Abstract—Patrolling environments by means of autonomous mobile robots has received an increasing attention in the last few years. The interest of the agent community is mainly in the development of effective patrolling strategies. Approaches based on game theory have been demonstrated to be very effective. They model the patrolling situation as a two-player leader-follower game, where the patroller is the leader and the intruder is the follower. These models present several limitations that prevent their use in realistic settings. In this paper, we extend the most general model from the state of the art along two directions, we propose algorithms to solve efficiently our extensions, and we experimentally evaluate them.

Index Terms—Algorithmic game theory, games for security, robotic patrolling with adversaries

I. INTRODUCTION

The problem of patrolling environments by means of autonomous mobile robots has become a topic of increasing interest in the last few years [1], [2], [3], [4]. The scientific challenge for the agent community is the development of effective patrolling strategies. The basic setting considers a patroller aiming at preserving an environment from intrusions. It has some ability to detect the intruder that, in turn, can hide and observe the robot patrolling the environment before attempting to intrude. The problem is particularly interesting when the patroller cannot employ a deterministic strategy for its movements, e.g., a fixed route, see [5]. In these cases unpredictable strategies should be employed.

The literature shows a large number of studies that apply game theory [6] to patrolling scenarios to produce effective unpredictable strategies [2], [3], [4]. These strategies usually grant the patrolling robot a larger expected utility than adopting a purely random strategy that does not consider the presence of the intruder [2]. A patrolling situation is commonly modeled as a two-player (i.e., the patroller and the intruder) strategic-form (i.e., simultaneous) game. The fact that the intruder can observe the patroller before acting leads to the adoption of a leader-follower solution concept [7], where the patroller is the leader and the intruder is the follower. The environment is commonly modeled as a set of connected cells that can be traversed by the robot and that may have different values for the patroller and the intruder. The problem of searching for a leader-follower patrolling strategy is cast to an optimization problem and commercial solvers [8], [9] are employed for its resolution.

The models proposed in the literature provide a coarse 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 [3] and we provide two different extensions with the aim to make the patrolling model closer to real-world situations. The first extension captures the movement of the intruder (Section III). In previous models [1], [3], [4], [10], the intruder is assumed to strike a target by directly appearing at it. In our extension, we force the intruder to reach the target moving along paths. We formulate a mathematical programming problem to capture our extension and we reduce the search space by 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, to the best of our knowledge, the first attempt to extend the well-known game theoretic iterated dominance [6] to the case of patrolling. The second extension captures the visibility limitations of the intruder (Section IV). In the previous models, the intruder is assumed to be in the position to observe perfectly the patroller’s movements. In our extension, we relax this assumption, allowing the intruder, before attempting an intrusion, to observe perfectly the patroller only in a portion of the environment. This introduces imperfect information in the game and, to the best of our knowledge, it is the first attempt to introduce imperfect information in leader-follower games. Also in this case, we formulate 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 extensions assure the patroller a larger expected utility than the original model and our reduction algorithms are very effective, saving more than 90% computational time, and, although our models extend the state-of-the-art model, the time needed for solving them is not larger than the time needed for solving the original model.

II. STATE OF THE ART

A. Mobile Robot Strategic Patrolling

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 use of mobile robots for patrolling applications is receiving a growing attention both in the multiagent [2], [3], [4], [11] and robotics [1], [12], [13] literature. This problem 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 (i.e., a fixed route) for their movements. As a result, they should adopt an unpredictable patrolling strategy, randomizing over the targets trying to reduce the intrusion probability of a possible intruder [1], [3]. The scientific challenge is the development of effective patrolling strategies.

Recently, game theoretic models [6] have been proposed to develop patrolling strategies that account for the behavior of the intruder. As shown in [2], these models allow one to derive strategies that are efficient in terms of expected utility for the patroller. Three are the main models discussed in the literature. In [1], the authors propose a model wherein the environment to be patrolled is a perimeter, the patrollers have no preferences over the areas constituting the perimeter, and the patrolling strategy does not depend on the preferences of the intruder. Essentially, the patrolling strategy is the patroller’s max-min strategy. In [4], [14], the authors propose a model that does not consider the specific environment topology with which the targets are connected. The preferences of the patroller and of the intruder are explicitly taken into account and the situation is modeled as leader-follower game [7], where the leader is the patroller and the follower is the intruder. According to this approach, the intruder can observe the patroller, derive a belief over its strategy, and then strike a target. In [3], the authors propose a model (from here on 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 [3]. The environment is composed of a set \( C = \{c_1, \ldots, c_n\} \) of cells to be patrolled, whose topology is modeled by a directed graph \( G \), which is represented 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 \( c_i \in T \) requires the intruder \( d_i > 0 \) turns to intrude (\( d_i \) is called the penetration time for cell \( c_i \)).

