

# Capturing Augmented Sensing Capabilities and Intrusion Delay in Patrolling-Intrusion Games

Nicola Basilico, Nicola Gatti, and Thomas Rossi

**Abstract**—Patrolling-intrusion games are recently receiving more and more attention in the literature. They are two-player non zero-sum games where an intruder tries to attack one place of interest and one patroller (or more) tries to capture the intruder. The patroller cannot completely cover the environment following a cycle, otherwise the intruder will successfully strike at least a target. Thus, the patroller employs a randomized strategy. These games are usually studied as leader-follower games, where the patroller is the leader and the intruder is the follower. The models proposed in the state of the art so far present several limitations that prevent their employment in realistic settings. In this paper, we refine the models from the state-of-the-art capturing patroller's augmented sensing capabilities and a possible delay in the intrusion, we propose algorithms to solve efficiently our extensions, and we experimentally evaluate the computational time in some case studies.

## I. INTRODUCTION

Patrolling-intrusion games are recently receiving more and more attention in the literature [1], [3], [5], [13]. These games are two-player (i.e., the patroller and the intruder) where the patroller tries to patrol an environment divided in some areas and the intruder tries to attack an area with some value. Patrolling situations are very common in strategic games and in real world applications. The scientific interest over them rises when no deterministic patrolling strategy can assure that intruder abstains from attacking and thus the patroller must follow a non-deterministic strategy, randomizing over the possible areas.

Customarily, their analysis is carried on with game theory tools. According to game theory we distinguish the *mechanism* from the *strategies* [9]. The mechanism defines the rules of the game, establishing the number of the players, prescribing the available actions and the sequential structure, and providing the mapping between players' actions and outcomes. The strategies define the specific behavior of the players during the game. Given the mechanism, each player, if it is *rational*, would act trying to maximize its own revenue. If all the players are rational, their strategies are in equilibrium. The basic concept of equilibrium is Nash equilibrium. Solving a game means to compute players' equilibrium strategies.

Different mechanisms were proposed for patrolling-intrusion games. For instance, in [1] the authors proposed a zero-sum game where the environment is a perimeter, in [13] the authors proposed a strategic-form game with

incomplete information, and in [3] the authors proposed an extensive-form game and subsequently approximated it as a repeated strategic-form game. The main problem behind the patrolling problem is to capture satisfactorily the topology of the environment along which the patroller moves. Recently, the work proposed in [5] allows one to model patrolling situations where the topology can be arbitrary. This model assumes that the intruder can observe the patroller's strategy before acting and this introduces a leader-follower position in the game where the leader is the patroller and the follower is the intruder. Acting as a leader is theoretically proved to provide a non shorter revenue to a player. In these situations the appropriate equilibrium concept is the *leader-follower equilibrium* [17]. The resolution of a game is usually tackled by resorting to operational research tools.

The models proposed in the literature so far provide a grain description of the patrolling situations, not assuring effective strategies in real-world settings. In this paper, we consider the most general model presented in the literature [5] and we provide two different extensions with the aim to make the patrolling model closer to real-world situations. The first extension captures patroller's augmented sensing capabilities and refines the intruder's movement model. In the previous models [1], [5], [13], the patroller is assumed to be able to sense only its current area and the intruder is assumed to strike a target directly appearing on it. In our extension, we allow the patroller to sense other areas in addition to the one in which it is (the sensors can present uncertainty) and we force the intruder to reach the target moving along arbitrary paths. We state a mathematical programming problem to capture our extension and we reduce the search space, removing all the dominated intruder's actions (i.e., actions that the intruder would never play independently of the patroller's strategy). This reduction is based on the game theoretic *iterated dominance* [9] to the case of patrolling. The second extension captures the delay between the turn at which the intruder decides to enter and the turn at which the intruder actually enters the environment. That is, the intruder needs to spend a lot of time to reach the environment from the observation point. In the previous models, no delay was considered. Our extension introduces imperfect information in the game. Also in this case, we state a mathematical programming problem to capture our extension and we provide some algorithms to reduce the search space. From the experimental point of view, our reducing algorithms are very effective, allowing one to save more than 90% computational time, and, although our models extend the state-of-the-art, the time needed for solving them is not larger

The authors are with the Dipartimento di Elettronica e Informazione of the Politecnico di Milano, Milano, Italy, {basilico, gatti}@elet.polimi.it, thomas.rossi@mail.polimi.it

than the one for the original model.

The paper is structured as follows. The next section surveys the game theoretic strategic patrolling models and overviews our contributions. Section 3 presents our first extension. Section 4 presents our second extension. Section 5 closes the paper.

## II. STATE OF THE ART

### A. Patrolling-Intrusion Games

A patrolling situation is characterized by one or more *patrollers*, a possible *intruder*, and some *targets*. The targets are areas with some interest for both patrollers and intruder. The interest for a patrolling situation rises 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 (i.e., a repeated cycle) for their movements. As a result, they should randomize over the targets trying to minimize the intrusion probability of the intruder [1], [5]. Patrolling-intrusion games come from the class of *hide-and-peek* games. The basic game, proposed by von Neumann [7], is a strategic-form game where the hider (intruder) chooses an area wherein to hide and the seeker (patroller) chooses a fixed number of areas in which to seek the hider. An extension is provided by the author in [10] where the game is repeated and the seeker can move along a fully connected graph. In [11], the authors extend the model to settings where the graph is not fully connected and the time horizon is finite. These models are coarse descriptions of patrolling-intrusion situations. Indeed, in such situations agents can have different preferences, time horizon is considered to be infinite (the patroller could continuously patrol), and the patroller can act as a leader and the intruder can act as follower, observing the strategy of the leader and acting on the basis of its observation. Three are the main models proposed in the literature for patrolling-intrusion games. We briefly review them.

