute of Science, Israel
noa.agmon@weizmann.ac.il

## ABSTRACT

The problem of multi-robot patrol in adversarial environments has been gaining considerable interest during the recent years.In this problem, a team of mobile robots is required to repeatedly visit some target area in order to detect penetrations that are controlled by an adversary. Little has been written so far on the nature of the event of penetration, and it is commonly assumed that the goal of the robots is to detect the penetration at any time during its occurrence.In this paper we offer a new definition of an event, with correlation to a utility function such that the detection of the event by the robots in different stages of its occurrence grants the robots a different reward. The goal of the robots is, then, to maximize their utility from detecting the event.

We provide three different models of events, for which we describe algorithms for calculating the expected utility from detecting the event and discuss the how the model influences the optimality of the patrol algorithm. In the first and basic model, we assume that there exists a reward function such that detecting an event at different times grants the robots with an associated reward. In the second model, the event might evolve during its occurrence, and this progression correlates to both different rewards and to growing probability of detection. Finally, we consider a general model, in which the event can be detected from distance, where the probability of detection depends both on the distance from the robot and on the current state of the event. Last, we discuss how the new event models presented in this paper set grounds for handling the problem of patrol in heterogeneous environments, where parts of the perimeter could be more sensitive to occurrence of events.

## Categories and Subject Descriptors

I.2.9 [**Robotics**]: Autonomous vehicles; I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems

## General Terms

Algorithms, Security

## Keywords

Multi-robot systems, Multi-robot path planning, Adversarial/game domains

**Cite as:** On Events in Multi-Robot Patrol in Adversarial Environments, Noa Agmon, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 591-598

## 1. INTRODUCTION

The problem of multi-robot patrol in adversarial environments has been gaining considerable interest during the recent years (e.g. [1, 2, 13, 4, 5]). In this problem, a team of mobile robots is required to repeatedly visit some target area in order to detect penetrations. These penetrations are controlled by an adversary, that is assumed to use the information it obtained on the patrolling robots in order to pass through the patrol path undetected. This problem is of interest mainly due to its immediate applicability in various security settings.

Little has been written so far on the nature of the event of penetration. Previous work only assumed that the time it takes the adversary to penetrate through the patrol path is not instantaneous. Specifically, Agmon et al. [2] denoted this penetration time by $t$. Basilico et al. [5] assume the intruder passes through the area in some possible paths. The goal of the robots, in all cases, is to detect the penetration during the $t$ time units the adversary is along the patrol path, with no reference to the relative time of detection, i.e., detecting the adversary at any time rewards the robots in “success”, and if it was undetected during $t$ time units, the robots have failed.

It is often the case that the relative time of detection has high implication on the success or failure of the patrol task, i.e., the robots should be motivated to detect events as soon as possible. In addition, the event might evolve and effect the probability of detecting it. To our knowledge, these problems remained open challenges, and this paper aims to solving them.

Therefore in this paper we offer a new definition of an event (rather than a simple penetration), in which the robots gain a different utility value (reward) according to the relative time the event was detected. The goal of the robots is, then, to maximize their utility from detecting the event.

This new perspective of the event makes the problem suitable for additional scenarios such as detection of fire and leaks of hazardous substances. Such events tend to have an evolving nature, thus the behavior of the robots, namely the choice of patrol algorithm, should be defined appropriately. Here, the nature of the adversary is to model the behavior of the system in various environments (similar to adversarial existence in distributed systems [9]). For example, making guarantees for the performance of the system in a strong adversarial environment corresponds to modeling the worst case behavior of the system. Similarly, performing against a weak adversary correlates to maximizing the expected utility of the robots under no predicted special threats.

The main contribution of this paper is threefold. First, it defines the new event model, and describes possible implementations of this model. Second, it shows how it is possible to determine in polynomial time the probability of detecting an event in these implementations. Finally, it describes the influence of this new event model and its applications on the choice of optimal patrol algorithm.

We provide three different models of events, for which we describe algorithms for calculating the expected utility from detecting the event. In the first and basic model, we assume that there exists a reward function such that detecting an event at time $1 \leq i \leq t$ grants the robots with an associated reward $\mathsf{rwd}_i$. The resulted expected utility is a function of the chosen patrol algorithm. We then show that when working in a strong adversarial model, in which the goal is to maximize the minimal expected utility, in some cases the optimal patrol algorithm remains the same for all non-increasing reward function. In addition, we prove that in a weak adversarial model, in which an event can occur at any point at random with uniform distribution, a simple deterministic algorithm is optimal for all possible reward functions.

We then describe a second model, in which the event might evolve during the $t$ time units of its occurrence, and this progression correlates to growing probability of detection as well as decreasing reward function. Specifically, we assume that the probability of detecting an event initially is not 1, and it increases as it evolves. An example for such events are gas leaks and fires, that are usually initially spatially limited, thus it is more difficult to detect them. We describe a polynomial time algorithm for calculating the expected utility in such cases, and prove that the simple deterministic algorithm remains optimal for a weak, random, adversarial environment.

Finally, we consider a general model, in which the event can be detected from distance, where the probability of detection depends both on the distance from the robot (probability decreases as the distance from the root increases), and on the current state of the event (probability of detection increases as the event progresses). Also here we describe a polynomial-time algorithm for calculating the expected utility from detecting the event, and show that in some cases the deterministic algorithm is no longer optimal in the weak adversarial model.

Last, we discuss possible implications of the new event models presented here. Mainly, we discuss how the new event models set grounds for handling the problem of patrol in heterogenous environments. In such environments, parts of the patrol path could be more sensitive to occurrence of events such as areas that are easier to penetrate or fields that are more likely to catch fire.

## 2. RELATED WORK

Systems of multiple robots working together to patrol in some target area, have been studied in various contexts, where researches usually concentrate either on optimizing some frequency criteria by the patrol algorithm [3, 7] or optimizing probability of adversarial detection [1, 2, 13].