In one turn the patroller can move between two adjacent cells in \( G \) and patrol 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 [3], [15]).

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 [3], such situation can be modeled as a strategic-form game [6] where the patroller chooses its patrolling strategy, namely, the set of probabilities \( \alpha_{i,j} \) 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 attack and when, namely, its actions include \( \text{enter-when}(c_i, c_j) \) 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 of the game is \( \text{intruder-capture} \), otherwise the outcome is \( \text{penetration-c}_{i} \); when the intruder performs a \( \text{stay-out} \) action and does not enter any cell, then the outcome is \( \text{no-attack} \).

Agents’ payoffs are defined as follows. We denote by \( X_i \) and \( Y_i \) the payoffs to the patroller and the intruder, respectively, when the outcome is \( \text{penetration-c}_{i} \). We denote by \( X_0 \) and \( Y_0 \) the payoffs to the patroller and to the intruder, respectively, when the outcome is \( \text{intruder-capture} \). For the sake of simplicity, we assume that, when the outcome is \( \text{no-attack} \), 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:

\[
X_i < X_0 \quad \text{and} \quad Y_0 \leq 0 < Y_i \quad \text{for all} \quad c_i \in T, \quad \text{and} \quad X_i = X_0 \quad \text{and} \quad Y_i = 0 \quad \text{for all} \quad c_i \in C/T.
\]

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 [7]. 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 (see [3] for full details). More precisely, at first we search for a patrolling strategy such that \( \text{stay-out} \) 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 computed in the following way. For each possible intruder’s best response \( \text{enter-when}(\cdot, \cdot) \), 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

Although the BGA model is a good patrolling model, its applicability to real cases can be improved. In this paper we present two significant extensions that make the BGA model closer to realistic settings. (Some easy extensions capturing different sensorial capabilities and uncertainty over sensors have been already discussed in [3].) We extend the BGA model to capture situations that are not currently addressed in the game theoretic patrolling literature [1], [3], [4], whereas they are studied in depth in the field of pursuit-evasion [12], [13]. The main problems addressed in this field are very close
to our patrolling problem and mainly focus on characterizing environments and strategies without providing algorithms for computing agents’ optimal strategies. For the sake of clarity, we present the two extensions separately; their integration is easy.

Our first extension refines the intruder’s movement, capturing the situation wherein the intruder moves along paths connecting an access area to the target. In the models studied in the literature so far, the intruder is assumed to appear directly at the target. Consider Fig. 1: numbered cells are the areas to be patrolled, black cells denote obstacles and black rectangles denote access areas. In cells representing target areas, penetration times \( d_i \) together with payoff values \((X_i, Y_i)\) are reported (we also assumed \( X_0 = 1 \) and \( Y_0 = -1 \)). The gray lines denote possible paths that the intruder can cover to reach the target 05 when entering from cell 03 or from cell 11. Our second extension captures the situation wherein the intruder cannot observe the whole environment and therefore cannot perfectly know the position of the patroller when attacking. This situation is common in realistic settings. Consider Fig. 2: when entering the environment from cells 03 or 11, the intruder can observe the position of the patroller only if it is in the gray cells. If the patroller is in the white cells, the intruder cannot know with certainty the cell wherein the patroller is. In order to simplify notation, in the example we chose an arbitrary set of hidden (white) cells. Modeling situations where hidden cells depend on the considered access area, patroller’s sensor range and environment topology is a straightforward generalization.

