The Impact of Adversarial Knowledge on Adversarial Planning in Perimeter Patrol

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

ABSTRACT

This paper considers the problem of multi-robot patrolling around a closed area, in the presence of an adversary trying to penetrate the area. Previous work on planning in similar adversarial environments addressed worst-case settings, in which the adversary has full knowledge of the defending robots. It was shown that non deterministic algorithms may be effectively used to maximize the chances of blocking such a full-knowledge opponent, and such algorithms guarantee a "lower bound" to the performance of the team. However, an open question remains as to the impact of the knowledge of the opponent on the performance of the robots. This paper explores this question in depth and provides theoretical results, supported by extensive experiments with 68 human subjects concerning the compatibility of algorithms to the extent of information possessed by the subjects. First, we analytically examine the case of a zero-knowledge opponent—a different extreme—and show that surprisingly, this seemingly best-case scenario (from the point of view of defending robots) is optimally addressed by a deterministic, non-randomizing patrol. Moreover, we show empirically that an optimal algorithm for the full-knowledge opponent fails miserably in this case. We then address the case in which the adversary gained partial information, propose the Combine algorithm that maximizes the expected probability of penetration detection along with minimizing the deviation between the probabilities of penetration detection along the perimeter, and support the performance of this algorithm in the experiments.

Categories and Subject Descriptors

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

General Terms

Algorithms, Experimentation, Security

Keywords

Multi-Robotics, Adversarial/game domains, Formal models of multi-robot plans, Multi-robot path planning

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

Cite as: Title, Author(s), *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril, Portugal, pp. XXX-XXX.

1. INTRODUCTION

This work considers the problem of multi-robot patrolling around a perimeter in different adversarial environments. The multi-robot patrol task requires a team of robots to jointly repeatedly visit some target area in order to monitor change in state [6, 2]. When working in adversarial environments, the robots' task is to detect changes in state that are controlled by an adversary. In our case, we assume the adversary is trying to penetrate through the perimeter into (or out of) the area. This problem is applicable in many security applications [1, 11, 10].

In this paper, we concentrate on the information obtained by the adversary, and its impact on the choice of patrol algorithm best suited to the task. Previous work usually assumed that the adversary has full knowledge of the environment, and uses this information in order to maximize its utility (e.g. [10, 1]). This assumption is, from the patrolling robots' point of view, the *worst case scenario* they face. These studies have shown that the use of non-deterministic components in the patrol algorithm is advantageous in such cases. These algorithms, that are designed to work in such worst case environments, guarantee some lower bound criteria on the performance of the robots, i.e., on their ability to block the adversary. The question that remained unaddressed is whether the algorithms that are optimal in the worst case scenario are also good in other adversarial settings, and if not - which algorithms are. We address this question in this paper.

We assume a general robotic movement model, in which the robots have directionality associated with their movement. Since the patrol path is cyclic, in each time cycle they can either move forward (with probability p, $0 \leq p \leq 1$), or turn around (with probability $(1-p)$. The action of turning around is costly in time, and takes τ time units. We define the *patrol scheme* as the value p of the patrol, and the current location of the robots.

If the adversary has full knowledge of the patrol scheme, then it will use this information in order to choose a penetration spot such that it will less likely be detected by the patrolling robots. Agmon et. al. [1] studied this case and suggested the algorithm MaxiMin for finding the probability p according to which the robots should switch direction in each time cycle. The probability p characterizing the movement of the robots is optimal in the sense that it increases as much as possible the probability of detecting penetrations in their weakest spot.

On the other hand, if the adversary has no knowledge of the patrol scheme, then the patrol scheme might be dif-

ferent. In this case, we assume the adversary chooses at random with uniform probability its penetration spot. The robots, on the other hand, wish to maximize the expected probability of penetration detection throughout the perimeter. We show the surprising result that as opposed to the sophisticated algorithms used when the adversary has full knowledge of the patrol scheme, here the optimal algorithm for the robots is the simple deterministic patrol in which the robots simply follow their patrol path without ever turning around. This results holds even if the cost of turning around is extremely low: one time cycle ($\tau = 1$).

If the adversary has even very short time to gather knowledge of the patrol scheme, it might easily deduce the deterministic patrol scheme and manage to penetrate successfully. For such cases we propose a new algorithm, Combine, which maximizes the expected probability of penetration detection throughout the perimeter along with minimizing the standard deviation between the probabilities of penetration detection throughout the perimeter.

In order to evaluate the behavior of people in this scenario with different amount of information, we have created the *Penetration Detection Game* (PenDet game). In this game, simulated robots execute different patrol schemes while patrolling around a perimeter, trying to detect penetrations. The player plays the role of the adversary, hence she is required to choose a section through, to her understanding, she will most likely penetrate without being detected . The game has three stages, where from stage to another the player gets more time to examine the patrolling robots, and by that attain more information about the patrol scheme. The robots execute the three different algorithms: MaxiMin, Combine and the deterministic algorithm.

Results from extensive experiments with 68 human subjects proved that the Combine algorithm performs best in environments in which the subjects were exposed to the patrol information for a short period of time. In case the subjects were given no information about the environment, then the deterministic algorithm performed substantially better than the other algorithms, even though the choices of penetration spots were not uniform. The MaxiMin algorithm failed miserably in this case. The MaxiMin algorithm, however, substantially outperformed the other algorithms in case the subject were given a long period of time in order to evaluate and study the system.