Agmon et. al. [1, 2] introduced the multi-robot *adversarial* perimeter patrol along with the robotic model we base our work upon. They describe an algorithm for finding the optimal patrol when working in different adversarial models, mainly the full knowledge adversarial model [1] and the zero-knowledge adversarial model [2]. They consider also the case of imperfect sensing, however in their work they relate to the imperfect sensing capabilities of the robots, and do not refer to probability of detection that changes as the event evolves, and their solution to imperfect sensing is strictly limited to patrolling along an open fence, and does not refer to perimeters.

This work can be considered as an extension of the work by Agmon et al. from Markov chains to Markov Decision Process (MDP) by adding utilities to the system. However this work does not make use of MDP's in in a classical manner, but incorporate the Markovian assumptions in dynamic programming inspired algorithms. Uncertainty in sensing, as handled in this paper, can lead to comparing this work to Partially Observable MDP's (POMDP), such as the work of Koenig et al. [8] for robot navigation. However, the robots are not uncertain about their state, and their level of uncertainty (probability of detection) is considered analytically and solved exactly optimally.

Sak et al. [13] considered multi-agent patrol in general graphs rather than perimeters, as is our focus here. In contrast to our work, they concentrated on an empirical evaluation using a simulation, of several non-deterministic patrol algorithms, and they do not prove optimality analytically. Also, they do not make any reference to evolving events or various probabilities of detecting events as we do here.

Other closely related work is the work by Paruchuri et al. [11, 10] and Pita et al. [12], which considered the problem of placing security checkpoints in adversarial environments. They use policy randomization for the agents' behavior in order to maximize their rewards. In their work, the adversary has full knowledge of the agents' behavior, therefore it can use it in order to minimize its probability of being caught in some checkpoint. They again do not consider events that may evolve, nor do they provide *optimal* polynomial-time solutions, as presented herein.

Amigoni et al. [4] and Basilico et al. [6, 5] also used a game-theoretic approach for determining the optimal strategy for patrolling agents, using leader-follower games. In their work they consider an environment in which a robot can move between any two nodes in a graph, with possible levels of information obtained by the adversary and with possible paths traveled by the intruder along the graph. Their solution is suitable for a single robot, and since the computation of the optimal strategy is exponential, they described a heuristic approach for finding a solution.

## 3. THE ROBOTIC MODEL, ENVIRONMENT AND THE NEW EVENT MODEL

In the perimeter-patrol problem in adversarial environments, we are given $k$ homogenous robots that are required to patrol around a closed polygon, where their goal is to detect penetrations that are controlled by an adversary. The perimeter of the polygon is divided into $N$ segments, each of uniform *time distance*, i.e., the robots travel through one segment per time cycle.

We base our patrol algorithm on the nondeterministic framework set by Agmon et al. [2], as follows. At each time step the robots have two options as to their next move:

$$\text{Probability of next move} = \begin{cases} p & \text{Go straight;} \\ 1-p & \text{Turn around} \end{cases}$$

We assume a realistic movement model of robots in which turning around is a costly operation (similar to differential drive robots commonly used in research labs), and model this cost in time, i.e., when turning around a robot resides in its current segment for $\tau$ time units. For simplicity reasons we assume herein that $\tau = 1$.

In [2] it was proven that an optimal patrol in both strong and weak adversarial model requires the robots to maintain uniform distribution between every two consecutive robots throughout the execution of the patrol. The optimality proof of this requirement is based on the fact that the probability of detecting penetrations decreases as the distance from the robot increases, thus by assuring that all robots are uniformly placed along the perimeter, it is guaranteed that the maximal distance between any two robots is minimized. Motivated by this optimality proof, and also proven later on in this paper, we require that the robots will be coordinated in the sense that if decided to turn around, then it is done simultaneously by all robots. In addition, the robots are required to be initially spread uniformly along the perimeter with uniform time distance between every two consecutive robots along the path. We denote the distance between every two consecutive robots by $d = N/k$.

In the new *event-oriented* multi-robot patrol problem, we are given a reward function $\mathsf{rwd} : \{1, \ldots, t\} \rightarrow \mathbf{N}$ such that the robots are rewarded the utility $\mathsf{rwd}_i$ if the event is detected at time $i$, $1 \leq i \leq t$. Generally, we assume that $\mathsf{rwd}_i$ is monotonically non-increasing, i.e., $\mathsf{rwd}_i \geq \mathsf{rwd}_{i+1}$. This is a realistic assumption for event detection, as usually we would prefer detecting events sooner rather than later.

In order to find the *optimal patrol algorithm* for the robots, we need to perform the following two steps:

1. Calculate the *expected* utility for detecting an event at each segment. This depends on the executed patrol algorithm, thus is a function of $p$ (probability of going straight at each time cycle).

2. Calculate the optimal patrol algorithm (characterized by $p$), which depends on the adversarial model.

In this paper we will refer to each of these steps, i.e., we will describe algorithms for determining the expected utility for detecting events, and show the influence of the adversarial model on the optimal choice of $p$.

## 3.1 The adversarial model

In previous work in multi-robot patrol in adversarial environments (e.g. [2, 6]), the nature of the adversary was well defined. The robots are instructed to patrol along a path and detect penetrations that are controlled by the adversary. Therefore if assuming a rational adversary, it will take advantage of the knowledge it obtained on the patrol to try and penetrate successfully, i.e., without being detected. Therefore the optimality of the patrol was examined in different adversarial models, where the model vary in the level of adversarial knowledge of the patrol, mainly zero knowledge or full knowledge.

When discussing events that are not necessarily security-related, the existence of the adversary takes a new perspective. In this case, the means of adversary is used to conveniently model the behavior of the system (similar to the classical Byzantine fault model in distributed systems [9]). A strong adversarial model (similar to the full-knowledge adversary) corresponds to the *worst case* behavior of the system, and an optimal patrol algorithm maximizes the minimal expected reward. For example, when considering the case of fire-detection, the question asked is what is the worst possible expected utility from fire detection, and the objective is to maximize this lowest utility as much a possible. On the other hand, a weak adversarial model (similar to the zero-knowledge adversary) corresponds to the behavior of the system when there is no known special threat, thus the goal is to maximize the total expected utility.

We will therefore discuss the implication of the adversarial model on the optimal patrol algorithm in the two adversarial environments: strong and weak.