### III. Capturing Intruder’s Movement

#### A. Mathematical Model

We enrich the BGA model by introducing access areas, constraining the intruder to move along paths connecting access areas to targets, and allowing the patroller to capture the intruder also along these paths. For the sake of simplicity, we assume that the intruder can disappear from a target after a number of turns equal to the penetration time of that target. (Modeling situations in which the intruder, once completed an intrusion, escapes from the environment following a path is an easy extension of the results presented in this section.) As is customary in pursuit-evasion field, we assume that the intruder can move infinitely fast. Therefore, covering a path takes only one turn, regardless of its length. If the intruder attacks a target \( t_i \) at time \( k \), then at time \( k \) it can be detected on every cell of the path it follows (including both target and access area), while in the remaining time interval \([k-1,k+d_i-1]\) it can be detected only in \( t_i \). We enrich the BGA model by introducing \( A \subseteq C \), i.e., the set of cells containing at least an access area and we call \( p^k_{\alpha_i,t_j} \) a sequence of cells representing the \( k \)-th path connecting cell \( c_i \) to \( c_j \). We need also to redefine intruder’s set of actions. Called \( P \) the set of paths connecting all the possible access areas with all the possible targets, the actions \( enter-when(\cdot, \cdot) \) are defined as \( enter-when(p^k_{\alpha_i,t_j}, c_o) \) for all \( \alpha_i \in A \), \( t_j \in T \), and \( c_o \in C \), meaning that, at the time point after the patroller is in cell \( c_o \), the intruder will attack target \( t_j \) from access area \( \alpha_i \) covering path \( p^k_{\alpha_i,t_j} \). We use the operator \( last(p) \) to denote the last cell of path \( p \), e.g., \( last(p^k_{\alpha_i,t_j}) = t_j \). We can now extend the mathematical programming formulation. At first, we search for a patroller’s strategy such that \( stay-out \) is the intruder’s best response. Such strategy (if it exists) is the optimal patrolling strategy and it can be found by solving the following bilinear feasibility problem: finding a strategy (if it exists) such that \( \alpha_{i,j} \) is the intruder’s best response. Such strategy (if it exists) is the optimal patrolling strategy and it can be found by solving the following bilinear feasibility problem: finding a strategy (if it exists) such that \( \alpha_{i,j} \) is the intruder’s best response. Such strategy (if it exists) is the optimal patrolling strategy and it can be found by solving the following bilinear feasibility problem:

\[
\sum_{j \in C} \alpha_{i,j} = 1 \quad \forall i \in C \tag{1}
\]

\[
\alpha_{i,j} \leq G(c_i,c_j) \quad \forall i,j \in C \tag{2}
\]

\[
\gamma_{i,j}^{h,p} = \alpha_{i,j} \quad \forall p \in P, i \in C \setminus p \tag{3}
\]

\[
\gamma_{i,j}^h = 0 \quad \forall p \in P, i \in C \setminus p \tag{4}
\]

\[
\gamma_{i,j}^h = \sum_{x \in C \setminus last(p)} \left( \gamma_{x,i}^{h-1} \cdot \alpha_{x,j} \right) \quad \forall h \in \{2, \ldots, d_{last(p)}\}, p \in P, i,j \in C \setminus p \tag{5}
\]

\[
Y_6 + (Y_{last(p)} - Y_0) \cdot \sum_{i \in C \setminus last(p)} \gamma_{w,i}^{d_{last(p)}-p} \leq 0 \quad \forall w \in C \setminus p \tag{6}
\]

```plaintext
<table>
<thead>
<tr>
<th></th>
<th>01</th>
<th>03</th>
<th>04</th>
<th>05</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(d_1 = 7)</td>
<td>(d_3 = 6)</td>
<td>(d_4 = 6)</td>
<td>(d_5 = 6)</td>
</tr>
<tr>
<td></td>
<td>((0.7,0.3))</td>
<td>((0.6,0.45))</td>
<td>((0.6,0.45))</td>
<td>((0.6,0.45))</td>
</tr>
</tbody>
</table>
```

---

**Fig. 1.** A patrolling scenario: paths from access areas 03 and 11 to target 05 are depicted as gray lines.

**Fig. 2.** A patrolling scenario: the intruder can observe only the gray cells.
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)-(5) are used to compute the intrusion probability: they state that the patroller, in the first turn, has to visit a cell not belonging to the path followed by the intruder; constraints (6) 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); constraints (7) express that no action $enter-when(p, z)$ gives the intruder an expected utility larger than that of $stay-out$. The non-linearity is due to constraints (6).

