

# Finding the Optimal Strategies for Robotic Patrolling with Adversaries in Topologically-Represented Environments

Francesco Amigoni, Nicola Basilico, and Nicola Gatti

**Abstract**—Using autonomous mobile robots to patrol environments for detecting intruders is a topic of increasing relevance for its possible applications. A large part of strategies for mobile patrolling robots proposed so far adopt some kind of random movements. Although these strategies are unpredictable for an intruder, they are not always efficient in getting the patroller a large expected utility. In this paper we propose an approach that considers a model of the adversary in a game theoretic framework to find optimally-efficient patrolling strategies. We show that our approach extends those proposed in literature and we experimentally analyze some of its features.

## I. INTRODUCTION

Protecting sites against intrusions is a topic of increasing importance, and robotic systems for autonomous patrolling have been developed in the last years [1], [2]. These systems usually adopt some kind of randomized patrolling strategies in order to make their routes unpredictable for an observing intruder [3]. However, unpredictable strategies are not always efficient in getting the patroller a large expected utility. Recently, models of the adversary have been considered in the attempt to improve the efficiency of the patrolling strategies [3], [4].

In this paper we propose an approach that, exploiting a model of the adversary in a game theoretic framework, finds optimal strategies that maximize the expected utility of the robotic patroller. Specifically, our approach is based on modeling a given patrolling situation as an extensive-form game [5]. We recall that a game is formally defined as a pair: the *mechanism*, which sets the rules of the game, and the *strategies*, which specify the players' behavior. When the mechanism forces players to act sequentially in turns, the game is an extensive-form game. The solution of a game is a set of equilibrium strategies that rational players should employ [5]. In this paper, we define a game in which two players (the patroller and the intruder) take sequential actions (like in chess); its solution gives the optimal patrolling strategy for the modeled situation. We formulate the problem of finding this solution as a mathematical programming problem that can be solved by optimization software tools [6]. The model we propose generalizes those proposed in previous works, because it considers environments with a topological structure (extending [4]) that can be of any form (extending [3]). Experimental results show that our approach is viable and can be applied in many situations.

The authors are with the Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milano, Italy {amigoni,basilico,ngatti}@elet.polimi.it

The main original contribution of this paper is the game theoretical extensive-form model for robotic patrolling that extends previously proposed models. In this way, we are able to find the optimal patrolling strategy in a large number of situations. We consider topological representations of environments, leaving the inclusion of metric issues in our model to future works.

This paper is organized as follows. The next section surveys the related works on robotic patrolling with adversaries. Section III introduces the proposed approach, including the game theoretical model and the algorithm for finding an optimal patrolling strategy. In Section IV we prove that our approach generalizes that in [3]. In Section V some experimental results are reported.

## II. ROBOTIC PATROLLING WITH ADVERSARIES

A patrolling situation can be considered as characterized by one or more *patrollers* and by some *targets* to be patrolled. These targets can be access points along a perimeter [1], [3] or areas with some value [7]. Patrolling can be conveniently performed using mobile robots [1], [2], [3]. The study of robotic patrolling is particularly interesting when, due to some characteristics of the setting (e.g., speed of the patrollers and time needed by a possible intruder to attack a target), the patrollers cannot employ a deterministic strategy for their movement. As a result, they should adopt an unpredictable patrolling strategy, randomizing over the targets trying to reduce the intrusion probability of a possible adversarial [3]. Some patrolling strategies of this type have been developed, but they usually do not consider any explicit model of the adversary [1], [8]. Only very recently, adversaries have been taken into account in developing patrolling strategies. As shown in [9], patrolling strategies that consider models of adversaries can give the robot a larger expected utility than strategies that do not. The two main methods proposed in literature for robotic patrolling with adversaries differ in that one does not explicitly model the preferences of the adversaries [3], whereas the other one does [4], [7]. Before briefly reviewing these methods, we note that similar strategic problems have been addressed in the pursuit-evasion field (e.g., [10], [11]). However, some assumptions, including the fact that the evader's goal is only to avoid capture and not to enter an area of interest and the fact that the evader sometime knows the current position of the pursuer but not its strategy, make the pursuit-evasion strategies not directly applicable to our patrolling scenario.