# 4. DETERMINING THE EXPECTED UTILITY - BASIC CASE

In this section we describe a method for determining the *expected* utility of the robots from detecting the event during its occurrence. This value depends on the probability that some robot will visit the segment during the $t$ time units of the duration of the event, thus will detect it.

Formally, the expected utility from detecting the event in section $s_i$, $1 \leq 1 \leq N$, is defined as the probability of detecting the event in segment $s_i$ at some time $j, 1 \leq j \leq t$ multiplied by the utility (reward) that is gained by the detection at that stage, i.e., $\mathsf{rwd}_j$. We denote the expected utility from detection at segment $s_i$ by $\mathsf{eud}_i$, and the probability of visiting $s_i$ at time $j$ in the segment by $\mathsf{pv}_i^j$.

Initially, we assume the robots have perfect detection capabilities, i.e., if the event occurs while under the sensorial range of the robot it will surely detect it. Therefore the probability of detecting the event during its occurrence in time $j$, $\mathsf{pv}_i^j$, is exactly the probability that some robot will visit segment $s_i$ at time $j$ of the event.

Therefore $\mathsf{eud}_i = \sum_{j=1}^{t} \mathsf{pv}_i^j \times \mathsf{rwd}_j$.

One of the building blocks upon which the algorithmic framework described in [2] is based upon, is the fact that the probability of detection decreases or remains the same as the distance from a robot decreases, i.e., it is a monotonic non-increasing function. This fact was used in [1] in proving that in order to achieve optimal probability of detection the robots should be uniformly placed (in time) around the perimeter throughout the execution of the patrol. In order to use this framework, we are therefore required to show that the expected utility from detecting events also decreases as the distance from the a robot increases.

LEMMA 1. *Given the monotonically non-increasing reward function* $\mathsf{rwd} : \{1, \ldots, t\} \rightarrow N$*, the expected utility from detecting an event decreases or remains the same as the distance from a robot increases.*

PROOF. Assume the robot $R$ resides in segment $s_0$, hence we consider segments $s_1, \ldots s_t$ to its right, and segments $s_{-1}, \ldots, s_{-t+1}$ to its left. Let us consider first the segments to its right (with positive indexes). It is necessary to show that $\mathsf{eud}_i > \mathsf{eud}_{i+1}$. Following [1] that have proven that as the distance of a segment from the current location of $R$ increases the probability of arriving there in $t$ time units decreases, then $\mathsf{pv}_i^j \geq \mathsf{pv}_{i+1}^j$, $\forall j \leq t$. It is possible to arrive for the first time at segment $s_i$ at times $i, i+2, i+4, \ldots$ and in segment $s_{i+1}$ at times $i+1, i+3, \ldots$, thus since $\mathsf{rwd}$ is monotonically non-increasing, the rewards of the first visits at $s_{j+1}$ are necessarily not larger than the rewards from first visiting $s_i$. The expected utility is, therefore, $\mathsf{eud}_i = \sum_{j=1}^{t} \mathsf{pv}_i^j \mathsf{rwd}_j \geq \sum_{j=1}^{t} \mathsf{pv}_{i+1}^j \mathsf{rwd}_j = \mathsf{eud}_{i+1}$. The segments to the left are a reflecting image of the segments to the right, thus the proof follows directly. □

In Algorithm 1, we describe the algorithm DetExpectedUtility for determining the expected utility from detecting an event at a segment $s_i$. The algorithm receives as input the segment $s_i$ and the reward function $\mathsf{rwd} = \{\mathsf{rwd}_1, \ldots \mathsf{rwd}_t\}$, and calculates the probability of visiting the segment $s_i$ at all times. This algorithm is based on the dynamic-programming inspired algorithm described in [1], and works as follows. It initializes a zero table $M$ with a value 1 in location $s_i$. Then it fills in the rows, one by one, using the Markov chain determining the transitions between states (lines $6 - 13$ in the algorithm).

Note that since the robots maintain uniform distance between them throughout the execution, the entire setting is symmetric, thus it is not necessary to compute the expected utility of *all* segments $s_i, 1 \leq i \leq N$, but only for $d = N/k$ segments between every two consecutive robots. The time complexity of DetExpectedUtility is the time complexity of filling in the matrix of size $\mathcal{O}(dt)$, and as it takes $\mathcal{O}(i)$ to fill in each entry in row $i$, so altogether the time complexity is $\mathcal{O}(d^2t^2)$ for each segment, thus $\mathcal{O}(d^3t^2)$ for all $d$ segments.

**Algorithm 1** Algorithm DetExpectedUtility($loc, d, t, \mathsf{rwd} = \{\mathsf{rwd}_1, \ldots, \mathsf{rwd}_t\}$)

```
1: Create matrix $Y$ of size $(2d+2) \times (t+1)$, initialized with 0s
2: Set $Y[0, s_{loc}] \leftarrow 1$
3: Fill in all entries in $Y$ gradually using the following rules.
4: for $r \leftarrow 1$ to $t$ do
5:     for $i \leftarrow 1$ to $d$ do
6:         For each entry $Y[r, s_i^{cw}]$, set value to $p \cdot Y[r - 1, s_{i+1}^{cw}] + (1-p) \cdot Y[r-1, s_i^{cc}]$
7:         For each entry $Y_[r, s_i^{cc}]$, set value to $p \cdot Y[r-1, s_{i-1}^{cc}] + (1-p) \cdot Y[r-1, s_i^{cw}]$
8:     end for
9:     For result states, set entry $Y[r, s_{res}] = Y[r-1, s_{res}] + p \cdot \mathsf{rwd}_i \cdot \{Y[r-1, s_1^{cw}] + Y[r-1, s_d^{cc}]\}$
10: end for
11: Return $Y_[t, s_{res}]$
```

## 5. WHEN EVENTS EVOLVE AND CORRELATE TO SENSING ABILITY

In many cases such as fire spreading and leaks of hazardous substances, it might be very difficult to detect the event in its early stages, and as the event evolves the probability of detection increases. In this section we introduce an algorithm for determining the expected utility of the robots as a function of the chosen patrol algorithm ($p$) in such cases. The input to the problem is the reward function as well as the probability of detecting the event at every time $i$, $1 \leq i \leq t$. We denote the probability of detection by $\mathsf{pdt} = \{p_d^1, \ldots, p_d^t\}$, where $p_d^i$ is the probability of detecting the event at time $i$.