If the above problem admits a solution, the resulting $\alpha_{i,j}$s constitute the optimal patrolling strategy. When the above problem is unfeasible, we pass to the second stage of the algorithm. Let us call $q$ a path from an access area $a_i$ to the cell $last(q) \in T$ and $z$ a generic cell. For each possible action $enter-when(q, z)$, we compute the patroller’s best strategy subject to the assumption that $enter-when(q, z)$ is a best response for the intruder. Each one of these strategies can be computed with a bilinear optimization problem, where the objective function is the patroller’s expected utility:

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

s.t. 

$$Y_0 + \left( Y_{last(q)} - Y_0 \right) \sum_{i \in C \setminus last(q)} d_{i,a}(q) \gamma_{z,i} \geq 0$$

$$Y_0 + \left( Y_{last(p)} - Y_0 \right) \sum_{i \in C \setminus last(p)} d_{i,a}(p) \gamma_{w,i} \geq 0 \quad \forall w \in C, p \in P \quad (8)$$

Constraints (8) express that no action $enter-when(p, w)$ gives the intruder an expected utility larger than that of $enter-when(q, z)$. The patroller’s optimal strategy is the strategy that gives the largest expected utility among all the ones computed above.

### B. Removal of Dominated Actions

Solving the previous leader-follower game is computationally expensive because each single bilinear programming problem is not easy and the number of such problems to be solved grows with the number of paths and cells in the environment. However, the search space can be dramatically reduced, saving computational time. The idea behind our reduction is very close to the game theoretic iterated dominance [6], [16]. That is, we remove all the intruder’s actions $enter-when(p, c)$ that the intruder will never play independently of the patroller’s strategy. Technically speaking, $enter-when(p, c)$ is dominated by $enter-when(p', c')$ if the former gives the intruder an expected utility never larger than the latter independently of the patroller’s strategy. We address this reduction in two steps. In the first one, we determine the minimal set of paths $\{p\}$ that the intruder would follow. In the second one, given each $p$, we determine the minimal set of cells $\{c\}$ that the intruder would consider in its actions $enter-when(p, c)$.

Let us discuss the first step. Although the number of paths is in principle infinite, the set of paths along which a rational intruder would move is finite. The literature provides a large number of works dealing with path finding (e.g., see [17]), usually based on $A^*$ [18]. 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, since the patroller’s randomization probabilities are unknown (they constitute the solution). Nevertheless, a number of paths can be discarded, never being followed 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 does not exist any $p''_{a_i,t_j}$ (with $u$ not necessarily different from $i$) such that all the cells of $p''_{a_i,t_j}$ are strictly contained in $p'_{a_i,t_j}$.

Consider Fig. 1, $p'_{03,05} = \langle \{03,04,05\}, \gamma_0 \rangle$ is strictly contained in $p''_{03,05} = \langle \{03,02,03,04,05\}, \gamma_0 \rangle$; it can be easily seen that $p'_{03,05} = \langle \{03,04,05\}, \gamma_0 \rangle$ is irreducible while $p''_{03,05}$ is not. From the above statements it follows that action $enter-when(p''_{a_i,t_j}, c)$ is dominated by $enter-when(p'_{a_i,t_j}, c)$ when $p_{a_i,t_j}$ is contained in $p''_{a_i,t_j}$. The proof is easy: in general 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 $p''_{a_i,t_j}$, the intruder’s probability to be captured when it follows $p'_{a_i,t_j}$ is not larger than the one when it follows $p''_{a_i,t_j}$. Thus we can safely consider only irreducible paths. Fig. 1 depicts all the irreducible paths with target 05.

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 $v$, $\eta(v) \in A$ iff $v$ is a terminal node; $v''$ is a successor of $v'$ iff $v'' \in \{v \mid G(\eta(v'), \eta(v)) = 1\}$ and $v''$ never appears in the branch from $v_0$ to $v'$;

2) for each pair of paths $(p', p'')$ from the root to leaves if $p'$ is strictly contained in $p''$ then remove $p''$.

Fig. 3 reports an example of the tree of paths with target 05.

Let us now discuss the second step. For each irreducible path $p$, we compute the cells $\{c\}$ 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 cases:

- If cell $c$ belongs to $p$ and from it the patroller can only move to cells belonging to $p$ (i.e., $c$ is a cell of the path not on its frontier), then $enter-when(p, c)$ is dominated by $stay-out$, e.g., consider Fig. 1, if $p = \langle 11,12,13,08,05 \rangle$ and $c = 13$, the intruder will be surely captured by making $enter-when(p, c)$.