The first model is proposed in [14] deals with 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, i.e., the time needed by the intruder to enter an area. No topology connecting the targets is considered. The authors model such a problem as a two-player (i.e., the patroller and the intruder) strategic-form game with incomplete information (i.e., the intruder's preferences over the areas can be uncertain to the patroller) [9]. The actions available to the patroller are all the possible routes of  $d$  areas, while the intruder chooses 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 [17].

The second model is proposed in [1]. The problem considered in this case is to patrol a perimeter divided in cells by employing a team of synchronized mobile robots moving

by at most one cell at each discrete turn. The proposed patrolling strategy is the one that maximizes the minimum expected utility for the patrollers or, equivalently, that maximizes the detection probability over the cells. This work is applicable to very special ring-like environments where the penetration time is the same for all the cells and the patrollers have no preferences over them.

The third model is proposed in [5]. The authors propose a model (from here BGA) which generalizes the two previous models: agents' preferences are explicitly taken into account and targets are connected by an arbitrary topology. The authors model this situation as a leader-follower game. In the present paper we build on this last model, being the most general one.

### B. Basic BGA Model

We describe the basic BGA model [5]. The environment is composed of a set  $C = \{c_1, \dots, c_n\}$  of cells to be patrolled, whose topology is modeled by a directed connected graph  $\mathcal{G}$ . We represent  $\mathcal{G}$  by a matrix  $G(n \times n)$ , where  $G(c_i, c_j) = 1$  if cells  $c_i$  and  $c_j$  are adjacent and  $G(c_i, c_j) = 0$  otherwise. A subset  $T \subseteq C$  contains all the cells that have some value for both patroller and intruder. We call these cells *targets*. We suppose that time is discrete and that each  $c_i$  requires the intruder  $d_i > 0$  turns to enter ( $d_i$  is called the *penetration time* for cell  $c_i$ ).

In one turn the patroller can move between two adjacent cells in  $\mathcal{G}$  and patrols the destination cell. The intruder can directly enter any cell  $c_i$  and, once entered, it must wait  $d_i$  turns. The patroller can sense only the cell in which it currently is (extensions can be found in [4], [5]).

The scenario is modeled as a leader-follower situation where the intruder can perfectly observe the patroller's actions and can act on the basis of its observation. As discussed in [5], such situation can be modeled as a strategic-form game [9] where the patroller chooses its patrolling strategy, namely, the set of probabilities  $\alpha_{i,j,s}$  with which it moves from  $c_i$  to  $c_j$  (the patroller's strategy is assumed to be Markovian), and the intruder chooses what cell to enter and when, namely, *enter-when*( $c_i, c_j$ ) meaning that the intruder enters  $c_i$  at the time point after it observed the patroller in  $c_j$ . When the intruder attempts to enter  $c_i$ , if the patroller visits  $c_i$  before  $d_i$  turns, then the outcome is *intruder-capture*, the outcome is *penetration- $c_i$*  otherwise; when the intruder performs *stay-out* action and does not enter any cell, then the outcome is *no-attack*.

Agents' payoffs are defined as follows. We denote by  $X_i$  and  $Y_i$  the payoffs to the patroller and to the intruder, respectively, when the outcome is *penetration- $c_i$* . 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 the values of all the cells.) Consistency constraints over these values

are:  $X_i < X_0$  and  $Y_0 \leq 0 < Y_i$  for all  $c_i \in T$ , and  $X_i = X_0$  and  $Y_i = 0$  for all  $c_i \in C/T$ .

According to game theory, a solution for the game we 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. In our case, the appropriate equilibrium concept is the *leader-follower equilibrium* [17]. This equilibrium gives the leader the maximum expected utility when the follower observes the leader's strategy and acts as a best responder. In the above patrolling game, the problem of computing the equilibrium strategies can be formulated as a multiple mathematical programming problem and solved by standard solvers [8], [15]. More precisely, at first we search for a patrolling strategy such that *stay-out* is a best response for the intruder. This can be formulated as a bilinear feasibility problem. If no such strategy exists, then the optimal patroller's strategy is found by computing, for each possible intruder's best response *enter-when*, the best patroller's strategy (this is a single bilinear optimization problem). Among the found strategies, the one which gives the largest expected utility to the patroller is selected.

### C. Extension Overview and Motivations

BGA provides a fine patrolling model. However, a large amount of details are neglected. In [6], we have extended the BGA model capturing the situation where the intruder cannot perfectly observe the patrolling environment. In this paper, we refine the BGA model along two different directions that are not studied in depth in the game theoretic patrolling literature [1], [5], [13]. For the sake of presentation, we present the two extensions by degrees. Our first extension refines both the patroller and the intruder's models. More precisely, we introduce patroller's augmented (also uncertain) sensing capabilities and we model the movement of the intruder allowing it to move along paths connecting the target to an access area. To the best of our knowledge, the patroller's augmented capabilities are studied only in [2]. This approach is not applicable to BGA, because it was developed *ad hoc* for the model introduced in [1] and cannot be easily generalized to richer models such as BGA. Furthermore, in the literature the intruder is considered to appear directly on the target. Consider Fig. 1: numbered cells are the areas to be patrolled, black cells denote obstacles, gray circles denote targets, black rectangles denote access areas, and gray cells are empty cells outside the patrolling environment. When patroller's sensing capabilities are not augmented, the patroller can sense only its current cell. With augmented sensing capabilities, for instance, when the patroller is in cell 04 can sense cell 04 with probability of one, cell 03 with probability of 0.5, and so on. The gray lines denote possible paths connecting the access point in cell 03 with targets in cells 06 and 07. Our second extension captures the situation wherein there is a delay between the time point in which the intruder decides to enter and the time point in which it actually enters a target. Such a delay is common in realistic settings. For instance, an intruder can decide to attack a target once it has observed the patroller

in cell 01, but it needs to spend some turns before reaching the access area, e.g., placed in cell 03.