2. RELATED WORK

Systems of multiple robots patrolling in adversarial environments has been studied in various approaches and contexts, from theoretical to empirical solutions. A theme running through studies of patrol in adversarial environments is emphasis on a full-knowledge adversary, as a worst case scenario ([11, 10, 1].

Agmon et. al. [1] studied the problem of perimeter patrol with the existence of an adversary. They assume a *strong* adversarial model, in which the adversary has full knowledge of the patrol scheme. In this case, the adversary can choose to penetrate through the segment in which it has minimal probability of being detected. They prove that a non-deterministic algorithm is advantageous and describe the algorithm MaxiMin for finding the probability p that characterizes the movement of the robots such that the minimal probability of penetration detection throughout the perimeter is maximized. In our work we consider other adversarial models, and show empirically that this algorithm is not suitable in those models.

Recent work by Paruchuri et. al. [11, 10] is closely related to our work. Similar to our assumptions, they assume that their agents work in an adversarial environment in which the adversary can exploit any predictable behavior of the agents. They use policy randomization for the agents behavior in order to maximize their rewards. They assume the adversary has *full knowledge* of the patrolling agents. Paruchuri et. al. further study ([10]) this problem in case the adversarial model is unknown to the agents, although still the adversary has full knowledge of the patrol scheme. They again provide heuristic algorithms for optimal strategy selection by the agents. In our work, we discuss different adversarial models determined by the extent of information revealed to the adversary. Also, we assume the robots have partial information about the adversary (specifically, they know the time it takes him to penetrate).

Most of the work in the area of multi-robot patrol did not assume the existence of an adversary, and concentrated on frequency criteria satisfied by the patrolling robots (e.g. [6, 5, 12, 3]). In addition, these studies concentrate on area patrol rather than perimeter patrol.

Carmel and Markovitch [4] consider games in which the opponent does not necessarily choose the worst option from the player's point of view. They develop the M* algorithm, which is a generalization of the minimax algorithm that can use an opponent model. They evaluated their algorithm in the domain of checkers, and have shown that their algorithm is advantageous compared to the minimax algorithm. In our work, we concentrate on adversarial models that differ in the information available to the adversary on the patrol scheme.

Theoretical work that is based on stochastic processes is the *predator-prey* [7] or *pursuit evasion* [13] problems. Here, the predator is trying to catch the prey in a graph environment. The predator has no knowledge of the location of the pray, and thus their movement is similar to a random walk. In our work, on the other hand, we suggest realistic robotic models in which the movement is correlated to the movement of a robot, and realistic adversarial models.

3. DIFFERENT ADVERSARIAL MODELS

3.1 Preliminaries

Following, we describe the robotic and environmental model used throughout our work.

We consider a team of k coordinated robots, patrolling around a perimeter. The perimeter is divided into N sections, such that each robot passes through one section per one time unit while monitoring it. Let ppd be the probability of penetration detection. The robots have directionality associated with their movement, therefore in each time unit each robot can either go straight or turn around. In this paper we consider the most realistic movement model in which turning around costs the system $\tau \geq 1$ extra cycles, i.e., if the robot decides to turn around, then it remains in the same segment during those τ cycles. An example for this kind of robots are the differential drive robots commonly used in research labs. This is a generalized movement model that was suggested in [1], in which they assume that $\tau = 1$.

If given a non-deterministic patrol algorithm, the robots continue their current course with probability p, and turn around (while staying in the same section for τ time units)

with probability $1-p$. In a deterministic patrol algorithm $p = 1$, i.e., the robots never switch their direction and continue their movement with no non-deterministic components. Note that once the robots decided to switch their direction, they will continue turning around for τ time units.

We follow [1] in assuming that the robots are coordinated, i.e., if they switch directions they do it simultaneously. Moreover, the robots are placed uniformly with $d = N/k - 1$ unoccupied segments between every two adjacent robots. The motivation for these assumptions as given in [1] refer to the full knowledge adversarial model, and we verify the justification of maintaining these assumptions also in the zero-knowledge adversarial model in Subsection 3.3. Note that these assumptions were proven to be optimal also for maintaining frequency constraints in a patrol[6, 3].

We assume the existence of an adversary that has to decide through which segment to penetrate. The time it takes it to penetrate is not instantaneous, and lasts t time cycles. This value is known to the patrolling robots. We consider t values between the boundaries $\lfloor \frac{d+\tau}{2} \rfloor \leq t \leq d-1$. In case $t < \lfloor \frac{d+\tau}{2} \rfloor$, even a non-deterministic algorithm cannot guarantee that the ppd in all segments will be greater than 0. On the other hand, if $t \geq d$, then the deterministic algorithm will detect all penetrations.

3.2 Compatibility of algorithms with adversarial models

When a team of robots patrols in an adversarial environment, the team must choose the patrol algorithm that will maximize their probability of penetration detection (ppd). The quality of the patrol algorithm, i.e., its guaranteed penetration detection, depends heavily on the actions of the adversary.

If the adversary chooses the weakest spot of the patrol, i.e., the segment in which it will less likely be detected, then the algorithm should strengthen the ppd in that point as much as possible. On the other hand, if the adversary has no knowledge of the patrol scheme, it cannot choose such a weak spot, hence the patrol algorithm should maximize the expected ppd throughout the perimeter. Therefore the *knowledge obtained by the adversary* on the patrol scheme is critical for the choice of the patrol algorithm.