The work in [3] does not employ any explicit model of the adversary's preferences. The authors consider the problem

of patrolling a perimeter divided in cells, each one allowing the access to an area of interest, by employing a team of synchronized mobile robots acting in turns. The perimeter is considered as a ring whose cells require the same penetration time, say  $d$  turns, to the intruder for entering. The robots keep an evenly separated formation by moving in a coordinated fashion. The patrolling strategy does not depend on the specific cells in which the robots are. The authors present different movement models for the robots. In the simplest one, all the robots move clockwise with probability  $p$  or move counterclockwise with probability  $1 - p$ . In the most realistic movement model, all the patrollers move to the cells they are headed to with probability  $p$  and reverse their heading, staying in their current cells for a turn, with probability  $1 - p$ . Since the patrollers do not know the intruder's preferences, they assume that the intruder will try to enter through its most convenient access. The patrolling strategy thus amounts to choose  $p$  that maximizes the minimum expected utility for the patrollers or, equivalently, that maximizes the detection probability. This work presents two main limitations: first, it is applicable only to very special ring-like environments where all the cells have the same penetration time and the patrollers have no preferences over the cells, second, as we show in Section IV, the strategy it produces is optimal only when also the intruder has no preferences over the cells.

The work in [4] explicitly models the preferences of the adversary. The authors consider the problem of patrolling  $n$  areas by using a single patroller such that the number of turns it would spend to patrol all the areas is strictly larger than the penetration time  $d$  of the intruder,  $d < n$ . They model such a problem as a two-player (i.e., the patroller and the intruder) strategic-form game (like rock-paper-scissors game) with incomplete information (i.e., the intruder's preferences over the areas can be uncertain to the patroller) [5]. The actions available to the patroller are all the possible routes of  $d$  areas, while the intruder can choose a single area to enter. The intruder is assumed to be in the position to repeatedly observe the actions of the patroller (staying hidden), derive a correct belief on the patroller's strategy, and find its best response given the patroller's strategy. The appropriate equilibrium concept, in which the patroller maximizes its expected utility, is the leader-follower equilibrium [12]. (A slight variation of this approach has been applied to the problem of patrolling  $n$  areas with  $m < n$  static checkpoints at the Los Angeles International Airport [13].) As discussed in [14], the approach in [4] presents two drawbacks. First, since it does not consider time spent by the robot to move between two areas, the model is applicable only when this time is negligible with respect to the time needed to patrol an area. This amounts to say that the model does not consider the topology of the environments, because the patroller can move between any two areas in a fixed time. Second, to avoid game theoretical inconsistencies, the decisions of the patroller must be over the next area to patrol, and not over the next route to follow.

### III. THE PROPOSED MODEL

#### A. The Basic Model

The model we propose captures adversarial robotic patrolling settings based on the following assumptions:

- time is discretized in turns;
- there is a single patrolling robot;
- the environment to be patrolled can be represented with a 2D grid map, whose cells can be either free or obstacles; in this paper we only consider the topology of the environment (i.e., the adjacency of the cells) and not its metric features (e.g., the size of the cells);
- the intruder can observe the patroller's strategy before acting (this amounts to say that the intruder knows the patrolling strategy, as in [4]);
- the intruder cannot do anything else for some turns once it has attempted to enter a cell (this amounts to say that penetration takes some time, as in [3], [4]).

The patroller's goal is to detect the intruder. If this happens, we say that the intruder has been "captured" by the patroller. In what follows we detail the models of the environment, of the patrolling robot's movement and sensing capabilities, and the game mechanism.

The environment is composed of a set  $C$  of  $n$  free cells to be patrolled, whose topology is modeled by a directed graph  $\mathcal{G}$ . We represent  $\mathcal{G}$  by a matrix  $T(n \times n)$ , where  $t_{i,j} = 1$  if cells  $i$  and  $j$  are adjacent and  $t_{i,j} = 0$  otherwise. A cell may represent an access point to an area with some value (e.g., a door, as in [3]) or can be an area with some value (e.g., an house, as in [4]). Agents can have different preferences over the cells, as defined later. Finally, each cell  $i$  requires the intruder  $d_i > 0$  turns to enter ( $d_i$  is called the penetration time for cell  $i$ ).

Our approach accommodates for different movement models for the patrolling robot. Here, we illustrate the simplest one: in one turn the patroller moves between two linked (adjacent) cells in  $\mathcal{G}$  and patrols the destination cell. (The model can be easily augmented, for example for capturing the time spent by the robot in turning around, as in [3].) As other approaches in literature, we do not adopt any movement model for the intruder: it can directly enter any cell. The patrolling robot is equipped with a sensor able to detect the intruder within a range  $r$ . For now, we assume that the patroller can sense only the cell in which it currently is, namely  $r = 1$ . We will extend  $r$  in Section III-D.