The probability of detecting the event in a segment $s_i$ in cases of imperfect detection does not correlate to the probability of only the *first* visit at some time $j, 1 \leq j \leq t$ (as seen previously), but to the probability of *any* visit to $s_i$ (since the event is not necessarily detected in the first visit). Denote by $\mathsf{pv}_i^j(l)$ the probability that some robot will visit segment $s_i$ at time $j$ of the event for the $l$'th time. Therefore the probability of detecting the event is defined as follows.

$$\mathsf{eud}_i = \sum_{j=1}^{t} (\mathsf{pv}_i^j(1) \cdot p_d^j \cdot \mathsf{rwd}_j) + \sum_{j=1}^{t} \{\mathsf{pv}_i^j(1) \cdot (1 - p_d^j)\} \times \quad (1)$$

$\{\sum_{j=1}^{t} \mathsf{pv}_i^j(2) \cdot p_d^j \cdot \mathsf{rwd}_j + \cdots \ \{\sum_{j=1}^{t} \mathsf{pv}_i^j(t) \cdot p_d^j \cdot \mathsf{rwd}_j\}\}$

In words, the expected reward is the probability of detecting the event in the first visit—at any possible time $i, 1 \leq i \leq t$—multiplied by the reward for detection at that time, plus the probability that it was not detected at the first visit but later on (second, third, etc.).

As shown in the model in the previous section, a basic requirement for using the algorithmic framework of multi-robot patrol is that the expected utility is monotonically non-increasing. Therefore also in the case of imperfect sensing due to event progression we need to prove that the expected utility from detecting an event in segment $s_i$ decreases of remains the same as the distance of $s_i$ from the robot increases. This will ratify the optimality of maintaining uniform distribution of the robots along the perimeter throughout the execution also in this model.

Lemma 2. *Let $S = \{s_{-t+\tau}, \ldots, s_{-1}, s_0, s_1, \ldots, s_t\}$ be a sequence of $2t$ segments, where robot $R$ resides in $s_0$ at time 0. Then $\forall i \geq 0$, $\mathsf{eud}_i \geq \mathsf{eud}_{i+1}$, and $\forall i \leq 0$, $\mathsf{eud}_i \geq \mathsf{eud}_{i-1}$.*

Proof. First, assume that $i > 0$ (positive indexes). If we show that $\mathsf{pv}_i^j(l) \geq \mathsf{pv}_{i+1}^j(l)$ for all $j, l$, then we are done, since the reward from visiting segment $i$ and segment $i+1$ at time $j$ is the same. We prove that by induction on $l$. As the base case, consider $l = 1$, i.e., we need to show that $\mathsf{pv}_i^j(l) \geq \mathsf{pv}_{i+1}^j(1)$. This is exactly proven in Lemma 1 in [1], based on the fact that the movement of the robots is continuous, therefore in order to get to a segment you must visit the segments in between (the formal proof uses also the conditional probability law).

We now assume correctness for $l' < l$, and prove that $\mathsf{pv}_i^j(l) \geq \mathsf{pv}_{i+1}^j(l)$. Denote the probability that a robot placed at a segment $s_i$ returns to $s_i$ within $r$ time units by $x_i(r)$. In our symmetric environment, for every $i$, $j$ and $l$, $x_i(r) = x_j(r)$. Moreover, $\forall r, x_i(r) \geq x_i(r-1)$. Therefore $\mathsf{pv}_i^j(l)$ can be described as $\sum_{r+u \leq j} \mathsf{pv}_i^u(l) \times x_i(r)$, and similarly $\mathsf{pv}_{i+1}^l = \sum_{r+u \leq j} \mathsf{pv}_{i+1}^u(l) \times x_{i+1}(r)$. By the induction assumption, $\mathsf{pv}_i^j(l-1) \geq \mathsf{pv}_{i+1}^j(l-1)$, and since $x_i(r) = x_{i+1}(r)$, it follows that $\mathsf{pv}_i^j(l) \geq \mathsf{pv}_{i+1}^j(l)$, proving the lemma for positive indexes.

The negative indexes are a reflecting image of the positive indexes, but with $t - \tau$ time units. Since the induction was proven for all $t$ values, the proof for the negative indexes follows directly. □

We now describe Algorithm FindUtilityWProb that finds the expected utility from event detection at segment $s_i$. The algorithm computes the probability of all visits to a segment during $t$ time units. The algorithm, similar to algorithm DetExpectedUtility, is inspired by dynamic programming. As stated previously, the main difference between the algorithms is that DetExpectedUtility considers only the first visit to a segment, where FindUtilityWProb considers *all* visits to a segments and the probability of detecting the event at that time. Figure 1 describes a representation of transition between segments as a Markov chain. This is later translated into constructing gradually a table using the dynamic programming-inspired rules, as described in Algorithm FindUtilityWProb. The time complexity of the algorithm is similar to the time complexity of DetExpectedUtility plus the complexity of substituting the pv values in Equation 1, therefore altogether time complexity of $\mathcal{O}(d^3t^2)$