In [1], the authors prove that a non-deterministic algorithm is advantageous in case the adversary has full knowledge of the patrol scheme. The probability p that characterizes the movement of the robots increases as much as possible the ppd in the weakest segment, i.e., maximizes the minimal ppd. However, when using this probability p, the ppd in other segments might decrease compared to other p values, specifically $p = 1$ (deterministic algorithm).

This demonstrates the tradeoff between deterministic and non-deterministic algorithms for patrol. Generally, when the robots never turn around, then more segments are visited by the robots since no time is wasted on turning. However, in this case the algorithm is fully predictable, therefore more vulnerable to exploitation by a knowledgeable adversary. Robots executing a random algorithm, on the other hand, visit less segments compared to a deterministic algorithm but visit them more times, which creates unpredictability, hence more difficult to exploit.

Figure 1 brings an example of the tradeoff between raising the minimal ppd and lowering the other ppd values of the segments. In this example, there are 12 unoccupied segments, and $t = 11$. The ppd is 1 in the first 11 segments if the algorithm is deterministic, and 0 in the last. If using the maximin patrol algorithm, then the minimal ppd increases from 0 to 0.4287 in the last segment. On the other hand, the average ppd if using the MaxiMin algorithm is 0.575, while the average ppd using the deterministic algorithm is 0.9166.

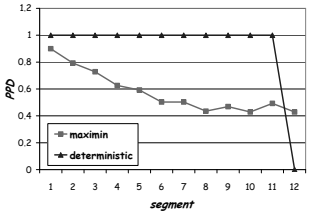

Figure 1: The ppd values in 12 segments, in case $t = 11$ and $\tau = 1$, illustrating the tradeoff between increasing the minimal ppd value and decreasing the ppd values in other segments.

Therefore the MaxiMin algorithm is not necessarily suitable when the adversary does not have full knowledge. If the adversary does have full knowledge, then the deterministic algorithm is *necessarily not* suitable, as the adversary will choose to penetrate through the last segment.

We prove in Subsection 3.3 the surprising result that the optimal patrol scheme in case the adversary has no knowledge of the patrol scheme, and chooses its penetration spot at random with uniform probability, is the simple, non-sophisticated deterministic algorithm, i.e., $p = 1$.

However, assuming the adversary has no information whatsoever on the patrolling robots is again, in many cases, not realistic. It might draw information even from the position of the robot, and change its course of action accordingly. Therefore we present the Combine algorithm, that addresses the case in which the adversary have partial knowledge of the system, i.e., it attained some information about the system, yet not a complete picture of the scheme.

3.3 Zero knowledge

Consider the extreme case in which the adversary has no prior knowledge of the patrol scheme and thus it will choose its penetration spot at random with uniform distribution. Therefore an optimal algorithm for this case will maximize the *expected* ppd throughout the perimeter. We discuss herein this case. We first establish some facts concerning the optimality of placement of the robots along the perimeter and their synchronization. Then we show that, surprisingly, the best algorithm for this case is the deterministic algorithm, i.e., $p = 1$. The logical explanation behind this result is that if using a deterministic algorithm, the ppd in t segments is 1 and 0 otherwise. If the robots switch their direction, then they might increase the ppd in some segments from 0, but they pay first in the time it takes them to turn around, and second it decreases the ppd from 1 in the other cells, so the overall benefit is inferior.

Following, we show that the requirement that the robots will be coordinated and preserve uniform distance of $d = N/k$ between every two consecutive robots is required also if we wish to maximize the *expected* ppd throughout the perimeter, similar to the requirement with the existence of a strong adversary.

The expected ppd in the system of N segments given k

robots and penetration is done at random with uniform distribution, is the average of ppd values in each segment. Denote the probability that a penetrator will be detected in segment s_i by robot R_j during t time units by ppd_i^j. This is the probability of R_j's first visit to s_i during this time. Therefore the ppd in segment s_i is the sum of probabilities of the first visit by some robot $R_j, 1 \leq j \leq k$, i.e., $\mathsf{ppd}_i = \sum_{j=1}^{k} \mathsf{ppd}_i^k$. The expected ppd is, then, $\mathbb{E}(\mathsf{ppd}) = \sum_{i=1}^{N} \sum_{j=1}^{k} \mathsf{ppd}_i^k$. However, $\sum_{j=1}^{k} \mathsf{ppd}_i^k$ might be greater than 1, therefore the overall expected ppd is $\mathbb{E}(\mathsf{ppd}) = \sum_{i=1}^{N} \min\{1, \sum_{j=1}^{k} \mathsf{ppd}_i^k\}$. If always $\sum_{j=1}^{k} \mathsf{ppd}_i^k < 1$, then the location of the robots is irrelevant to the value of $\mathbb{E}(\mathsf{ppd})$.

However, this is not the case - consider for example the case in which the robots are located in adjacent segments. Agmon et. al. [1] have shown that as the distance between a robot R_i and a segment s_j increases, the probability of arriving to it during t time units decreases, i.e., ppd_j^i decreases. In order to maximize the expected ppd, it is necessary to place the robots such that $\forall s_i, 1 \leq i \leq N, \sum_{j=1}^{k} \mathsf{ppd}_i^k \leq 1$. Since the patrol path is circular, decreasing the distance between two robots R_a and R_b, necessarily increases the distance between two robots R_x and R_y. Therefore the optimal placement of the robots is with uniform distance between them, i.e., $d = N/k$. Guaranteeing that this optimality measure is maintained is by keeping the robots synchronized, i.e., if they choose to switch directions they should do it simultaneously.