- If there exists a cell $c'$ that is non-adjacent to any cell of path $p$ (recall that, in the first intrusion’s turn, the intruder can be captured anywhere on the path) and such
that the patroller, when it starts from \( c' \) and reaches \( \text{last}(p) \) before \( d_{\text{last}(p)} \) turns, must always visit cell \( c \), then \( \text{enter-when}(p, c) \) is dominated by \( \text{enter-when}(p, c') \). Consider Fig. 1, if \( p = (11, 12, 13, 08, 05) \) and \( d_{05} = 6 \), \( \text{enter-when}(p, 04) \) is dominated by \( \text{enter-when}(p, 06) \), indeed, if the patroller starts from 06, it must always pass through cell 04 to reach 05 in less than \( d_{05} = 6 \) turns. The rationale is that the probability with which the patroller visits 05 from 06 in less than \( d_{05} = 6 \) turns is non-strictly smaller than the probability the patroller visits 05 from 04 in less than \( d_{05} \) turns.

The set \( \{c\} \) of cells such that action \( \text{enter-when}(p, c) \) is not dominated can be found resorting again to tree search techniques. We build a tree of paths where, called \( v_0 \) the root, \( \eta(v_0) = \text{last}(p) \) and a node \( v'' \) is a successor of \( v' \) iff \( v'' \in \{v\mid G(\eta(v'), \eta(v)) = 1\} \) and \( v'' \neq v' \). The maximum level of the tree is \( d_{\text{last}(p)} \). That is, we consider all the paths whose length is \( d_{\text{last}(p)} \) and its last cell is \( \text{last}(p) \). Fig. 4 reports the tree of paths generated as prescribed above with \( p_{11,05} = (11, 12, 13, 08, 05) \). 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. 4, given \( p = (11, 12, 13, 08, 05) \), cells 08, 12, and 13 are discarded;

2) if a cell \( \eta(v) \) is represented only once in the tree and for every cell \( c \) such that \( G(\eta(v), c) = 1 \) it holds that \( c \notin p \), then for every node \( w \) that is ancestor of \( v \) the cell \( \eta(w) \) is discarded, e.g., in Fig. 4, 06 appears only once and no adjacent cell to it is on the path and thus 01, 02, 03, 04, 05 are dominated;

3) 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 \( G(\eta(w), c) = 1 \) it holds that \( c \notin p \), then \( \eta(v) \) is dominated. For example, in Fig. 4, 11 is always ancestor of 10 and 10 always follows 11, but there is a cell (11) adjacent to 10 that is on the path, therefore cell 11 is not dominated by cell 10.

In Fig. 4, black nodes denote cells \( c \) such that actions \( \text{enter-when}(p_{11,05}, c) \) are dominated.

### C. Experimental Evaluation

We experimentally evaluate the computational time of our algorithms. We implemented the two algorithms described in this section for reducing the search space in C. We solved the feasibility and the optimization problems by using SNOPT 7.2 solver [9] through AMPL [8] on a Pentium 3GHz 1 GB RAM Linux computer.

When comparing the path-extended model with the basic BGA, an average increase of the patroller’s expected utility was observed. This is reasonable since the introduction of path-following constraints makes the intruder weaker than the one modeled in [3]. We evaluated the computational times for solving four different configurations: the patrolling problem in absence of paths (basic BGA model [3]) and the problem with paths (as formulated in Section III-A); we evaluated this last configuration without any reduction (non-reduced), with the reduction only on the paths (semi-reduced), with the reduction on the paths and the cells (fully-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. When an execution exceeds 10 hours we stopped it (only the non-reduced configuration does not terminate in 10 hours). In Tab. I, we report the experimental results with the setting depicted in Fig. 1. (The computational time spent for reducing the game is negligible, i.e., less than 1 s, and then it is omitted.) The first consideration is that our reduction algorithms significantly reduce the number of
bilinear problems and, accordingly, the computational time. The second consideration is that, although our model refines the BGA model, the computational effort does not increase.