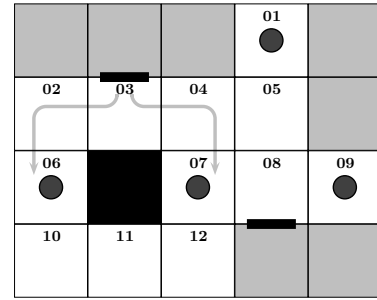


Fig. 1. A patrolling scenario: numbered cells are the areas to be patrolled, black cells are obstacles, gray circles denote targets, black rectangles denote access areas, and gray cells are empty cells outside the patrolling environment. The gray lines are paths connecting access areas 03 to target 06 and 07.

## III. CAPTURING PATROLLER'S AUGMENTED SENSING CAPABILITIES AND INTRUDER'S MOVEMENT

### A. Mathematical Model

We enrich BGA model along two directions. On the one hand we enrich the patroller model by augmenting its sensing capabilities. On the other hand we enrich the intruder model, by introducing access areas, constraining the intruder to move along paths connecting access areas to targets, and allowing the patroller to detect the intruder also along these paths. For the sake of presentation, we assume that the patroller can disappear from the target after a number of turns equal to the penetration time. (Capturing situations in which the patroller moves along paths when it leaves the environment is an easy extension of the result presented in this section.) As is customary in the literature, we assume that the intruder can move infinitely fast. Therefore, covering an entire path takes only one turn, regardless of its length. As a result, when the intruder enters, it can be captured along the entire path and not only in the target it entered. Technically speaking, if the intruder attacks target  $t_i$  at time  $\bar{k}$ , then at time  $\bar{k}$  it can be detected on every cell of the path (including both target and access area), while in the remaining time interval  $[\bar{k} + 1, \bar{k} + d_i - 1]$  it can be detected only in  $t_i$ . The intruder is considered to be captured once it has been detected. Formally, we enrich the mathematical model by introducing:

- matrix  $S(n \times n)$ , where  $S(c_i, c_j)$  is the probability that the patroller, from cell  $c_i$ , detects an intruder in cell  $c_j$ ,
- $A \subseteq C$ , where  $A = \{a_1, \dots, a_l\}$ , is the set of cells containing at least an access area,
- $p_{c_i, c_j}^k$  is the  $k$ -th path connecting cell  $c_i$  to cell  $c_j$ .

Notice that the employment of matrix  $S$  allows one to capture very general settings. For instance, when the patroller can detect the intruder only in its current cell and makes that with probability of one,  $S$  is the identity matrix. In realistic settings,  $S(y, y) = 1$  and the value of  $S(y, z)$  decreases as  $z$  gets far from  $y$ . For instance, consider Fig. 1, we have  $1 = S(01, 01) > S(01, 05) > S(01, 08)$  and so on. We need

also to redefine intruder's space of actions. Called  $P$  the set of paths connecting all the possible access areas with all the possible targets, the actions  $enter\text{-}when(\cdot, \cdot)$  are defined as  $enter\text{-}when(p_{a_i, t_j}, c_l)$  for all  $a_i \in A$ ,  $t_j \in T$ , and  $c_l \in C$ , meaning that, at the time point after the patroller is in cell  $c_l$ , the intruder will attack target  $t_j$  from access area  $a_i$  covering path  $p_{a_i, t_j}$ . We use the operator  $last(p)$  to denote the last cell of path  $p$ , e.g.,  $last(p_{a_i, t_j}) = t_j$ . We can now extend the mathematical programming formulation. At first, we search for a patroller's strategy such that  $stay\text{-}out$  is the intruder's best response. Such strategy (if it exists) is the optimal patrolling strategy and it can be found solving the following bilinear feasibility problem ( $\alpha_{i,j}$ s are defined as in Section II-B,  $\gamma_{i,j}^{h,p}$  denotes the probability that the patroller reaches  $c_j$  in  $h$  steps, starting from  $c_i$  and not detecting the intruder when this makes  $p$ ):

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

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

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

$$\gamma_{i,j}^{1,p} = \left( \prod_{c_y \in p} (1 - S(c_j, c_y)) \right) \alpha_{i,j} \quad \forall p \in P, c_i \in C, c_j \in C \setminus p \quad (4)$$

$$\gamma_{i,j}^{h,p} = (1 - S(j, last(p))) \cdot \sum_{c_x \in C \setminus last(p)} (\gamma_{i,x}^{h-1,p} \alpha_{x,j}) \quad \forall h \in \{2, \dots, d_{last(p)}\}, p \in P, c_i, c_j \in C, c_j \neq last(p) \quad (5)$$

$$Y_0 + (Y_{last(p)} - Y_0) \cdot \sum_{c_i \in C \setminus last(p)} \gamma_{z,i}^{d_{last(p)},p} \leq 0 \quad \forall c_z \in C, p \in P \quad (6)$$