Next, we show that surprisingly, the optimal patrol scheme for this case is the deterministic patrol algorithm ($p = 1$), for all $\tau \geq 1$. Denote the ppd in segment s_j by R_0 after switching its direction r times by $\mathsf{ppd}_j^0(r)$.

LEMMA 1. *Consider a sequence of $2d$ segments with one robot R_0 in the mid segment at time 0. If the robots switched directions $r \geq 1$ times during t cycles of execution, then $\sum_{j=1}^{2d} \mathsf{ppd}_j^0(r) < t$ for every $\tau \geq 1$.*

PROOF. We prove, by induction on r, that for every $r \geq 1$, $\sum_{j=1}^{2d} \mathsf{ppd}_j^0(r) < \sum_{j=1}^{2d} \mathsf{ppd}_j^0(r-1)$. Note that the sum of ppd for $p = 1$ ($r = 0$) is exactly t, hence by proving the induction we prove the lemma.

As the base case, consider $r = 1$ and $\tau = 1$. Note that since $r > 1$ then necessarily $p < 1$. During t cycles, R_0 can visit and monitor at most t segments. Therefore we should consider $t - 1$ segments from both sides of R_0 (one cycle is "wasted" on turning around). We are interested only in the probability of *first* visit at a segment in order to determine the ppd in that segment. Without loss of generality, we assume R_0 is headed to the right.

In order to prove the lemma, it is enough to show that the sum of expected ppd if $p < 1$ is less than the sum of ppd in case $p = 1$, which is t. For that we check the addition of ppd to the segments to the left of R_0 in case $p < 1$, and compare this addition to the reduction of ppd to the segments to the right of R_0 (from 1 in each segment if $p = 1$). Denote the segment in which R_0 initially resides on by s_0, the segments to its right in ascending order $(s_1, \ldots, s_t)$ and segments to its left by descending order $(s_{-1}, s_{-2}, \ldots, s_{-t+1})$. First of all, the biggest decrease factor is to segment s_t, from 1 to 0. Next, the ppd in each segment $s_i, 1 \leq i \leq t-1$, to the right of R_0 decreased from 1 to p^i. Therefore the sum of reduction is $1 + (1-p) + (1-p^2) + \ldots + (1-p^{t-1}) = t - \sum_{i=1}^{t-1} p^i$. On the other hand, the addition of ppd to the segments to the left of R_0 is as follows. The ppd in segment s_{-1} is $(1-p)p + p(1-p)pp + pp(1-p)pppp + \ldots + p^{t/2-1}(1-p)p^{t/2}$. Similarly, the ppd in segment s_{-2} is $(1-p)pp + p(1-p)ppp + \ldots + p^{t/2-3}(1-p)p^{t/2+2}$. Generally, the sum of all ppd in the segments to the left of R_0 is $(1-p)p + (1-p)p^2 + 2(1-p)p^3 + \ldots + \frac{t}{2}(1-p)p^{t-1} = (1-p)[p + p^2 + 2p^3 + 2p^4 + 3p^5 + 3p^6 + \ldots + t/2p^{t-2} + t/2p^{t-1}$. For every $t \geq 2$, $t - \sum_{i=1}^{t-1} p^i$ is greater than the above expression (for $t = 1$, $r = 1$, this is straightforward, as the ppd in all segments but s_0 is 0).

In order to prove the lemma for a general r, we divide the sequence into two: the sequence to the right of R_0 and to the left of R_0. For every $1 < j \leq r$, let $\sum_{i=-t+j+1}^{-1} \mathsf{ppd}_i^0 = \delta(j)$, $\mathsf{ppd}_{-t+j}^0 = \delta'(j)$, $\sum_{i=1}^{t-j} \mathsf{ppd}_i^0 = \alpha(j)$ and $\mathsf{ppd}_{t-j}^0 = \alpha'(j)$ (see Figure 2).

We now assume correctness for $r' < r$, i.e., if R_0 switches directions $r' < r$ times during the execution then $\sum_{l=1}^{2d} \mathsf{ppd}_l(R_0) < t$, and prove that this holds also for $r' = r$, $\tau = 1$.

The sum of ppd_i^0 for $r-1$ number of direction switches is $\delta(r-1) + \delta'(r-1) + \alpha(r-1) + \alpha'(r-1)$. For r switches, since the robots spend an extra time cycle for turning around, the two extreme segments with ppd > 0 are now unreachable, hence in this case $\delta'(r-1)$ and $\alpha'(r-1)$ no longer exist. Now, $\delta(r) + \delta'(r)$ is similar to changing the initial direction of the robot (by multiplying by $1-p$), and obtaining exactly $\alpha(r-1)$, hence $\delta(r) + \delta'(r) < (1-p)\alpha(r-1)$. Similarly, $\alpha(r) + \alpha'(r) < (1-p)\delta(r-1)$. Altogether, $\sum_{l=1}^{2d} \mathsf{ppd}_l^0(r) = \delta(r) + \delta'(r) + \alpha(r) + \alpha'(r) < (1-p)\alpha(r-1) + (1-p)\delta(r-1)$ and since $(1-p) < 1$ this is smaller than $\sum_{l=1}^{2d} \mathsf{ppd}_l^0(r-1)$. By the induction assumption, this is smaller than t.

The proof follows directly for $\tau > 1$, as the number of segments that become unreachable increases from 1 to τ for each direction switch, while the probability of penetration detection in other segments in the same. Therefore necessarily the sum of ppd after $r \geq 1$ direction switches with cost τ is now considerably smaller than the sum of ppd after r that costs only one extra time cycle, which is less than t, hence with cost τ for each switch this also holds. □

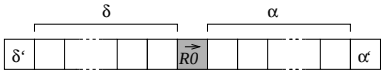

Figure 2: Illustration of proof of Lemma 1.