According to the above evaluation scheme, we evaluated 10 different randomly-generated patrolling settings with size (in terms of cells and paths) comparable to the setting of Fig. 1. The results, omitted for reasons of space, are perfectly observable by the intruder from cells that are inaccessible to the patroller. Consider Fig. 2 and suppose that the intruder is going to attack at time point after the game is initiated, confirming the feasibility problem can be stated as follows:

\[ \alpha_{i,j}^{*} = \sum_{c_k \in H} \alpha_{i,k} \quad \forall c_i, c_j \in H \]  

The values of \( \alpha_{i,j}^{*} \) when \( c_j \in C \setminus H \) will be zero while when \( i \in C \setminus H \) and \( j \in H \) the equality \( \alpha_{i,j}^{*} = \alpha_{i,j} \) will hold. The idea behind the computation of \( \alpha_{i,j}^{*} \) is that, since the patroller has not become visible, then the probability of the event in which the patroller moved from \( c_i \in H \) to \( c_j \notin H \) is equal to zero. Technically speaking, we need to normalize by Bayes rule the probabilities \( \alpha_{i,j} \). That is:

\[ \alpha_{i,j}^{*} = \sum_{c_k \in H} \alpha_{i,k} \alpha_{i,j} \]

We call \( \Omega_{t} \) the observation horizon set for a state \( s = (c_i, o) \), i.e., \( o \in O_{t} \), and on the basis of \( \alpha_{i,j}^{*} \) we compute \( \beta_{i,j}^{v} \)’s as follows:

\[ \beta_{i,j}^{v} = \sum_{c_k \in C \setminus \Omega_{t}} \beta_{i,k}^{v-1} \alpha_{i,j}^{*} \quad \forall c_i, c_j \in H, c_j \in H, k \in O_{t} \setminus \{0\} \]  

The values of \( \beta_{i,j}^{v+1} \) for \( c_i, c_j \notin H \) will be zero. Finally, the feasibility problem can be stated as follows:

\[ \sum_{c_j \in C} \alpha_{i,j} \geq 1 \quad \forall c_i \in C \]  

\[ \sum_{c_j \in C} \alpha_{i,j} \leq G(c_i, c_j) \quad \forall c_i \in C \]  

\[ \gamma_{h,t} = \beta_{i,j}^{(v+1)} \quad \forall s \in S, c_i \in T, c_j \in C \]  

\[ Y_0 + \left( Y_1 - Y_0 \right)^t \cdot \sum_{c_t \in C \setminus C_t} \gamma_{r,t}^t \leq 0 \quad \forall c_t \in T, r \in S \]
Constraints (12)-(14) are equal to those of the feasibility problem of Section III-A; constraints (15) relate the intruder’s estimation over the patroller’s position to the intrusion probabilities $\gamma$: constraints (16) express the Markov hypothesis over the patroller’s decision policy; constraints (17) express that no action $enter-when(t,s)$ gives the intruder a larger expected utility than $stay-out$. 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 $enter-when(w,s)$:

$$\max \quad X_0 + \left( X_w - X_0 \right) \sum_{c_i \in C \setminus C_w} \gamma^{d_{w,0}}$$

s.t.

$$Y_0 + \left( Y_w - Y_0 \right) \sum_{c_i \in C \setminus C_w} \gamma^{d_{w,0}} \geq \gamma^{d_{w,0}} \geq Y_0 + \left( Y_w - Y_0 \right) \sum_{c_i \in C \setminus C_w} \gamma^{d_{w,0}} \quad \forall c_1 \in T, r \in S (18)$$

The meanings of the objective function and of constraints (18) are analogous to those of Section III-A.

### B. Removal of Dominated Actions

Analogously to what we did in Section III-B, we aim at removing actions that the intruder would never play, independently of the patroller’s strategy. On the one hand, the problem is simpler, since we do not consider paths. This allows us to focus only on the determination of the minimal set of states $\{s\}$ to be considered for the actions $enter-when(t,s)$. On the other hand, the problem is harder, since the states are, in principle, infinite, being possibly infinite the length of the observation. The main result of this section is that we can safely consider a finite set of states preserving the optimality of the solution. We provide an algorithm to find the intruder’s non-dominated actions.

The algorithm develops into two steps. In the first step, we suppose the whole environment to be perfectly observable for the intruder and we determine the set of cells that will appear in the states of non-dominated actions. In the second step, we determine the maximum value for $\gamma$ for to be considered for each cell $c$ in states $s = \langle c, o \rangle$. We show that actions whose states present a larger value for $\gamma$ are dominated.