We model the above scenario as a two-player dynamic repeated game [5]. A strategic-form game is repeated at each turn, in which the patroller and the intruder play simultaneously. The patroller chooses the next cell to patrol among those adjacent to its current cell; formally, its actions are  $move(i, j)$  where  $i$  is its current cell and  $j$  is the destination cell, such that  $t_{i,j} = 1$ . The intruder, if it has not previously attempted to enter any cell, chooses whether or not to enter a cell and, in the former case, what cell to enter (formally,  $wait$  and  $enter(i)$ ). If, instead, the intruder has previously decided to enter a cell  $i$ , it cannot take any action for  $d_i$  turns after decision. The repeated game is dynamic

since it changes at each repetition (turn): the positions of the patroller (i.e., its current cell) and of the intruder (i.e., trying to get inside a cell or waiting) change. We assume the patroller’s policy to be Markovian. The probability according to which the patroller chooses the next cell to visit depends only on the cell in which it currently is. We denote by  $\alpha_{i,j}$  the probability for the patroller to move from cell  $i$  to cell  $j$ . The set  $\{\alpha_{i,j}\}$  of all these probabilities thus constitutes the patrolling strategy of the robot. Under the hypotheses of patroller’s Markovian strategy and intruder’s observability of the patroller’s actions, the intruder’s strategy during the game can be conveniently represented by using the following macro-actions:  $\{enter-when(i, j) : 1 \leq i, j \leq n\} \cup \{stay-out\}$  where  $i$  is the cell to enter and  $j$  is the cell where the patroller was at the last observation. Action  $enter-when(i, j)$  corresponds to make *wait* while the patroller is not in cell  $j$  and then make  $enter(i)$ ;  $stay-out$  corresponds to make *wait* forever. The possible outcomes of the game are: *intruder-capture*, when the intruder attempts to enter a cell  $i$  at  $t$  and the patroller visits cell  $i$  in the time interval  $\{t, t + 1, \dots, t + d_i - 1\}$ ; *penetration- $i$* , when the intruder enters a cell  $i$  at  $t$  and the patroller does not visit cell  $i$  in the time interval  $\{t, t + 1, \dots, t + d_i - 1\}$ ; *no-attack*, when the intruder never enters any cell.

Agents’ payoffs are defined as follows. We denote by  $X_i$  and  $Y_i$  (with  $i \in \{1, 2, \dots, n\}$ ) the payoffs to the patroller and to the intruder, respectively, when the outcome is *penetration- $i$* . We denote by  $X_0$  and  $Y_0$  the payoffs to the patroller and to the intruder, respectively, when the outcome is *intruder-capture*. For the sake of simplicity, we assume that, when the outcome is *no-attack*, the payoff to the patroller is  $X_0$  and the payoff to the intruder is 0. (The rationale is that, when the intruder never enters, it gets nothing and the patroller preserves values of all the cells.) Consistency constraints over these values are:  $X_i \leq X_0$  and  $Y_0 \leq 0 \leq Y_i$  for all  $i \in \{1, 2, \dots, n\}$ .

An example of environment captured by our model is depicted in Fig. 1. The bold numbers identify the free cells; black cells are obstacles; the values reported in cell  $i$  are  $(X_i, Y_i)$ . The values  $(X_0, Y_0)$  are  $(1, -1)$ . Note that the payoffs to the patroller are given in such a way that it prefers the intruder entering cell 04 rather than cell 06 (this is equivalent to say that cell 04 contains “less value” than cell 06). Note also that cells 04, 06, and 12 have some interest for the intruder (for them,  $Y_i > 0$ ). They may represent houses with goods to rob or access points to a warehouse. In this sense, our model is very general and can represent several situations. In a real setting, values  $X_0$ ,  $X_i$ s,  $Y_0$ , and  $Y_i$ s can be estimated according to the value of the real places corresponding to cells, while penetration times  $d_i$ s can be estimated according to the difficulty to intrude in the places.

## B. The Solution Concept

According to game theory, a solution for the game we have defined is a strategy profile  $\sigma^* = (\sigma_p^*, \sigma_i^*)$ , where  $\sigma_p^*$  is the strategy of the patroller and  $\sigma_i^*$  is the strategy of the intruder, that is in equilibrium. (We recall that a player’s

<b>01</b> (1,0) $d_{01} = 1$	<b>02</b> (1,0) $d_{02} = 1$	<b>03</b> (1,0) $d_{03} = 1$	<b>04</b> (.8,.4) $d_{04} = 6$	<b>05</b> (1,0) $d_{05} = 1$
<b>06</b> (.7,.5) $d_{06} = 4$		<b>07</b> (1,0) $d_{07} = 1$		<b>08</b> (1,0) $d_{08} = 1$
<b>09</b> (1,0) $d_{09} = 1$	<b>10</b> (1,0) $d_{10} = 1$	<b>11</b> (1,0) $d_{11} = 1$	<b>12</b> (.8,.4) $d_{12} = 5$	<b>13</b> (1,0) $d_{13} = 1$