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 cells; constraints (4) express the probability that the patroller can sense the intruder in the path. We provide some details. The probability that the patroller, from cell  $c_j$ , detects the intruder along path  $p$  is given by the probability of several non-independent events, i.e., to detect the patroller in each cell of  $p$ . However, by using De Morgan rule, we can express such probability as the product of probability of independent events. More precisely,  $1 - S(c_j, c_y)$  is the probability that the patroller, from cell  $c_j$ , does not detect the intruder in cell  $c_y$ , and thus  $\prod_{c_y \in p} (1 - S(c_j, c_y))$  is the probability that the patroller, from cell  $c_j$ , does not detect the intruder in any cell of path  $p$ . Constraints (5) express the Markov hypothesis over the patroller's decision policy (i.e., the patroller's randomization probabilities depend only on the cell wherein it is). Notice that in this case, differently from constraints (4), it is considered only the probability to detect the intruder in the target, i.e.,  $S(j, last(p))$ . Constraints (6) express that no action  $enter\text{-}when(p, z)$  gives the intruder an expected utility larger than that of  $stay\text{-}out$ . The non-linearity is due to constraints (5). If the above problem admits a solution, the resulting  $\alpha_{i,j}$ s constitute the optimal patrolling strategy, assuring the intruder not to enter any target. When the above problem is unfeasible, we pass to the second stage of the algorithm. For each possible action  $enter\text{-}when(q, z)$ , we compute the patroller's best strategy subject to  $enter\text{-}when(q, z)$  is a best response for the intruder. These strategies can

be computed as a bilinear optimization problem, where the objective function is the patroller's expected utility:

$$\max \quad X_0 + (X_{last(q)} - X_0) \sum_{c_i \in C \setminus last(q)} \gamma_{z,i}^{d_{last(q)},q}$$

s.t.

constraints (1)-(6)

$$Y_0 + (Y_{last(q)} - Y_0) \sum_{c_i \in C \setminus last(q)} \gamma_{z,i}^{d_{last(q)},q} \geq Y_0 + (Y_{last(p)} - Y_0) \sum_{c_i \in C \setminus last(p)} \gamma_{w,i}^{d_{last(p)},p} \quad \forall p \in P, c_w \in C \quad (7)$$

Constraints (10) express that no action  $enter\text{-}when(p, w)$  gives to the intruder an expected utility larger than that of  $enter\text{-}when(q, z)$ . The patroller's optimal strategy is the strategy that assures the largest expected utility among all the ones computed above.

## B. Search Space Reduction

Computing a leader-follower equilibrium in our game model requires a significant computational effort, especially considering that the number of mathematical programming problems to be solved is large. However, a remarkable amount of computational time can be saved by simplifying the game in a pre-processing phase. The approach we followed to obtain such complexity reduction takes inspiration from the game theoretic iterated dominance [9] where the underlying assumption is that a player's action  $a$  is dominated by another action  $a'$  if the former gives the player an expected utility never larger than the latter independently of the opponent's strategy. Therefore, exploiting this technique, we can exclude all the patroller's and intruder's actions that they would never play.

In order to reduce the patroller's action space we remove from the environment a subset of cells that the patroller can safely exclude from its routes. We accomplish this in two steps. In the first step we denote  $d(c_i, c_j)$  the minimum distance between cells  $c_i$  and  $c_j$  and we remove all cells  $c_x$  such that  $d(c_x, c_t) > d_t$  for at least one target  $c_t \in T$ . Indeed, moving in such cells will guarantee to the patroller that at least one target will be successfully penetrated. It can be easily shown that if the new obtained graph is not connected, then no patrolling strategy can be effective in the original graph. Thus, with this step we can also assess if the problem is well posed or not.

In the second step we further reduce the set of cells starting from the basic assumption that moving along minimum paths cannot penalize the patroller's expected utility. We call  $R_{i,j} = \{r_{i,j}^k\}$  the set of minimum paths connecting two targets  $c_i, c_j \in T$ . Since for a pair of targets  $c_i, c_j$  there can be in general more than one minimum path connecting them, we call the  $k$ -th minimum path  $r_{i,j}^k$  while the number of those minimum paths will be denoted by  $|R_{i,j}|$ . For every pair of targets  $c_i, c_j$  we select one or more shortest paths and we constrain the patroller's movements over cells belonging to the selected paths. To accomplish this, we consider every minimum path  $r_{i,j}^k$  and we select it if and only if at least one of the following conditions holds:

- $|R_{i,j}| = 1$ , i.e., the minimum path between  $c_i$  and  $c_j$  is unique,
- it does not exist a path  $r_{i,j}^{k'} \in R_{i,j}$  (with  $k \neq k'$ ) such that for every target  $c_t$  it holds that  $\max_{c_i \in r_{i,j}^{k'}} d(c_i, c_t) \leq \min_{c_j \in r_{i,j}^k} d(c_j, c_t)$ .

When the second condition is not verified,  $r_{i,j}^k$  is dominated by  $r_{i,j}^{k'}$  and therefore it will not be selected. In other words, a dominated path for the patroller is a path that introduces unnecessary traveling costs with respect to the distance from all the targets of the environment. It is straightforward to see that using only non dominated paths cannot decrease the expected utility of the patroller. Last, we notice that first condition is reported only for the sake of clarity, since its validity logically implies the validity of the second one.

When extending the patroller's sensing capabilities the environment can be further reduced. Since the patroller can sense a subset of cells in the proximity of the current cell, it can avoid to visit a target limiting its movements to cells from where that target can be sensed. Therefore, we can modify the first step of the patroller's action space reduction, performing a selection of the minimum connected set of cells such that every target can be sent from at least one cell of such subset.

In synthesis, assuming that the problem is well posed, patroller's actions are restricted to movements leading in cells that are not discarded in the first step and that belong to at least a path selected by the second one.

Similarly to the patroller's case, the reduction of the intruder's action space is decomposed in two steps. Recalling that an intruder's action is denoted by *enter-when*( $p, c$ ), in the first step we determine the minimal set of paths  $ps$  that the intruder would select. In the second one, for each selectable  $p$ , we determine the minimal set of cells  $cs$  that the intruder would consider in its actions *enter-when*( $p, c$ ).