COROLLARY 2. *If the team of robots switched their direction $r > 0$ times during the execution (each directions switch lasts $\tau > 0$ time units), and the adversary chooses at random with uniform distribution its penetration spot, then the expected ppd throughout the perimeter is less than t/d.*

This is since the expected ppd throughout the perimeter is $1/N \sum_{j=i}^{k} \sum_{i=1}^{N} \mathsf{ppd}_i(R_j) < 1/N \sum_{j=i}^{k} t = kt/N = t/d$.

THEOREM 3. *The expected ppd throughout the perimeter assuming uniform adversary is maximal in case the value p characterizing the patrol of the robots equals 1, i.e., the patrol is deterministic for every $\tau \geq 1$.*

PROOF. If $p = 1$, then each robot $R_j, 1 \leq j \leq k$, assures $\mathsf{ppd}(R_j) = 1$ in exactly t segments, hence the expected ppd throughout the perimeter is $1/N \sum_j = 1^k t = kt/N = t/d$. Following Corollary 2, every patrol scheme that causes the robots to switch direction once or more, i.e., $p < 1$ has expected ppd less than t/d. Therefore the deterministic patrol guarantees maximal expected ppd. □

3.4 Partial information

We have shown that the algorithm maximizing the expected ppd is the deterministic algorithm (Theorem 3). However, the deterministic algorithm creates high deviations between the ppd values in the segments: in t segments the ppd equals 1, and in the other $d-t+1$ segments the value is 0. In addition, the deterministic algorithm is easy to detect, therefore if the adversary has even little time to study the system, it might deduce the type of the algorithm and choose to penetrate through a segment with $\mathsf{ppd}=0$, hence penetrate successfully. If we want to adapt a more "risk averse" approach, we would want to minimize the deviation between the ppd values throughout the perimeter. Therefore we present the Combine(d,t,w) that maximizes the expected ppd for the given d and t while minimizing the deviation between the ppd along the segments. This combination is done using the weight $w, 0 \leq w \leq 1$ for maximization of expected ppd and weight $1-w$ for minimization of the deviation. A full description of Algorithm Combine is given in Figure 3. Note that this algorithm uses a procedure of Algorithm MaxiMin for finding the ppd in each segment. This procedure is dynamic-programming inspired, separates each state into two based on directionality (clockwise and counterclockwise) and by assigning values in a matrix it determines the ppd in a segment. An absorbing state is used in order to represent the fact that the ppd is determined only by the *first* visit to a segment. Denote the standard deviation between the ppd values of a vector of functions $F = \{f_1, \ldots, f_{d-1}\}$ by stdev(F). Denote each segment $i, 1 \leq i \leq d-1$ by s_i.

Algorithm Combine(d,t,w)

1. Calculate F as follows:
   (a) **For** each $s_{init} = s_i \in \{s_1, \ldots, s_{d-1}\}$ do:
   (b) Create the matrix M of size $(2d+2) \times (t+1)$, initialized with 1 in $M_0(s_{init})$ and 0s otherwise, using the following rules.
      i. **For** each entry $M_t(s_i^{cw})$ set value to $p \cdot M_{t-1}(s_{i+1}^{cw}) + q \cdot M_{t-1}(s_i^{cc})$.
      ii. **For** each entry $M_t(s_i^{cc})$ set value to $p \cdot M_{t-1}(s_{i-1}^{cc}) + q \cdot M_{t-1}(s_i^{cw})$.
      iii. **For** absorbing states, set entry $M_t(s_{abs}) = M_{t-1}(s_{abs}) + p \cdot [M_{t-1}(s_1^{cw}) + M_{t-1}(s_d^{cc})]$.
   (c) $F \leftarrow$ row t of M.
2. $Q_1 \leftarrow 1/d \sum i = 1^d F_i$
3. $Q_2 \leftarrow 1 - \text{stdev}(F)$
4. $Q \leftarrow wQ_1 + (1-w)Q_2$
5. return $p = \max\{Q\}$

Figure 3: Description of Combine algorithm.

4. EVALUATION

In order to evaluate the behavior of the adversary and the performance of the robots in adversarial environment with different knowledge types, we created the Penetration Detection game (PenDet-game). This game was played by 68 human subjects that played the role of the adversary, and tried to penetrate through a perimeter defended by a team of simulated patrolling robots. We describe the game in Subsection 4.1, and the results in Subsection 4.2.

4.1 The PenDet-game

In the PenDet-game, a human player played the role of the adversary, working against a team of simulated patrolling robots. Therefore the player was required to pick a segment through which he thought he could penetrate without being detected. The game consisted of three stages, where in each stage the player had more time to study the system, i.e., more information concerning the patrolling robots was revealed gradually to the player. In the following, we describe the game in detail.

Note that our choice of performing experiments in this simulated environment, rather than actual robots is not trivial. The reason for preferring to conduct such experimental research, is that managing to evade patrolling robots using current lab-robots is simple — the adversary can simply jump over them. Moreover, in order to evaluate performance of the patrol algorithms extensive experiments were required. This is again impossible to create with real robots. Note that there are empirical results from running experiments with real robots in systems with adversarial teams, e.g. the Robocup game [8], however it was conducted between two teams of robots, not humans vs. robots.