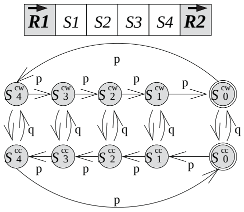

**Figure 1: Representation of the system as a Markov chain along with state transition. The robots are initially placed at the external segments, heading right. State $s_0$ represents the segment currently occupied by a robot.**

```
Algorithm 2 FindUtilityWProb($loc, d, t,$ rwd $= \{$rwd$_1, \ldots,$ rwd$_t\}$, pdt $= \{p_d^1, \ldots, p_d^t\}$)
Create a 0 matrix $M$ of size $(2d+2) \times (t+1)$.
Set $M[0, loc^{cw}] \leftarrow 1$.
Fill all entries in $M$ gradually using the following rules.
for $r \leftarrow 1$ to $t$ do
    for $i \leftarrow 1$ to $d$ (all other states) do
        For each entry $M[r, s_i^{cw}]$ set value to
        $p \cdot M[r-1, s^{cw}_{(i+1 \mod d)}] + (1-p) \cdot M[r-1, s_i^{cc}]$.
        For each entry $M[r, s_i^{cc}]$ set value to
        $p \cdot M[r-1, s^{cc}_{(i-1 \mod d)}] + (1-p) \cdot M[r-1, s_i^{cw}]$.
    end for
    for $s_0^{cw}$ and $s_0^{cc}$ do
        Set $M[r, s_0^{cw}] \leftarrow f \times \{p \cdot M[r-1, s_1^{cw}] + (1-p) \cdot M[r-1, s_0^{cc}]\}$
        Set $M[r, s_0^{cc}] \leftarrow f \times \{p \cdot M[r-1, s_d^{cc}] + (1-p) \cdot M[r-1, s_0^{cw}]\}$
    end for
    pv$_{loc}^i(r) \leftarrow$ polynomial coefficients of $f^i$ from sum of $M[r, s_0^{cw}] + M[r, s_0^{cc}]$, for $1 \le i \le t$.
end for
Return the result obtained by substituting the pv$^j_{loc}(r)$ values in Equation 1.
```

## 6. DETECTING EVOLVING EVENTS FROM A DISTANCE WITH DIFFERENT PROBABILITIES OF DETECTION

In this section, we introduce an additional generalization of the event detection model: Detecting events from a distance, where the probability of detection changes both as the distance from the event changes, and as the event evolves. We assume that the probability of detection monotonically non-increases as the distance from the robot increases.

We assume a robot can sense $D-1$ segments beyond its current segment with some probability greater than 0. The input to the problem is, therefore, a matrix $M_d$ of size $D \times t$, such that entry $[i,j] \in M_d$ is the probability that the event is detected from distance $i$ at time $j$ of the event. In addition, the algorithm receives as input the vector $\mathsf{rwd} = \{\mathsf{rwd}_1, \ldots, \mathsf{rwd}_t\}$ corresponding to the rewards given to the robots for detecting the event at tim $i$, $1 \le i \le t$.

The probability of detecting the event at time $i$ equals the probability that it did not detect it earlier, multiplied by the probability that it is detected at this stage. The probability of being detected at this stage is the probability of being detected in distance $1, \ldots, D$ from the robot for the $1st, 2nd, i'th$ time. Denote the probability of the $l$'th occurrence $(1 \le l \le t)$ of segment $s_i$ in time $j$ in distance $m$ $(1 \le m \le D)$ from some robot by $\mathsf{pv}_i^j(l,m)$. Therefore the equation for determining the expected utility from detecting the event is as follows.

$$\mathsf{eud}_i = \sum_{m=1}^{D} (\mathsf{pv}_i^1(1,m) \cdot M_d[m,1]) \cdot \mathsf{rwd}_1 + \qquad (2)$$

$$\sum_{m=1}^{D} (\mathsf{pv}_i^1(1,m) \cdot (1 - M_d[m,1])) \times \sum_{l=1}^{2} \sum_{m=1}^{D} (\mathsf{pv}_i^2(l,m) \cdot M_d[m,2] \cdot \mathsf{rwd}_2) + \ldots$$

. The total number of components is $d \cdot \sum_{i=1}^{t} i(t-i+1) = \mathcal{O}(Dt^2)$

In order to find the expected utility from detecting an event at a segment $s_{loc}$, it is left to calculate $\mathsf{pv}^j_{loc}(l,m)$, i.e., the probability of $s_{loc}$ being under the sensorial range of some robot at distance $m$ in time $j$ for the $l$'th time. This is done using Algorithm FindRangeReward, where the output is a function of $p$, after all values of $\mathsf{pv}^j_{loc}(l,m)$ are substituted in Equation 2. The time complexity of FindRangeReward is $\mathcal{O}(d(d^2t^2 + Dt^2)) = \mathcal{O}(d^3t^2)$

## 7. OPTIMALITY OF THE PATROL ALGORITHM FOR DIFFERENT ADVERSARIAL MODELS

We have presented algorithms for determining the expected utility from event detection at every segment $s_i$, $1 \le i \le N$ in various event models. The output of these algorithms is the expected utility as a function of $p$, i.e., it depends on the choice of the patrol algorithm. Previous work has shown that the optimality of the patrol algorithm depends on the level of adversarial knowledge. Specifically, it was shown that the algorithm MaxiMin [2] maximizes the minimal probability of penetration detection, hence is optimal for strong adversarial models. On the other hand, it was proven that the simple deterministic algorithm in which $p = 1$ maximizes the *expected* probability of penetration detection throughout the perimeter, thus optimal for a weak adversarial model in which the adversary chooses its penetration spot at random with uniform distribution. We would therefore like to examine the optimality of the patrol algorithm in these two extreme adversarial environments: extremely weak and extremely strong.

### 7.1 Basic event model - Implication of reward function on optimality of patrol

**Weak adversarial model:**

In the weak adversarial model, we assume the events have a complete random nature, i.e., an event can happen at any segment with uniform probability. In this case we prove that