We focus on the first step. Although the set of paths is in principle infinite, the paths along which a rational intruder would move are finite. The literature provides a large number of works dealing with path finding (e.g., see [16]), usually based on A\* [12]. They focus on searching for the path with minimum cost. In our case the problem of finding the path with the minimum cost (i.e., the minimum capture probability) is not well defined, not being known the patroller's randomization probabilities (they constitute the solution). Nevertheless, a number of paths can be discarded, never being played by the intruder independently of the patroller's strategy. The basic idea is that a rational intruder would try to consider minimal paths for each target to reduce the probability of being captured. We say that, given  $t_j \in T$ , path  $p_{a_i, t_j}$  is *irreducible* if there not exists any  $p'_{a_k, t_j}$  (with  $k$  not necessarily different from  $i$ ) such that all the cells of  $p'_{a_k, t_j}$  are strictly contained in  $p_{a_i, t_j}$ . Consider Fig. 1,  $p_{03, 06} = \langle 03, 02, 06 \rangle$  is strictly contained in  $p'_{03, 06} = \langle 03, 04, 03, 02, 06 \rangle$ ; it can be easily seen that  $p_{03, 06} = \langle 03, 02, 06 \rangle$  is irreducible. Action *enter-when*( $p_{a_k, t_j}, c$ ) is dominated by *enter-when*( $p_{a_i, t_j}, c$ ) when  $p_{a_i, t_j}$  is contained in  $p'_{a_k, t_j}$ . The proof is easy: the intruder's expected utility depends on the value of a target

and on the capture probability. When two paths have the same target, the intruder's expected utility depends only on the capture probability. When a path  $p_{a_i, t_j}$  is contained in a path  $p'_{a_k, t_j}$ , the intruder's probability to be captured when it follows  $p_{a_i, t_j}$  is not larger than the one when it follows  $p'_{a_k, t_j}$ . Thus, we can safely consider only irreducible paths. Fig. 1 depicts some irreducible paths.

The set of the irreducible paths can be computed building a tree of paths where each node  $v$  represents a cell of the environment. Let us call  $\eta(v) \in C$  the cell represented by a node  $v$ . For each target  $t_j$ :

- 1) build the tree of paths with root  $v_0$ , where  $\eta(v_0) = t_j$ ; given a node  $v$ ,  $\eta(v) \in A$  (recall that  $A$  is the set of access areas) iff  $v$  is a terminal node;  $v''$  is a successor of  $v'$  iff  $v'' \in \{v | G(\eta(v'), \eta(v)) = 1\}$  and  $v''$  never appears in the branch from  $v_0$  to  $v'$ ;
- 2) for each pair of paths ( $p', p''$ ) from the root to leafs if  $p'$  is strictly contained  $p''$  then remove  $p''$ .

Fig. 2 reports an example of the tree of paths with target 01.

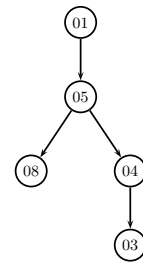


Fig. 2. Tree of paths for computing the set of irreducible paths with target 01.

We focus on the second step. For each irreducible path  $p$ , we compute the cells  $cs$  such that action *enter-when*( $p, c$ ) is not dominated. Fixed a path  $p$ , *enter-when*( $p, c$ ) is dominated by another intruder's action (independently of the patroller's strategy) in two possible cases as follows.

- If cell  $c$  is such that the patroller will be at the next turn on a cell from which it will observe a cell belonging to  $p$  with probability of one, then *enter-when*( $p, c$ ) is dominated by *stay-out*, e.g., consider Fig. 1 when the patroller can sense with probability of one only its current cell, if  $p = \langle 03, 02, 06 \rangle$  and  $c = 02$ , the intruder will be surely detected by making *enter-when*( $p, c$ ).
- *enter-when*( $p, c$ ) is dominated by *enter-when*( $p, c'$ ) if there exists a cell  $c'$  such that the two following conditions hold. 1) From  $c'$  the patroller cannot reach in one turn any cell from which it can observe at least a cell of  $p$  with positive probability. For instance, consider Fig. 1 and suppose that the patroller is in 08 and  $S(08, 08) = 1$  and  $S(08, 05) = S(08, 07) = S(08, 09) = 0.5$ . Suppose that  $p = \langle 03, 02, 06 \rangle$ . From  $c' = 08$  the patroller cannot reach in one turn any cell from which it can observe at least a cell of  $p$  with positive probability. 2) The patroller, when it starts from  $c'$  and reaches any cell from which it can observe with probability of one  $last(p)$  in at most  $d_{last(p)}$  turns, must visit cell  $c$ . For

instance, consider Fig. 1 and suppose that  $S(x, y) = 1$  only if  $x = y$ . If  $p = \langle 03, 02, 06 \rangle$  and  $d_{06} = 6$ ,  $enter\text{-}when(p, 03)$  is dominated by  $enter\text{-}when(p, 01)$ , indeed, if the patroller starts from 01, it must visit cell 03 to reach 06 by  $d_{06} = 6$ . Therefore the probability that the patroller observes cell 06 in  $d_{06} = 6$  from cell 01 is not larger than the same probability from cell 03.

The set  $\{c\}$  of cells such that action  $enter\text{-}when(p, c)$  is not dominated can be found resorting again to tree search techniques. We build a tree of paths where, called  $v_0$  the root,  $\eta(v_0) = last(p)$  and a node  $v''$  is a successor of  $v'$  iff  $v'' \in \{v | G(\eta(v'), \eta(v)) = 1\}$  and  $v'' \neq v'$  and  $v'' \neq father(v')$ . Each branch in the tree can have a different depth. Exactly, consider a branch, say  $l$ , and consider the deepest node in this branch from which the patroller can observe the root node (a target) with probability one, say  $x$ . The depth of  $l$  is given by the depth of  $x$  plus  $d_{last(p)}$ . That is, the tree represents all the paths that lead in at most  $d_{last(p)}$  turns to a cell from which the patroller can observe with probability of one the target. Fig. 3 reports the tree of paths generated as prescribed above with  $p_{03,06} = \langle 03, 02, 06 \rangle$  with  $d_{06} = 5$ . Given a tree of paths so built, dominance can be found as follows:

- 1) for every node  $v$ , if  $\eta(v) \in p$  and for every cell  $c$  such that  $G(\eta(v), c) = 1$  it holds that  $c \in p$ , then  $\eta(v)$  is to be discarded, e.g., in Fig. 3, given  $p_{03,06} = \langle 03, 02, 06 \rangle$ , cells 02, is to be discarded;
- 2) if for every  $v$  such that  $\eta(v)$  and for every  $w$  such that  $\eta(w)$  it holds that  $v$  is ancestor of  $w$  and for every cell  $c$  such that  $G(\eta(w), c) = 1$  and for every  $z \in p$  it holds that  $S(c, z) = 0$ , then  $\eta(v)$  is dominated. For example, in Fig. 3 01 occurs only once and then all the nodes that precede 01 contain dominated cells; every occurrence of 09 is preceded by 08, and then 08 is dominated.

In Fig. 3, black nodes denote  $cs$  such that actions  $enter\text{-}when(p_{03,06}, c)$ s are dominated.

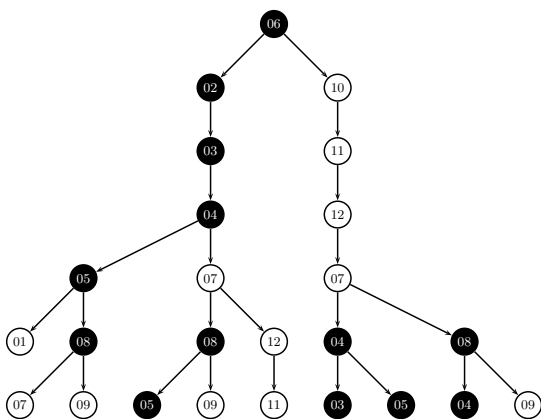


Fig. 3. Tree of paths for the computation of the dominated actions.

### C. Experimental Evaluation

We experimentally evaluate the computational time of our algorithms. We implemented the two algorithms described in

	original		with augmented sensing and paths	
	non-reduced	reduced	non-reduced	reduced
bilinear problems	25	15	224	44
total time	584.4 s	336.4 s	15525.0 s	440.0 s
average time	23.3 s	22.4 s	69.2 s	9.9 s
max time	28.2 s	27.2 s	77.7 s	11.76 s
min time	19.6 s	18.4 s	63.15 s	8.2 s

TABLE I  
EXPERIMENTAL EVALUATION FOR THE SETTING DEPICTED IN FIG. 1.

the previous section for reducing the search space in C. We solved the feasibility and the optimization problems by using SNOPT [15] through AMPL [8]. SNOPT is a commercial software particularly effective with bilinear programming problems. Our tests were executed on a UNIX computer with dual quad-core 2.33GHz CPU and 8GB RAM.

In Tab. I, we report the experimental results with the setting depicted in Fig. 1 with the following parameters:  $d_{01} = d_{06} = d_{09} = 6$  and  $d_{07} = 4$ ,  $X_{01} = 0.7$  and  $Y_{01} = 0.3$ ,  $X_{06} = 0.5$  and  $Y_{06} = 0.5$ ,  $X_{07} = 0.7$  and  $Y_{07} = 0.26$ , and  $X_{09} = 0.3$  and  $Y_{09} = 0.8$ . (The computational time spent for reducing the game results to be negligible, i.e.,  $< 1$  s, and then it is omitted.) We considered four configurations: the original patrolling problem (as formulated in [5]) with and without reductions and the problem with augmented sensing capabilities and paths (as formulated in Section 3.1); we evaluated this last configuration without any reduction (non-reduced), and with the reduction on the paths and the cells (reduced). In the non-reduced configuration, we considered all the paths without cycles. For each configuration, we report the number of bilinear problems to be solved, the time spent for solving the whole problem, and the average, max and min time in seconds of bilinear problems. The first result is that our reduction algorithms significantly reduce the number of bilinear problems and the computational time. The second result is that, although our model refines the state-of-the-art, it is no more computationally expensive than the state-of-the-art.

According to the above evaluation scheme, we evaluated 10 different patrolling settings with size (in terms of cells, targets, paths, and so on) comparable to the setting depicted in Fig. 1. The results, omitted for reasons of space, are close to the ones reported in Tab. I. The percentage of computational time saved thanks to our algorithms is in the range 97% – 99%. This confirms the effectiveness of our reduction algorithms. Our algorithm allows one to compute the optimal patroller’s strategy in concrete settings only offline. The computational time can be significantly reduced when the game is equivalent to a zero-sum game. In these situations, the patroller’s strategy can be computed by solving a unique optimization problem similar to the ones described in Section 3.1. The computational time required to solve such an optimization problem belongs to the range 9-15 s.

#### IV. CAPTURING INTRUSION DELAY

##### A. Mathematical Model

We enrich BGA model by introducing a delay, say  $z$ , between the turn in which the intruder decides to enter a given target and the turn in which it actually enters. During this time interval the intruder cannot observe the patroller. This raises several complications on the intruder's decisions, since it does not know the exact position of the patroller at the turn in which it actually enters.

The mathematical programming formulation must take into account that the intruder may not know with certainty the position of the patroller when it strikes. Technically speaking, the intruder must derive a probabilistic belief over the position of the patroller in order to compute its expected utility from making a given action  $a$ . However, this belief must be consistent with the patroller's strategy that, in its turn, is computed such that intruder's action  $a$  is a best response. Hence, since the computation of the intruder's beliefs and of the patroller's strategy depend each other, they cannot be computed separately.