Fig. 1. Example of environment

strategy is a randomization over its actions.) Usually, solution concepts refer to Nash equilibrium, a stable strategy profile from which no player has incentive to deviate [5]. When, as in our case, the intruder can observe the patroller before acting, the appropriate equilibrium concept is the *leader-follower equilibrium* [12]. Indeed, it has been proved [12] that this equilibrium gives the leader, in our case the patroller, the maximum expected utility, knowing that the follower, in our case the intruder, will observe its strategy and will act as a best responder. This means that the leader-follower equilibrium strategy for the robot is its optimally efficient patrolling strategy. The leader-follower equilibrium exhibits several interesting properties [12], including that it always exists and that the follower employs pure strategies, i.e., it plays one of its actions without randomizing.

The literature provides algorithms for finding leader-follower equilibria only in strategic-form games by solving a multi-linear programming problem [15].<sup>1</sup> Essentially, since at the equilibrium the follower will employ pure strategies, it is possible to enumerate them and, for each pure strategy  $\sigma_i = i$ , to calculate the maximum expected utility  $EU_p(BR_i = i)$  of the patroller such that  $i$  is a best response for the intruder. There are as many maximization problems as the actions of the intruder and each single problem is linear. The patroller will induce the intruder to follow the strategy  $i$  such that  $EU_p(BR_i = i)$  is maximum. Unfortunately, no algorithm is available for dynamically repeated games, such as the one we are dealing with. However, by considering the macro-actions introduced in the previous section, the dependence from the turns is removed and the dynamic repeated game is reduced to a strategic-form game with constraints on the patroller’s strategies (basically, they are constrained to be Markovian). This game can be solved by an algorithm, introduced in the next section, that extends the basic algorithm just described.

## C. Finding a Solution Via Mathematical Programming

In this section we formulate the problem of finding the leader-follower strategy of the patroller as a mathematical programming problem. This formulation follows from a typical approach in algorithmic game theory [16]. We denote by  $\gamma_{i,j}^{h,w}$  the probability that the patroller reaches cell  $j$  in  $h$  steps, starting from cell  $i$  and not passing through cell  $w$ . For the sake of presentation, we assume all  $d_i$ s to be equal to  $d$ .

<sup>1</sup>An alternative formulation based on mixed integer linear programming is provided in [7]. We do not consider it in our work, since it makes the corresponding mathematical programming problem too hard to solve.

The general case is an easy extension. Our solving algorithm develops into two stages.

In the first stage we check if there exists at least one patroller's strategy such that *stay-out* is a best response for the intruder. If such a strategy exists, then the patroller will follow it, being its payoff maximum when the intruder abstains from the intrusion (recall that  $X_0 \geq X_i$  for all  $i$ ). This stage is formulated as the following bilinear feasibility problem in which  $\alpha_{i,j}$ s are the unknown variables ( $C \setminus i$  is the set obtained by removing the element  $i$  from  $C$ ):

$$\alpha_{i,j} \geq 0 \quad \forall i, j \in C \quad (1)$$

$$\sum_{j \in C} \alpha_{i,j} = 1 \quad \forall i \in C \quad (2)$$

$$\alpha_{i,j} \leq t_{i,j} \quad \forall i, j \in C \quad (3)$$

$$\gamma_{i,j}^{1,w} = \alpha_{i,j} \quad \forall w, i, j \in C, j \neq w \quad (4)$$

$$\gamma_{i,j}^{h,w} = \sum_{x \in C \setminus w} (\alpha_{x,j} \gamma_{i,x}^{h-1,w}) \quad \forall h \in \{2, \dots, d\}, \quad \forall w, i, j \in C, j \neq w \quad (5)$$

$$Y_0 \left( 1 - \sum_{i \in C \setminus w} \gamma_{z,i}^{d,w} \right) + Y_w \sum_{i \in C \setminus w} \gamma_{z,i}^{d,w} \leq 0 \quad \forall z, w \in C \quad (6)$$

Constraints (1)-(2) express that probabilities  $\alpha_{i,j}$ s are well defined; constraints (3) express that the patroller can only move between two adjacent free cells; constraints (4)-(5) express the Markovian hypothesis over the patroller's decision policy; constraints (6) express that no action *enter-when*( $w, z$ ) gives to the intruder an expected utility larger than that of *stay-out*. Notice that the non-linearity is due to the constraints (5). If the above problem admits a solution, the resulting  $\alpha_{i,j}$ s are the optimal patrolling strategy for the robot. When the above problem is unfeasible, we pass to the second stage of the algorithm.