The game consists of four robots patrolling around a treasure pot. In the game screen, the player can see the circle representing the perimeter and the patrolling robots (Figure 4). The distance between the robots and the time it takes to penetrate change from one subgame to the other. These values are presented explicitly to the player throughout the subgame. For simplicity reasons, we designed the game with $\tau = 1$, i.e., each time the robots switched directions they stayed in the same segment during that time cycle.

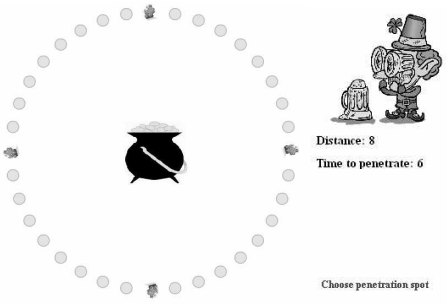

Figure 4: The PenDet-game screen.

In order to simulate different knowledge states, the game has three stages.

1. In the first stage, the player was shown a static picture with the current location of the robots. The direction of the robots (where they are facing) could be easily deduced from the picture of the robots. The player was requested to choose the segment through which he believes he will penetrate without being detected. This stage consists of three such sub-games. In each sub-game the distance between the robots and/or the penetration time t was different.
2. In the second stage, the player was shown five seconds of the patrol. After these five seconds passed the player was requested to click on the segment through which he thought he has best chances to penetrate without being detected. No feedback is given to the player

regarding whether it succeeded or failed in his attempt to penetrate. This stage consists of six sub-games. In each sub-game the patrol scheme of the robots, the distance between the robots and/or the penetration time t is different.

3. The third stage of the game is a three minute game, in which the player can try to penetrate as many times as he wants (as long as time permits) simply by clicking on the section through which it decides to penetrate. The player can see whether it succeeded or failed in his attempt. This stage also consists of six sub-games. In each sub-game the patrol scheme of the robots, the distance between the robots and/or the penetration time t is different.

Each game (and all its subgames) was played *once* by each player, so that the primacy of the choices taken by the players in the first stages is maintained.

The PenDet-game was played by 68 human subjects (29 Female/39 Male). All subjects were senior undergraduate students in computer science. Each subject played the game *once*. The game was set online, and the students were required to play it as part of course requirements.

The information regarding the d, t, p values tested, is given in the Table 1.

Table 1: The d, t, p values tested in the experiment.

d	t	p Det	p Combine	p MaxiMin
16	9	1	0.93	0.87
8	5	1	0.92	0.75
8	6	1	0.96	0.7
12	9	1	0.97	0.77
12	11	1	1	0.82
16	15	1	1	0.85

Three different patrol algorithms were executed in three stages of the game. The patrol algorithm is determined by the probability p characterizing the robots' movement. The first algorithm corresponds to the zero knowledge adversary, therefore following Theorem 3 this is the deterministic patrol algorithm ($p = 1$). We denote this algorithm by Det. Note that the player does not have completely *no knowledge* of the patrol scheme, as it knows the distance between the robots and the direction it is facing. However, this information is minimal and reveals nothing concerning the patrol algorithm. The second algorithm is the Combine algorithm described in Figure 3. The third algorithm corresponds to the full knowledge adversary. Note that the player does not have *full* knowledge, but it gets a long period of time to study the system, which brings it close to a full knowledge adversary. In this case, the p values represent the probability yielding the maximal minimal ppd along the perimeter. These value of p were calculated using the MaxiMin algorithm described in [1].

Det was executed in the first and second stages of the game, where MaxiMin and Combine were executed in all three stages of the game. The reason for omitting Det from the third stage is threefold. First, the combined algorithm reaches 1 as t gets close to d, and is exactly 1 (deterministic) in case $t = d - 1$. Therefore checking two cases of deterministic algorithms would give a clear picture of this behavior. Second, we assumed that in this case the learning curve of the players will be steep, i.e., they will understand that this is a deterministic algorithm and succeed in nearly all attempts to penetrate. Last, we thought that this algorithm might bore the players and unmotivate them to play thoughtfully for long. This was verified by feedback we received from subjects in early development stages of the game.

In order to evaluate the performance of the three algorithms, we executed the algorithms on the input retrieved from the players. In the first stage, each player provided 3 input lines, each compatible to one pair of d and t. For each such input line, we executed each algorithm 100 times in order to evaluate the penetration detection ratio. In the second stage, each input from each player corresponds to a triplet, d, t, p. Therefore each input line was continued 100 times only with the algorithm it initially executed. In the third stage, each player provides multiple input lines, each such line corresponds to one choice of penetration spot, suitable for a triplet d, t, p of the subgame. All such input lines were recorded as is, and the information regarding successful/failure of penetration was extracted from the game itself.

4.2 Experimental results

In the following, we describe the results of the experiments with the PenDet-game. First we describe the bottom line summary of the results, then we discuss in detail the results of each stage of the game.

Figure 5 describe the summary of the results obtained from all three stages of the game. It presents the maximal, minimal and average penetration detection ratio obtained by each of the algorithms we tested in each stage of the game. Comparing these values, it is clearly seen that in the first stage the deterministic algorithm is the best: its average is considerable higher than the average of the other algorithms, and the minimal penetration detection is also considerable higher. Note that the maximal value of penetration detection is equal to the maximal value of the Combine algorithm, since this value is resulted from the Combine algorithm when it offers deterministic behavior. In the second stage, the average value of penetration detection obtained by all three algorithms is similar, yet the Combine algorithm is considerably better than the deterministic algorithm in its minimal value, and substantially better than the MaxiMin algorithm in its maximal value. In stage 3, MaxiMin significantly outperforms the Combine algorithm in both average and maximal penetration detection. The minimal penetration detection is similar, as when t is small relative to d, MaxiMin cannot guarantee high ppd values, thus they are similar to the ppd guaranteed by the Combine algorithm.