Suppose that  $c_i$  is the current cell of the patroller, we denote by  $\beta_{i,j}^v$  the probability that the patroller is in  $c_j$  after  $v$  turns. That is, once the intruder decides to enter when the patroller is in  $c_i$ ,  $\beta_{i,j}^v$  gives the probability that the patroller is in  $c_j$  after  $v$  turns. The values  $\beta_{i,j}^1$  can be easily defined:

$$\beta_{i,j}^1 = \alpha_{i,j} \quad \forall c_i, c_j \in C \quad (8)$$

The values  $\beta_{i,j}^v$  with  $1 < v \leq z$  are defined as follows:

$$\beta_{i,j}^v = \sum_{c_k \in C} \beta_{i,k}^{v-1} \alpha_{k,j} \quad \forall c_i, c_j \in C, 1 < v \leq z \quad (9)$$

The values  $\beta_{i,j}^z$  provides the probability distributions over the cells at the turn in which the intruder actually enter. These must be considered in the computation of the intruder's expected utility. More precisely, the feasibility programming problem can be stated as follows:

constraints (8)-(9)

$$\alpha_{i,j} \geq 0 \quad \forall c_i, c_j \in C \quad (10)$$

$$\sum_{j \in C} \alpha_{i,j} = 1 \quad \forall c_i \in C \quad (11)$$

$$\alpha_{i,j} \leq G(c_i, c_j) \quad \forall c_i, c_j \in C \quad (12)$$

$$\gamma_{i,j}^{1,p} = \left( \prod_{c_y \in p} (1 - S(c_j, c_y)) \right) \cdot \sum_{c_x \in C} (\beta_{i,x}^z \alpha_{x,j}) \quad \forall p \in P, c_i \in C, c_j \in C \setminus p \quad (13)$$

$$\gamma_{i,j}^{h,p} = (1 - S(j, last(p))) \cdot \sum_{c_x \in C \setminus last(p)} (\gamma_{i,x}^{h-1,p} \alpha_{x,j}) \quad \forall h \in \{2, \dots, d_{last(p)}\}, p \in P, c_i, c_j \in C, c_j \neq last(p) \quad (14)$$

$$Y_0 + (Y_{last(p)} - Y_0) \cdot \sum_{c_i \in C \setminus last(p)} \gamma_{z,i}^{d_{last(p)},p} \leq 0 \quad \forall c_z \in C, p \in P \quad (15)$$

Constraints (10)-(12) are analogous to constraints (1)-(3) present in the feasibility problem stated in Section 3.1; constraints (13) are analogous to constraints (4), except that the position of the patroller is not certain, but it is

given by a probability distribution produced as prescribed by constraints (10)-(11); constraints (14)-(15) are analogous to constraints (5)-(6). If this problem does not admit any solution, we can search for the optimal patrolling strategy by solving the following optimization problem for each possible action  $enter\_when(t, s)$ :

$$\max \quad X_0 + (X_\omega - X_0) \sum_{c_i \in C \setminus c_\omega} \gamma_{s,i}^{d,\omega}$$

st.

constraints (8)-(15)

$$Y_0 + (Y_t - Y_0) \cdot \sum_{c_i \in C \setminus c_t} \gamma_{r,i}^{d,t} \leq 0 \quad \forall c_t \in T, r \in S \quad (16)$$

The meanings of the objective function and of constraints (16) are analogous to the corresponding ones stated in Section 3.1.

##### B. State Space Reduction

Analogously to what we did in Section 3.2, in this section we aim at removing actions that each agent would never play independently of its opponent's strategy. The extension to the case with delay is straightforward. More precisely, the reduction of the patroller's action space holds to be exactly as in the absence of delay. The unique extension concerns the determination of the intruder's dominated actions.

The two conditions for dominance introduced in Section 3.2 must reformulated as follows:

- If cell  $c$  is such that the patroller will be at the next  $z + 1$ -th turn on a cell from which it will observe a cell belonging to  $p$  with probability of one, then  $enter\_when(p, c)$  is dominated by  $stay\_out$ , e.g., consider Fig. 1 when the patroller can sense with probability of one only its current cell, if  $p = \langle 08, 07, 12, 11, 10, 06 \rangle$  and  $c = 11$  and  $z = 1$ , the intruder will be surely detected by making  $enter\_when(p, c)$ .
- $enter\_when(p, c)$  is dominated by  $enter\_when(p, c')$  if there exists a cell  $c'$  such that the two following conditions hold. 1) From  $c'$  the patroller cannot reach in  $z + 1$  turns any cell from which it can observe at least a cell of  $p$  with positive probability. For instance, consider Fig. 1 and suppose that the patroller is in 09 and  $S(c_i, c_j) = 1$  if the Manhattan distance between  $c_i$  and  $c_j$  is non-strictly shorter than 1. Suppose that  $p = \langle 03, 02, 06 \rangle$  and  $z = 1$ . From  $c' = 09$  the patroller cannot reach in two turns any cell from which it can observe at least a cell of  $p$  with positive probability. 2) The patroller, when it starts from  $c'$  and reaches any cell from which it can observe with probability of one  $last(p)$  in at most  $d_{last(p)} + z$  turns, must visit cell  $c$ . For instance, consider Fig. 1 and suppose that  $S(x, y) = 1$  only if  $x = y$ . If  $p = \langle 03, 02, 06 \rangle$  and  $d_{06} = 6$  and  $z = 1$ ,  $enter\_when(p, 08)$  is dominated by  $enter\_when(p, 09)$ , indeed, if the patroller starts from 09, it must visit cell 08 to reach 06 by  $d_{06} + 1 = 7$ . Therefore the probability that the patroller observes cell

	original		with augmented sensing and paths	
	non-reduced	reduced	non-reduced	reduced
bilinear problems	30	20	250	70
total time	726.0 s	377.2 s	18750.0 s	1372.2 s
average time	24.6 s	18.9 s	75.1 s	19.6 s
max time	28.1 s	22.2 s	112.44 s	60.0 s
min time	19.6 s	17.3 s	69.97 s	8.9 s

TABLE II  
EXPERIMENTAL EVALUATION FOR THE SETTING DEPICTED IN FIG. 1  
WITH  $z = 1$ .

	original		with augmented sensing and paths	
	non-reduced	reduced	non-reduced	reduced
bilinear problems	39	22	272	70
total time	932.5 s	509.4 s	24750.0 s	1564.8 s
average time	23.9 s	23.1 s	90.9 s	22.6 s
max time	30.0 s	29.9 s	120.3 s	67.2 s
min time	18.3 s	20.1 s	63.15 s	14.4 s

TABLE III  
EXPERIMENTAL EVALUATION FOR THE SETTING DEPICTED IN FIG. 1  
WITH  $z = 2$ .