In the second stage we find the best response  $i$  of the intruder such that the patroller's expected utility is maximum. This is formulated as a multi bilinear programming problem, where the single bilinear problem in which *enter-when*( $q, s$ ) is the best response for the intruder is defined as follows:

$$\max \quad X_q \sum_{i \in C \setminus q} \gamma_{s,i}^{d,q} + X_0 \left( 1 - \sum_{i \in C \setminus r} \gamma_{s,i}^{d,q} \right)$$

s.t.

constraints (1)-(5)

$$\begin{aligned} Y_0 \left( 1 - \sum_{i \in C \setminus q} \gamma_{s,i}^{d,q} \right) + Y_q \sum_{i \in C \setminus q} \gamma_{s,i}^{d,q} &\geq \\ \geq Y_0 \left( 1 - \sum_{i \in C \setminus w} \gamma_{z,i}^{d,w} \right) + Y_w \sum_{i \in C \setminus w} \gamma_{z,i}^{d,w} &\quad \forall z, w \in C \end{aligned} \quad (7)$$

The objective function is the maximization of the patroller's expected utility. Constraints (7) express that no action *enter-when*( $w, z$ ) gives a larger value to the intruder than action *enter-when*( $q, s$ ). We can formulate  $n^2$  above problems, for all the possible *enter-when*( $q, s$ ) actions ( $q, s \in \{1, 2, \dots, n\}$ ). If a problem is feasible, its solution is a set of  $\alpha_{i,j}$ s, namely a possible patrolling strategy for the robot. From all the solutions of feasible problems, we pick out the one that gives the patroller the maximum expected utility.

We report in Fig. 2 the optimal patroller's strategy for the setting of Fig. 1, as calculated with the algorithm described here. The expected utility for the patroller is 0.805 and the corresponding induced best response for the intruder is *enter-when*(04,01), namely to enter cell 04 when the patroller is in 01. Cells 05, 08, and 13 are excluded from the route of the patroller. Indeed, visiting these cells would allow the intruder to perform an always successful action, for example *enter-when*(06,08) (see Fig. 1).

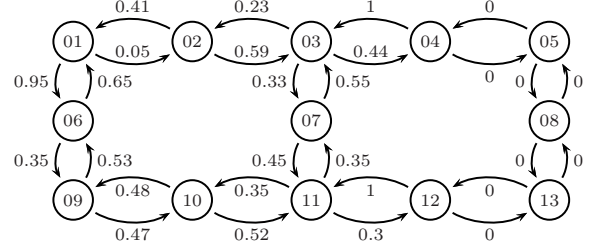


Fig. 2. Patroller's optimal strategies in the setting of Fig. 1

#### D. Augmenting Patroller's Sensing Capabilities

In this section we extend the sensing model of the patrolling robot by considering that it can sense the presence of the intruder beyond its current cell. We introduce a matrix  $V(n \times n)$  where  $v_{i,j} = 1$  if cell  $j$  can be sensed by the patroller from cell  $i$  and  $v_{i,j} = 0$  otherwise. Matrix  $V$  embeds a model of the detecting sensor of the robot. A general sensing model of the patroller can be considered in our approach by substituting constraints (4)-(5) above with:

$$\gamma_{i,j}^{1,w} = \alpha_{i,j} (1 - v_{j,w}) \quad \forall w, i, j \in C, j \neq w \quad (8)$$

$$\gamma_{i,j}^{h,w} = (1 - v_{j,w}) \sum_{x \in C \setminus w} (\alpha_x \gamma_{i,x}^{h-1,w}) \quad \forall h \in \{2, \dots, d\}, \quad \forall w, i, j \in C, j \neq w \quad (9)$$

In this case  $\gamma_{i,j}^{h,w}$  is the probability that the patroller reaches cell  $j$  in  $h$  steps, starting from cell  $i$  and not having sensed cell  $w$ . For example, let us consider that the patroller is able to sense its current cell and the free cells that are one cell away from it ( $r = 2$ ). For instance, in Fig. 1, from cell 05, it can sense cells 04 and 08; from cell 06, it can sense cells 01 and 09; from cell 11, it can sense cells 07, 10, and 12; and so on. With this sensor model, the optimal patrolling strategy for the robot is reported in Fig. 3. Comparing with Fig. 2, cells 04 and 12 have been excluded from the patrol route. This makes sense, since the patroller, due to the augmented sensing capabilities, is able to patrol them from adjacent cells that are more "central" (03 and 11, respectively).