The game results are mainly evaluated in terms of actual percentage of penetration detection from all three algorithms, which corresponds to the robots' performance in different scenarios. In some cases, we have found that the choice of the player of the section through which he decided to penetrate yielded interesting results. This corresponds to the decisions taken by the adversary after attaining different levels of information.

Stage 1*:*

In the first stage, nearly no information was given to the player. Therefore the players could have chosen at random the penetration spots. In Figure 6 we see that, however, in most cases the players chose to penetrate through one of two segments in the middle. This is not surprising, considering that people are drawn to central positions when instructed to choose between positions that have no apparent special

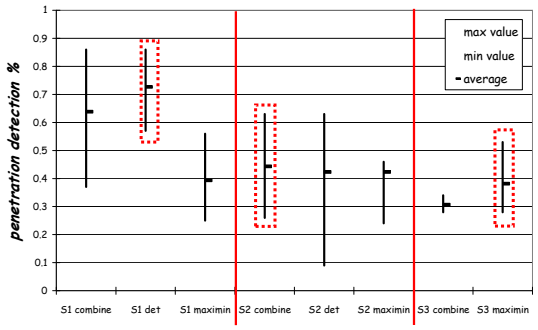

Figure 5: A summary of results, divided into three stages: no information (stage 1), short-term revelation of information (stage 2), and long term revelation of information (stage 3) for the three patrol algorithms. Each line represents the maximal, minimal and average penetration detection. The best performing algorithm in each stage is noted by a surrounding dotted rectangle.

characteristics [9]. In addition, the direction of the robot is visible to the player, hence he might take that into consideration. This is apparent in the case where $d = 12$ and $t = 11$, in which 28% of the players chose to penetrate through the last segment.

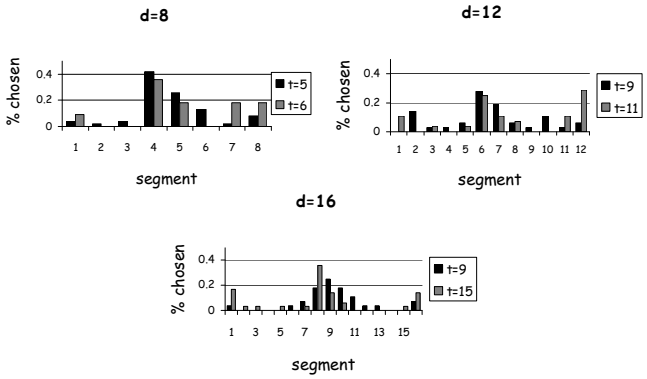

Figure 6: Choices of penetration positions in stage 1 for different values of d: $d = 8, 12, 16$. The x axes represents the segment, and the y axes the percentage of subjects that chose to penetrate through that segment.

As seen clearly in Figure 7, the algorithm that managed to detect the highest percentage of the penetrations is the deterministic algorithm (statistically significant, using t-test with p-value < 0.01). Therefore the deterministic algorithm is indeed more suitable for detecting penetrations in case the adversary has nearly no knowledge of the patrol scheme. Moreover, even if the adversary has some knowledge - in our case the distance between the robots and the direction they are currently heading - this algorithm still nicely performs. However, the expected values of ppd ("theoretical expected ppd" in Figure 7) are higher than what was obtained in the actual game. The reason is that the algorithm is maximizing the expected ppd if the adversary chooses at random its penetration spot. However, as we have seen previously, this is not necessarily the case (in other words: it expects a less sophisticated adversary). On the other hand, the MaxiMin algorithm expects to be teamed up against a much more sophisticated adversary, therefore the actual penetration detection percentage is higher than the theoretical values. Note that the Combine algorithm coincides with the deterministic algorithm for some scenarios ($t/d = 11/12, 15/16$), therefore the penetration detection percentages of both are identical in those cases.

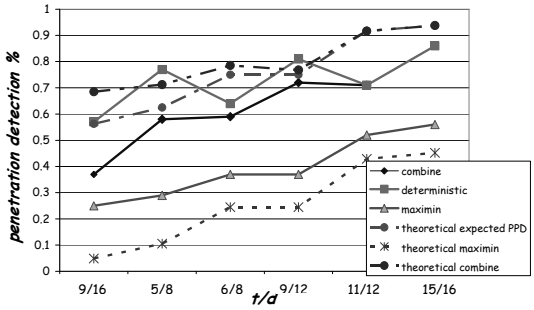

Figure 7: Performance of the three different algorithms in stage 1 (adversary with nearly zero knowledge).

Stage 2:
When only a small amount of information was revealed to the player concerning the patrol scheme (5 seconds), then the Combine algorithm performed generally better compared to the other algorithms. Moreover, We checked the minimal 25% values and the maximal 25% values of penetration detection obtained by the algorithms, and found that the Combine algorithm is statistically significantly better than MaxiMin and significantly better than Det algorithm in cases where Combine is non deterministic (using t-test with $p-value < 0.01$) - see Figure 8. Note that the theoretical values of ppd in both Combine and Det are considerable higher than the actual penetration detection ratio. This is clear, as the robots expect an adversary with no knowledge about the system, yet are faced against an adversary that has gained some information. On the other hand, the theoretical values of the MaxiMin still pose as a lower bound to the performance of the robots.