06 in  $d_{06} + 1 = 7$  from cell 09 is not larger than the same probability from cell 08.

Dominated cells can be found by building the path tree (as is described in Section 3.2), but, in this case, the the depth of the tree is  $d_i + z$ .

### C. Experimental Evaluation

The implementation of the reducing algorithms and of the mathematical programming model has been accomplished as described in Section 3.3. In this case we evaluate the impact of delay both in the original BGA model (neglecting augmented sensing capabilities and paths) and the in the complete case described in Section 4.1. We report the experimental results for the setting depicted in Fig. 1 in Tab. II when  $z = 1$  and in Tab. III when  $z = 2$ . The case with  $z = 0$  is discussed in Section 3.3. Exactly as for the extension discussed in Section 3, in this case the reduction algorithm works very well, allowing one to save significantly computational time. However, as  $z$  increases, the reduction is less effective. This is because the number of dominated actions decreases.

Following the above evaluation scheme, we experimentally evaluated 10 different patrolling setting keeping  $z \in \{0, 1, 2\}$ . The results, omitted for reasons of space, are close to the ones reported in Tab. II and III. The percentage of computational time saved thanks to our algorithms is in the range 90%–94%. This confirms the effectiveness of our algorithm.

## V. CONCLUSIONS AND FUTURE WORKS

The problem of producing effective strategies for patrolling-intrusion games is receiving more and more attention. The models available in the literature provide coarse descriptions. In the paper we proposed two different extensions in the attempt to make patrolling models closer to real-world

applications. We enriched the model behind the patroller's sensing capabilities and behind the intruder's movement. For both these extensions we provided mathematical programming formulation, reducing algorithms, and an experimental evaluation. Our algorithms allow one to compute efficiently the patroller's strategy in concrete settings only offline, being the computational times for the case studies evaluated in the paper larger than a few seconds. Therefore, it is applicable to all the situations in which the environment topology is known *a priori*.

In future works, we shall explore two different routes. In the first one, we shall integrate our framework together with machine learning techniques. Our algorithms will detect whether or not the opponent is rational and in the negative case will try to exploit its behavior to improve the patroller's revenue. In the second route, we shall develop an algorithm to compute suboptimal strategies.

## REFERENCES

- [1] N. Agmon, S. Kraus, and G. Kaminka. Multi-robot perimeter patrol in adversarial settings. In *Proc. ICRA*, pages 2339–2345, 2008.
- [2] N. Agmon, S. Kraus, and G. Kaminka. Uncertainties in adversarial patrol. In *Proc. AAMAS*, Budapest, Hungary, May 10-15 2009.
- [3] F. Amigoni, N. Gatti, and A. Ippedico. A game-theoretic approach to determining efficient patrolling strategies for mobile robots. In *Proc. IAT*, pages 500–503, 2008.
- [4] N. Basilico, N. Gatti, and F. Amigoni. Finding the optimal strategies in robotic patrolling with adversaries in topologically-represented environments. In *Proc. ICRA*, Kobe, Japan, May 12-17 2009.
- [5] N. Basilico, N. Gatti, and F. Amigoni. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *Proc. AAMAS*, Budapest, Hungary, May 10-15 2009.
- [6] N. Basilico, N. Gatti, T. Rossi, S. Ceppi, and F. Amigoni. Extending algorithms for mobile robot patrolling in the presence of adversaries to more realistic settings. In *Proc. IAT*, Mialn, Italy, September 11-15 2009.
- [7] M. Flood. The hide and seek game of Von Neumann. *Management Science*, 18(5):107–109, 1972.
- [8] R. Fourer, D. Gay, and B. Kernighan. A modeling language for mathematical programming. *Management Science*, 36(5):519–554, 1990.
- [9] D. Fudenberg and J. Tirole. *Game Theory*. The MIT Press, 1991.
- [10] N. Gatti. Game theoretical insights in strategic patrolling: Model and algorithm in normal-form. In *Proc. ECAI*, pages 403–407, 2008.
- [11] E. Halvorson, V. Conitzer, and R. Parr. Multi-step multi-sensor hide-seeker games. In *Proc. IJCAI*, Pasadena, USA, 2009.
- [12] R. Leigh, S. Louis, and C. Miles. Using a genetic algorithm to explore A\*-like pathfinding algorithms. In *Proc. CIG*, pages 72–79, 2007.
- [13] 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*, pages 895–902, 2008.
- [14] P. Paruchuri, J. Pearce, M. Tambe, F. Ordonez, and S. Kraus. An efficient heuristic approach for security against multiple adversaries. In *Proc. AAMAS*, pages 311–318, 2007.
- [15] Stanford Business Software Inc. <http://www.sbsi-sol-optimize.com/>.
- [16] A. Stentz. Optimal and efficient path planning for partially known environments. In *Proc. ICRA*, pages 3310–3317, 1994.
- [17] B. von Stengel and S. Zamir. Leadership with commitment to mixed strategies. CDAM Research Report LSE-CDAM-2004-01, London School of Economics, 2004.