#### IV. A SYNCHRONIZED MULTIROBOT SETTING

In this section we show (a) that our model can easily capture settings with multiple synchronized robots and (b) that the patroller's strategy produced by our approach, when the patroller and the intruder have no preferences over the cells, and by the approach presented in [3] are the same. Hence, since our approach is able to capture more general

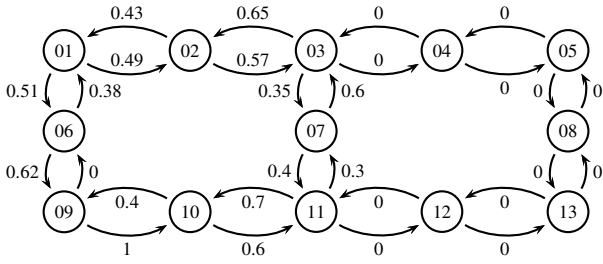


Fig. 3. Patroller’s optimal strategy in the setting of Fig. 1 with  $r = 2$

settings (like that in Fig. 1 with  $r = 2$ ), our results generalize and extend those presented in [3].

We recall that the algorithm in [3] is applicable to ring-like environments where all the cells have the same penetration time and the team of synchronized patrolling robots have no preferences over the cells. Our model can capture this multirobot setting by exploiting sensing augmentation of previous section. Basically, we can imagine to have a single *virtual robot* that is able to sense all the cells wherein the real robots are. This can be easily accounted for by appropriately setting matrix  $V$ . Consider the example reported in Fig. 4 with four robots (depicted as black circles). With  $r = 1$  the virtual robot is able to sense cells 01, 04, 09, and 12. If the robots are synchronized (as in [3]), we can select a real robot  $r$ , say that in cell 01, and express the patrolling strategy of the virtual robot in terms of the patrolling strategy of  $r$ . The actions available to the intruder are of the form *enter-when*( $i, j$ ) where  $i$  is the cell to enter and  $j$  is the cell wherein  $r$  is. Applying the algorithm of Section III-C to this model returns the optimal strategy for the patrolling multirobot system. Fig. 5 shows this strategy for the setting of Fig. 4.

01 ● (1,0) $d_{01} = 1$	02 (1,0) $d_{02} = 1$	03 (1,0) $d_{03} = 1$	04 ● (.8,.4) $d_{04} = 2$	05 (1,0) $d_{05} = 1$
06 (.7,.5) $d_{06} = 2$				07 (1,0) $d_{07} = 1$
08 (1,0) $d_{08} = 1$	9 ● (1,0) $d_{09} = 1$	10 (1,0) $d_{10} = 1$	11 (.8,.4) $d_{11} = 5$	12 ● (1,0) $d_{12} = 1$

Fig. 4. A synchronized multirobot scenario

We now show that the optimal strategy in our model (i.e., the leader-follower strategy) and the one in [3] (i.e., the maxmin strategy) are the same when the patroller (intended as a single virtual robot) and the intruder have no preferences over the cells. We state the following theorem.

**Theorem 4.1:** When both the patroller and the intruder have no preferences over the cells, the patroller’s leader-follower strategy in the game of Section III-A is a maxmin strategy.

*Proof.* The situation when agents have no preferences over the cells can be easily modeled by setting  $X_i$  and

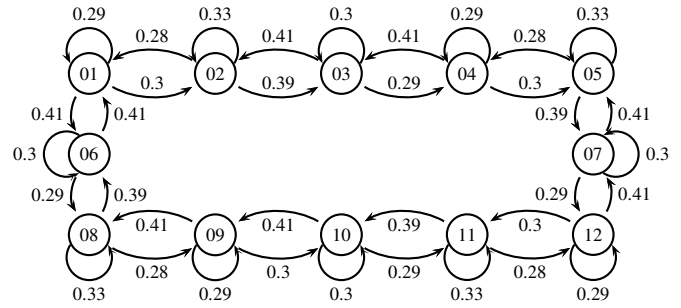


Fig. 5. Patrollers’ optimal strategy in the setting of Fig. 4

$Y_i$  to be the same over all the cells. In this situation, the game described in Section III-A can be easily translated into an equivalent zero-sum game. By the von Neumann and Morgenstern theorem [5], in a two-player zero-sum game agents’ maxmin strategies and minmax strategies are equivalent and they are the optimal agents’ strategies. That is, they constitute a Nash equilibrium. Moreover, by the von Stengel and Zamir theorem [12], in a two-player zero-sum game the leader-follower strategies of the leader are equivalent to those in the Nash equilibrium. Hence, under the hypotheses of the theorem, the patroller’s optimal strategies are maxmin strategies. □