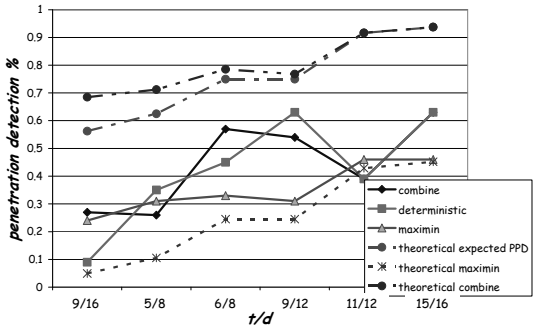

Figure 8: Performance of the three different algorithms in stage 2 (adversary with little knowledge)

Stage 3:
We present the results from stage 3 in two ways: the overall performance and the performance after omitting the first 30 seconds of the game. We consider the first 30 seconds to be a learning period, mainly for the Combine algorithm, when it produces a deterministic schedule.

The following results are obtained when comparing the performance of the robots when omitting the first 30 seconds. When using the combine algorithm, then when the algorithm is deterministic the penetration detection decreases from 30% or more to 20%. Therefore even when the adversary observed the patrol for only 30 seconds, it managed to substantially increase its chances of successful penetration. In fact, penetration detection ratio using the MaxiMin algorithm is statistically significantly better compared to using the Combine algorithm (t-test with p-value < 0.01).

This fact is even more interesting, as we expect that when the penetration time t is higher, the robots will more likely detect the penetration (it can be clearly seen for the MaxiMin for all d, t pairs and for Combine in all non-deterministic cases without removing the learning phase). However, since the deterministic patrol scheme is simple and easily detected, when used even with high values of t, the adversary takes advantage of it and manages to penetrate with a higher probability. This fact again strengthens the motivation for inventing an additional patrol scheme, the Combine algorithm, which is described in Figure 3. Note that the theoretical MaxiMin ppd guarantees still a lower bound to all non deterministic behaviors tested here.

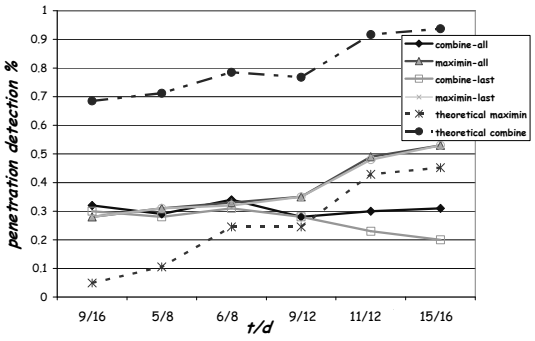

Figure 9: Performance of the two different algorithms in stage 3 (adversary with full knowledge)

5. SUMMARY AND FUTURE WORK

This paper considers the problem of multi-robot patrol around a closed area, in the presence of an adversary trying to penetrate the area. Previous work mostly concentrated on strengthening the abilities of the team of agents facing a strong adversary [10, 1]. However, adapting this behavior by a team of robots is not beneficiary in case they are not facing an adversary with full knowledge of the patrol scheme.

In this work, we have shown that as opposed to the optimality of non-deterministic algorithms when working against a strong adversary, in case the adversary has no knowledge of the patrol scheme, a simple deterministic algorithm is optimal. This is a surprising results, especially in the case in which turning around is not a costly operation. We propose an algorithm to deal with adversary having partial information, Combine, that maximizes the expected probability of penetration detection along with minimizing the deviation between the probabilities.

We performed extensive experiments with human subjects concerning the compatibility of algorithms to the extent of information possessed by the subjects. We have shown that the Combine algorithm performed best if some information was revealed to the subjects. The optimality of the deterministic algorithm for subjects with no knowledge and the optimality of the non-deterministic algorithm for adversary with more information was verified by the experiments.

There are various points we wish to address as future work. First, it could be shown that the MaxiMin algorithm is in equilibrium, however this is not the case for the other algorithms. Therefore we would like to further examine the system in game theoretical tools. We are interested in dealing with errors in the movement model of the robots and in sensor problems. For example, what are the consequences of a coordination failure between the robots, what happens if some robots deviate from their expected path or velocity, and so on. We are also currently in contact with commercial companies in order to evaluate the performance of our algorithm in a large scale simulation.

6. REFERENCES

[1] N. Agmon, S. Kraus, and G. A. Kaminka. Multi-robot perimeter patrol in adversarial settings. In *ICRA*, 2008.

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

[3] 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.

[4] D. Carmel and S. Markovitch. Incorporating opponent models into adversary search. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 120–125, 1996.

[5] Y. Chevaleyre. Theoretical analysis of the multi-agent patrolling problem. In *IAT*, 2004.

[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] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa, and H. Matsubara. Robocup: A challenge problem for ai and robotics. In *RoboCup-97: Robot Soccer World Cup I*, pages 1–19, 1998.

[9] S. Kraus, J. S. Rosenschein, and M. Fenster. Exploiting focal points among alternative solutions: Two approaches. *AMAI*, 28(1–4):187–258, 2000.

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

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

[12] H. Santana, G. Ramalho, V. Corruble, and B. Ratitch. Multi-agent patrolling with reinforcement learning. In *AAMAS*, pages 1122–1129, 2004.

[13] 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.