**Algorithm 3** $\mathsf{FindRangeReward}(loc, d, t, L, \mathsf{rwd} = \{\mathsf{rwd}_1, \ldots, \mathsf{rwd}_t\}, M_d)$

```
Create matrix $X$ of size $(2d+2) \times (t+1)$, initialized with 0s.
Set $X[0, loc^{cw}] \leftarrow 1$.
Initialize the vector $Res$ of size $t$ with 0s.
Fill all entries in $X$ gradually using the following rules.
for $r \leftarrow 1$ to $t$ do
    for each entry $X[r, s_i^{cw}]$, $1 \leq i \leq d$ do
        Set $u \leftarrow p \cdot X[r-1, s_{i+1 \mod d}^{cw}] + (1-p) \cdot X[r-1, s_i^{cc}]$
        if $i + L \geq d$ then
            $u \leftarrow u \times f_{d-i}$
            $Res[t] \leftarrow Res[t] + u$
        end if
        Set $X[r, s_i^{cw}] \leftarrow u$
    end for
    for each entry $X[r, s_i^{cc}]$, $1 \leq i \leq d$ do
        Set $u \leftarrow p \cdot X[r-1, s_{i+1 \mod d}^{cw}] + (1-p) \cdot X[r-1, s_i^{cc}]$.
        if $i - L \leq 0$ then
            $u \leftarrow u \times f_i$
            $Res[t] \leftarrow Res[t] + u$
        end if
        Set $X[r, s_i^{cc}] \leftarrow u$
    end for
    $\mathsf{pv}_{loc}^j(l, m) \leftarrow$ polynomial coefficient of $f_m^l$ of $Res[j]$, for all $1 \leq l \leq t$, $0 \leq m \leq L$ (while substituting all other $f_{m'}^l$ with 1, $m' \neq m$ in the equation).
end for
Return the result obtained by substituting the $\mathsf{pv}_{loc}^j(l, m)$ values in Equation 2.
```

the deterministic algorithm in which $p = 1$ is optimal when using any reward function (not necessarily only a monotonically non-decreasing function).

Theorem 3. *The optimal patrol algorithm that maximizes the total expected utility of the robots is the deterministic algorithm ($p = 1$) for any possible reward function.*

Proof. In [2] it was proven that the expected probability of detection, corresponding to the sum of expected probabilities of first arrival at all segments during $t$ time units, is maximized for $p = 1$, i.e., the total expected probability of arrival is $t/d$ $\forall t$. Therefore we can conclude that the total expected probability of first arrival at time $t'$, for every $t' \leq t$ is highest when $p = 1$ and equals $t'/d$, hence the maximal expected utility can be of value $1/d(\mathsf{rwd}_1 + \mathsf{rwd}_2 + \ldots + \mathsf{rwd}_t)$. The total expected utility when $p = 1$ is defined as $1/d \sum_{i=1}^{d} \sum_{j=1}^{t} \mathsf{pv}_i^j \cdot \mathsf{rwd}_j = 1 \cdot \mathsf{rwd}_j$, which is exactly the maximal value for every reward function. □

**Strong adversarial model:**

In case we model the worst-case behavior of the system, similar to facing a full-knowledge adversary that uses its knowledge of the patrol to create an event on the weakest spot of the patrol, we use the MaxiMin algorithm described in [1]. That algorithm calculates the maximal point in the lower envelope (integral intersection) of the $d$ curves representing the functions of probability of penetration detection. It was shown there that this point is either an intersection of two or more curves, or a local maxima.

The same algorithm can be used here as well for determining the point $p$ that corresponds to the maximin point. We show that in some cases the optimal $p$ remains the same for any reward function that is monotonically non-increasing. Specifically, this happens when the duration of the event is very short relative to the distance between the robots. The significance of this result is in showing the limitation of the system in such cases—no matter how the reward function acts, it cannot influence the final outcome.

Theorem 4. *The optimal patrol remains indifferent for all monotonically non-increasing reward function if $t = \lfloor \frac{d}{2} \rfloor + 1$ and $t > 2$.*

Proof. We will show that in case $t = \lfloor \frac{d}{2} \rfloor + 1$, the point that maximizes the minimal expected utility is obtained in the local maxima of the curve corresponding to $\mathsf{eud}_{t+1}$. We then show that this point remains smaller than all other curves for every non-increasing reward function. By Lemma 1 it follows that $\mathsf{eud}_{t+1} \leq \mathsf{eud}_{t+i} \forall i > 1$. Similarly, $\mathsf{eud}_t \leq \mathsf{eud}_j \forall j < i$, since the expected utility in all $d$ segments correlate to the expected visit of only one robot. Therefore it suffices to show that $\mathsf{eud}_{t+1}$ is smaller than $\mathsf{eud}_t$ in the point of local maxima of $\mathsf{eud}_{t+1}$.

First, assume that $d$ is odd. The probability of first arriving at segment $s_{t+1}$ is exactly $(1-p)p^{t-1}$, hence $\mathsf{eud}_{t+1} = \mathsf{rwd}_t(1-p)p^{t-1}$. To find the local maxima, we derive this function and compare to 0, i.e., $\frac{d\mathsf{eud}_{t+1}}{dp} = (t-1)\mathsf{rwd}_t p^{t-2} - t\mathsf{rwd}_t p^{t-1} = 0$, resulting in a local maxima when $p = \frac{t-1}{t}$, regardless of the value of $\mathsf{rwd}_{t+1}$. Now it is left to show that when $p = \frac{t-1}{t}$, $\mathsf{eud}_{t+1} = \mathsf{rwd}_t(1 - \frac{t-1}{t})(\frac{t-1}{t})^{t-1} \leq \mathsf{eud}_t = \mathsf{rwd}_t p^t = \mathsf{rwd}_t(\frac{t-1}{t})^t$, which is true for every $t > 1$.

If $d$ is even, then the expected utility from detecting an event in segment $s_{t+1}$ is $\mathsf{eud}_{t+1} = (1-p)p^{t-2}\mathsf{rwd}_{t-1}$. In order to use Lemma 1 we need first to show that $\mathsf{eud}_{t-1} \geq \mathsf{eud}_t$. $\mathsf{eud}_t = \mathsf{rwd}_t p^{t-1} \leq \mathsf{rwd}_{t-1} p^{t-1} = \mathsf{eud}_{t-1}$, since we assume that $\mathsf{rwd}$ is monotonically non-increasing. It is left to show that in the point of local maxima $\mathsf{eud}_t \geq \mathsf{eud}t + 1$. Similar to the previous derivative, the local maxima is obtained here when $p = \frac{t-2}{t-1}$. Obtaining the value of $\mathsf{eud}_t$ and $\mathsf{eud}_{t+1}$ at this point we get that this is true for all $t > 2$. □