## V. EXPERIMENTAL RESULTS

In this section we discuss some experimental issues related to the proposed approach. The computation of the optimal patrolling strategy, namely the solution of the mathematical programming problems of Section III-C, has been obtained by defining our model with AMPL [17] and using the SNOPT 7.2 [6] solver on a Pentium R 3 GHz 1 GB RAM Linux computer. The average time for solving each bilinear programming problem of Figs. 2 and 3 is 7 s and 4.9 s, respectively. The dimension of each bilinear programming problem, and consequently the time needed to solve it, grows with  $n$  and  $d$  and decreases with  $r$ . For example, considering different settings with  $r = 1$ , average computational times for each bilinear programming problem range from 0.05 s when  $n = 5$ , to 4.33 s when  $n = 10$ , and to 80 s when  $n = 20$ . Moreover, considering the problem in Fig. 1 with  $r = 1$  and changing  $d$  from 1 to 10, the average computational time grows linearly from 0.5 s to 1 minute for each bilinear programming problem. Qualitatively speaking, our computational times are of the same order of magnitude of those reported in [7] for similar scenarios. Still from a qualitative point of view, the number of cells ( $n = 20$ ) our approach can currently manage in reasonable time compares well to the “manageable” number of houses to patrol in [7] ( $m = 7$ , but with more adversarial types) and to the “manageable” distance between two consecutive robots in [3] ( $\delta = 16$ , where this distance plays a role similar to the number of areas to patrol).

Given a patrolling scenario modeled according to our approach, the patrolling strategy we derive is *optimal* for the scenario, since it gives the patroller the maximum expected utility. Hence, there is little point in comparing our approach

with other simpler patrolling strategies, for example with random movements. We have already shown that our approach generalizes the approach in [3]. Moreover, the comparison of the patrolling strategy produced with our approach with that produced with the approach in [4] is unfair since the latter model does not consider the topology of the environment. Hence, we present an experimental analysis of the parameters  $d$  and  $r$ , that play an important role in our approach.

Let us consider the environment depicted in Fig. 1 and assume that the penetration times are all equal,  $d_i = d \forall i \in C$ . Given a single patroller able to sense only its current cell ( $r = 1$ ), we can observe that if  $d$  is larger than an upper bound  $d \geq d_u^1 = 12$  the intruder will always be captured with a deterministic strategy. The same holds, when  $r = 2$ , for  $d \geq d_u^2 = 8$ . Furthermore, when  $r = 1$  and  $d$  is smaller than a lower bound  $d \leq d_l^1 = 4$  there is at least one cell  $i$  that can be successfully penetrated by the intruder regardless of the patrolling strategy. For example, assuming  $d = 2$ , the action *enter-when*(06,03) is always successful for the intruder since it is impossible for the patroller to reach cell 06 starting from cell 03 in  $h \leq 2$  steps, namely before the intrusion is successfully ended. Also when  $r = 1$  and  $d = 4$ , the optimal patroller strategy cannot prevent the intrusion and focuses only on cells 06 and 12, obtaining the patroller an expected utility of 0.8, corresponding to the fact that the intruder enters cell 04. Similarly, when  $r = 2$ , for  $d \leq d_l^2 = 3$  the intrusion is unavoidable. Finally, when  $r = 1$ , we found that for  $d = 11$  the optimal patrolling strategy forced the intruder to never attempt to enter (according to Section III-C, the first stage of our algorithm found a patrolling strategy for which *stay-out* is the intruder's best response). (Note that the above considerations relative to bounds  $d_u^r$  and  $d_l^r$  can be generalized to other patrolling scenarios.) The trend of the patroller's expected utility with respect to  $d$  is shown in Fig. 6, for different sensing models ( $1 \leq r \leq 4$ ). When the value of  $d$  increases (modeling a weaker intruder) the patroller's expected utility increases. Furthermore, augmenting the sensing capabilities of the patroller (from  $r = 1$  to  $r = 4$ ) leads to an improvement of its expected utility.

## VI. CONCLUSIONS AND FUTURE WORKS

In this paper we have presented an approach to develop optimal strategies for patrolling mobile robots. These strategies have been derived by considering a model of the adversary within a game theoretic framework. Our approach extends [3], by considering more general environments, and [4], by considering the topology of the environment. Experimental results show the viability of our approach.