Figure 2 demonstrates how different rewards correspond to different values of $p$. The curves represent the expected reward of each of the $d$ segments, and the bold line marks the lower envelope, i.e., the integral intersection of all curves. The maximal point in the lower envelope is the value that maximizes the minimal expected utility of the robots. It is interesting to see that as we give high weight to the first several time units relative to the last ones, the optimal probability $p$ grows closer to 1. However, when given high weight to too few time units (here $\mathsf{rwd}_1 = 9$ and all other $\mathsf{rwd}_i = 1$) or too many of them (in the example $\mathsf{rwd}_i = 9, 1 \leq i \leq 8$ and $\mathsf{rwd}_9 = 1$) the result is closer to the uniform-reward case. In the example illustrated in Figure 2, $d = 12$ and $t = 9$. In the uniform reward case, the maximin point is obtained in $p = 0.7741$, when $\mathsf{rwd}_i = 9, 1 \leq i \leq 5$ and $\mathsf{rwd}_j = 1, 6 \leq j \leq 9$ the optimal $p$ equals 0.925, and when $\mathsf{rwd}_i = 9, 1 \leq i \leq 8$ and $\mathsf{rwd}_9 = 1$, the optimal $p$ is 0.8577.

## 7.2 Imperfect detection and weak adversarial model

When the nature of the event correlates to different probabilities of detection as the event evolves, we would again like to examine how the expected utility functions correlate to different adversarial models. In a strong adversarial model,

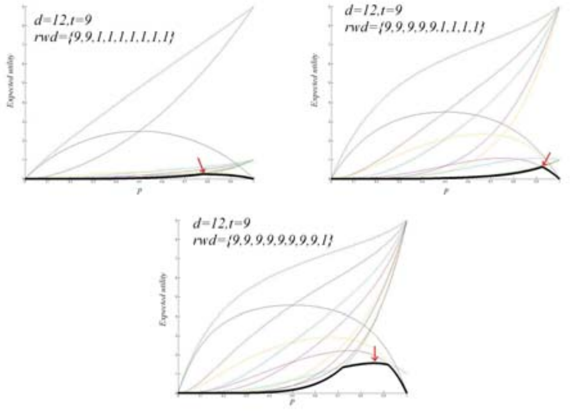

**Figure 2: Examples of the influence of different rewards on the lower envelope and the MaxiMin point. The curves represent the expected utility as a function of $p$, the bold line draws the lower envelope of the curves, and the arrows points to the maximal point in the lower envelope (the MaxiMin value).**

in which our goal is to maximize the minimal expected utility from event detection, it is possible to use the MaxiMin algorithm, as described in [1].

A more interesting case is the weak, random, adversarial model. In Theorem 3 it was shown that in the case of perfect sensing the simple deterministic algorithm ($p = 1$) is optimal regardless of the reward function. The rational behind the optimality proof of the deterministic algorithm lies in the fact that it is not worthwhile to go back and revisit segments.

However, in case the probability of detecting the event is imperfect this argument does not necessarily hold, i.e., revisiting a segment does have added value. Moreover, this is more evident in case the probability of detecting the event grows as the event evolves, i.e., $p_d^{i+1} > p_d^i$.

In the following, we show the surprising result that even if $p_d^i < 1$, if the event occurs at a random location with uniform distribution, then it is still best to patrol deterministically around the perimeter. Moreover, we strengthen our result by showing that even if the robot makes a post analysis of its decision to go straight or turn around, it will also decide to keep on going straight.

Lemma 5. *Assume an event occurs at random with uniform distribution in any possible segment of the d unoccupied segments between two robots. Therefore the gain to the utility from event detection by the robots from revisiting a segment is smaller than the gain from initially visiting a new segment, for every $p_d^j > 0$.*

Proof. The gain from revisiting a segment at time $j$, denoted by $G_r^j$ is the probability that the robot did not detect the event during its first visit multiplied by the probability that the event indeed occurs at that segment, multiplied by the reward from detecting the event at time $j$. Formally, $G_r^j = \frac{1}{d} \times (1 - p_d^l) \times p_d^j \times \mathsf{rwd}_j$ for some $l < j$. On the other hand, the gain from initially visiting a new segment at time $j$, denoted by $G_i^j$ is $\frac{1}{d} \times p_d^j \times \mathsf{rwd}_j$. Since $p_d^l > 0$, it follows that $G_r^j < G_i^j$. □

Theorem 6 follows directly from Lemma 5.

Theorem 6. *In case the probability of detecting an event increases as the event evolves, and the event may occur at any spot at random with uniform distribution, the deterministic algorithm maximizes the expected utility throughout the perimeter for all possible values of* pdt.

We strengthen this result by showing that it is beneficial for the robot to keep visiting new segments in case the events location is determined at random with uniform distribution (with probability $1/d$) even if the robot calculates its benefit post factum, i.e., after visiting a segment. Denote the probability that the event indeed occurred in segment $s_i$ by $PN_i$, and the probability that the robot visited $s_i$ without detecting it by $ND_i$. Therefore, by conditional probability law, if $ND_i > 0$, $P(PN_i \mid ND_i) =$

$$\frac{PN_i \bigcap ND_i}{ND_i} = \frac{1/d(1-p_d^j)}{(d-1)/d + 1/d(1-p_d^j)} = \frac{1-p_d^j}{d-p_d^j}$$

On the other hand, the probability that the event occurred in segment $s_{i+1}$ knowing that the robot did not detect it in segment $s_i$ is

$$\frac{1 - \frac{1-p_d^j}{d-p_d^j}}{d-1} = \frac{1}{d-p_d^j} > \frac{1-p_d}{d-p_d}$$

In other words, the probability of revealing new information in visiting a new segment is greater than the probability of revealing new information from revisiting a segment that was already visited at least once, even after knowing that the event was not detected in the revisited segment. The intuition is that by visiting a new segment, the probability of event detection grows by $p_d^j$, where if the robots revisits a segment, it carries along with it the probability of arriving there again, multiplied by $p_d^j$. Since the probability of arriving again is smaller than 1, the gain from revisiting a segment is smaller.

## 7.3 Detection from a distance - determinism not necessarily optimal

An interesting result in the weak adversarial model is seen in the case where robots can detect an event from ahead with some probability. Here, the optimality of the deterministic algorithm is no longer absolute. The logic behind this statement is that if the cost of turning around (in number of cycles) is smaller than the profit of what the robots gain (number of new visible segments), then it might be worthwhile to turn around. Therefore it is possible to prove optimality of the deterministic algorithm for maximizing the expected utility only if $L \leq \tau$, where $\tau$ is the time it takes the robot to turn around.

As an example that demonstrates the fact that the deterministic algorithm is no longer optimal if $L > \tau$, consider the case in which $L = 3$, $d = 4$ , $t = 2$ and $\tau = 1$. Assume that the reward function is uniform and that the probability of detection is uniform and equals 1.The maximal expected utility is 0.85 and it is obtained for $p = 0.5$, whereas if $p = 1$ (deterministic algorithm) the expected utility is only 0.8.

Theorem 7. *If the robots can sense ahead with some probability that can change as the event evolves, the deterministic algorithm guarantees the maximal expected utility for random-uniform events if $L - 1 \leq \tau$.*

The proof of this theorem resembles the proof of Theorem 6 and is omitted due to lack of space.

## 8. ON DIFFERENT REWARDS IN DIFFERENT LOCATIONS - A DISCUSSION

Previous work in adversarial planning using game theoretic approaches such as the work by Amigoni et al. [6] and Pita et al. [12] are based on the fact that the adversary has different utilities that depends on where it decides to act. On the other hand, the robots have their own utility, and their solutions try to optimize the utility of the robots. Their solutions are exponential, thus they provide heuristics and approximations in order to be able to provide reasonable solutions in polynomial time. To this point, solutions that provide *optimal* patrols computed in polynomial time such as the work by Agmon et al. [2] do not take into account possible utilities of the robots nor of the adversary. This work, therefore, sets important grounds for considering the utility of the robots.

Consider the case of heterogenous environments, in which the perimeter has areas that are more vulnerable for a possibility of events, for example areas that are more likely to be penetrated through or fields that might catch fire more easily. If these locations are static, then one might decide to place cameras for surveillance. However, if this is impossible, or the vulnerability might change its location in time, another solution should be found. In this case, we suggest using a variation of the basic solution presented in this paper: adding rewards to the robots for early detection of events at different locations. It is possible to define different rewards to different segments. However, this breaks the symmetry of the system, thus it requires to consider all $N$ segments in the calculation of the expected utility (rather than $d = N/k$ segments). We intend to further investigate this aspect and try to improve the time complexity in future work.

## 9. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a new approach for handling events in multi-robot perimeter patrol in adversarial environments. In this new approach, the detection of an event is associated with reward that depends on the relative time of detection. We described polynomial-time algorithms for determining the expected utility from detecting the event at each point along the perimeter in three different models. (1) Utility from detection is correlated to the time of detection. (2) The events' progression correlates to both different rewards and to growing probability of detection. (3) The event can be detected from distance, where the probability of detection depends both on the distance from the robot and on the current state of the event. We show how these models influence on the choice of optimal patrol algorithm in different adversarial settings, namely, the strong adversarial model (equivalent to modeling the worst case behavior of the system) and the weak adversarial model (correlates to modeling the system when no particular threat is predicted).

Several points are of our interest for future work. First and foremost, we plan on further developing solutions for heterogenous environment by exploiting the framework of rewards presented in this paper. We would also like to examine the problem of patrol in various graph environments (trees, grids, general graphs) rather than in linear graphs, and handling events in patrol in different adversarial environments (rather than facing only strong and weak adversaries).

## 10. ACKNOWLEDGEMENTS

I would like to thank Prof. Sarit Kraus from BIU and Prof. Manuela Veloso from CMU for the useful conversations and important discussions.

## 11. REFERENCES

[1] N. Agmon, S. Kraus, and G. A. Kaminka. Multi-robot perimeter patrol in adversarial settings. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2008.

[2] N. Agmon, V. Sadov, S. Kraus, and G. A. Kaminka. The impact of adversarial knowledge on adversarial planning in perimeter patrol. In *Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-08)*, 2008.

[3] M. Ahmadi and P. Stone. A multi-robot system for continuous area sweeping tasks. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2006.

[4] F. Amigoni, N. Gatti, and A. Ippedico. Game-theoretic approach to determining efficient patrolling strategies for mobile robots. In *Proceedings of Agent Intelligent Technologies (IAT-08)*, 2008.

[5] N. Basilico, N. Gatti, and F. Amigoni. Extending algorithms for mobile robot patrolling in the presence of adversaries to more realistic settings. In *Proceedings of IAT*, pages 565–572, 2009.

[6] N. Basilico, N. Gatti, and F. Amigoni. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *AAMAS*, pages 57–64, 2009.

[7] Y. Chevaleyre. Theoretical analysis of the multi-agent patrolling problem. In *Proceedings of Agent Intelligent Technologies (IAT-04)*, 2004.

[8] S. Koenig, R. Goodwin, and R. G. Simmons. Robot navigation with markov models: A framework for path planning and learning with limited computational resources. In *Reasoning with Uncertainty in Robotics*, pages 322–337, 1995.

[9] N. A. Lynch. *Distributed Algorithms.* Morgan Kaufmann, 1996.

[10] P. Paruchuri, J. P. Pearce, M. Tambe, F. Ordonez, and S. Kraus. An efficient heuristic approach for security against multiple adversaries. In *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-08)*, 2007.

[11] P. Paruchuri, M. Tambe, F. Ordonez, and S. Kraus. Security in multiagent systems by policy randomization. In *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-07)*, 2007.

[12] J. Pita, M. Jain, F. Ordonez, , M. Tambe, S. Kraus, and R. Magorii-Cohen. Effective solutions for real-world stackelberg games: When agents must deal with human uncertainties. In *Proceedings of the Eighth International Conference on Autonomous Agents and Multiagent Systems (AAMAS-09)*, 2009.

[13] T. Sak, J. Wainer, and S. K. Goldenstein. Probabilistic multiagent patrolling. In *Proc. of the 19th Brazilian Symposium on Artificial Intelligence (SBIA-08)*, pages 124–133, 2008.