The work presented here can be improved in many ways. For example, in order to address environments of a more realistic size, more efficient algorithms for solving the multi bilinear programming problems of Section III-C are required [18]. Moreover, the environment model and the movement and sensing models of the patrolling robot could be refined by introducing metric aspects. In general, the applicability of the proposed model to real scenarios will be the goal of future work. Finally, the extension of the proposed

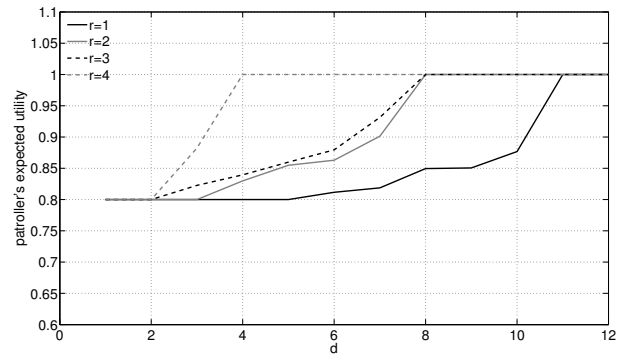


Fig. 6. Patroller's expected utility vs.  $d$  and  $r$

model to a general multirobot setting in which the robots are not synchronized requires further investigation.

## REFERENCES

- [1] L. Martins-Filho and E. Macau, "Patrol mobile robots and chaotic trajectories," in *Mathematical Problems in Engineering*. Hindawi, 2007.
- [2] S. Ruan, C. Meirina, F. Yu, K. Pattipati, and R. Popp, "Patrolling in a stochastic environment," in *Proc. Int'l Command and Control Research Symposium*, 2005.
- [3] N. Agmon, S. Kraus, and G. Kaminka, "Multi-robot perimeter patrol in adversarial settings," in *Proc. ICRA*, 2008, pp. 2339–2345.
- [4] P. Paruchuri, J. Pearce, M. Tambe, F. Ordonez, and S. Kraus, "An efficient heuristic approach for security against multiple adversaries," in *Proc. AAMAS*, 2007, pp. 311–318.
- [5] D. Fudenberg and J. Tirole, *Game Theory*. The MIT Press, 1991.
- [6] Stanford Business Software Inc., "http://www.sbsi-sol-optimize.com/."
- [7] P. Paruchuri, J. Pearce, J. Marecki, M. Tambe, F. Ordonez, and S. Kraus, "Playing games for security: An efficient exact algorithm for solving Bayesian Stackelberg games," in *Proc. AAMAS*, 2008, pp. 895–902.
- [8] P. Paruchuri, M. Tambe, F. Ordonez, and S. Kraus, "Security in multiagent systems by policy randomization," in *Proc. AAMAS*, 2006, pp. 273–280.
- [9] F. Amigoni, N. Gatti, and A. Ippedito, "A game-theoretic approach to determining efficient patrolling strategies for mobile robots," in *Proc. IAT*, 2008, pp. 500–503.
- [10] V. Isler, S. Kannan, and S. Khanna, "Randomized pursuit-evasion in a polygonal environment," *IEEE T ROBOT*, vol. 5, no. 21, pp. 864–875, 2005.
- [11] R. Vidal, O. Shakernia, J. Kim, D. Shim, and S. Sastry, "Probabilistic pursuit-evasion games: Theory, implementation and experimental results," *IEEE T ROBOTIC AUTOM*, vol. 18, no. 5, pp. 662–669, 2002.
- [12] B. von Stengel and S. Zamir, "Leadership with commitment to mixed strategies," London School of Economics, CDAM Research Report LSE-CDAM-2004-01, 2004.
- [13] J. Pita, M. Jain, J. Marecki, F. Ordonez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus, "Deployed armor protection: the application of a game theoretic model for security at the Los Angeles International Airport," in *Proc. AAMAS*, 2008, pp. 125–132.
- [14] N. Gatti, "Game theoretical insights in strategic patrolling: Model and algorithm in normal-form," in *Proc. ECAI*, 2008, pp. 403–407.
- [15] V. Conitzer and T. Sandholm, "Computing the optimal strategy to commit to," in *Proc. EC*, 2006, pp. 82–90.
- [16] Y. Shoham and K. Leyton-Brown, *Multiagent Systems: Algorithmic, Game Theoretic and Logical Foundations*. Cambridge University Press, 2008.
- [17] R. Fourer, D. Gay, and B. Kernighan, "A modeling language for mathematical programming," *Management Science*, vol. 36, no. 5, pp. 519–554, 1990.
- [18] N. Basilico, N. Gatti, and F. Amigoni, "Leader-follower strategies for robotic patrolling in environments with arbitrary topologies," in *Proc. (AAMAS)*, 2009.