We focus on the first step. We suppose the whole environment to be perfectly observable for the intruder. Given a target $t$, the aim is to find the minimal set of cells $\{c\}$ such that $enter-when(t,c)$ is not dominated. The solving algorithm is similar to the algorithm presented in Section III-B for the computation of the minimal set of cells. In this case, paths are not present. The revised algorithm builds the tree of paths exactly as the original one, the difference is in the determination of the dominance. In this case:

1) if a cell $\eta(v)$ is represented only once in the tree then for every node $w$ that is an ancestor of $v$ the cell $\eta(w)$ is discarded;

2) if for a pair of cells $c$ and $c'$, every $v$, such that $\eta(v) = c$, is an ancestor of a $w$ such that $\eta(w) = c'$, then $c$ is discarded.

We denote by $ND_{nh}$ the set of non-dominated non-hidden cells and by $ND_h$ the set of non-dominated hidden cells. Consider Fig. 2, given $t = 05$, $ND_{nh} = \{07, 10, 12\}$ and $ND_h = \{06, 08, 09, 13\}$. The cells that will appear in the states of non-dominated actions are all the cells $c \in ND_{nh}$ and all the cells $c \in C \setminus H$ from which the patroller can reach a cell in $ND_h$ without traversing any non-hidden cell except $c$ itself. Consider Fig. 2, cells 02 and 04 will appear in states of non-dominated actions. Indeed, from 02, the patroller can reach $06 \in ND_h$ without traversing any non-hidden cell.

In the second step we provide the observation horizon length for any state. For each cell $c \in ND_{nh}$, action $enter-when(t,\langle c, 0 \rangle)$ is not dominated, i.e., the longest horizon to consider is $o = 0$. Consider Fig. 2, given $t = 05$, non-dominated actions are $enter-when(05, \langle 07, 0 \rangle)$, $enter-when(05, \langle 10, 0 \rangle)$, and $enter-when(05, \langle 12, 0 \rangle)$. For each cell $c$ belonging to $ND_h$, the determination of the observation horizon length is more complex. The basic idea is that the non-dominated actions are those such that in their states the probability that the patroller is in $c$ is maximum. We cannot determine the state in which such probability is maximum, anyway we can remove states where such probability is not larger than that in other states. Given a cell $c \in ND_h$ and a non-hidden cell $c'$ from which the patroller can reach $c$ without passing from other non-hidden cells, the observation horizon $o$ to be considered is the shortest distance between $c$ and $c'$. This is because, if the patroller disappeared from $c'$, then the maximum probability that it is in $c$ is after a number of turns equal to the distance between the two cells. The proof is based on Markov chain properties. Consider Fig. 2, if $c = 06$ and $c' = 02$, then state $\langle 02, 2 \rangle$ will appear in a non-dominated action. Similarly, other states appearing in non-dominated actions are: $\langle 10, 2 \rangle$ (i.e., a state in which the probability that the patroller is in 06 can be maximum). The longest observation horizon for the game depicted in Fig. 2 is 2. This value will be used in constraints (11). In general, each cell $c \in ND_h$ induces the presence of some states; the exception is when $c$ is adjacent to a non-hidden cell. In this case, it induces only the state $\langle c', 0 \rangle$ such that $c'$ is the adjacent non-hidden cell and $o = 1$. Consider Fig. 2, $\langle 10, 1 \rangle$ is the state in the which the probability that the patroller is in 09 is maximum, for this reason state $\langle 02, 3 \rangle$ is discarded.

### C. Experimental Evaluation

The reduction algorithms and the mathematical programming model has been implemented as described in Section III-C. In this case we report the results for the original configuration, the semi-reduced configuration wherein all the states with $o \leq 2$ are considered, and the fully-reduced configuration. We do not consider any non-reduced configuration, since it is not well defined, being in principle infinite, the longest observation horizon. Similarly to the path-extended model, an average increase of the patroller’s expected utility is observed also in this case. As for the first extension, in
Tab. II we report the computational times spent to solve the setting depicted in Fig. 2. Also in this case the reduction algorithm works very well, allowing one to save significantly computational time. Moreover, the computational time is not larger than that required to solve the original problem where the whole environment is perfectly observable.

We experimentally evaluated 10 different randomly-generated patrolling settings with size comparable to the setting depicted in Fig. 2. The percentage of computational time saved is in the range 47% – 55%. This confirms the effectiveness of our algorithm.

V. CONCLUSIONS AND FUTURE WORKS
The problem of producing effective strategies for mobile robot patrolling is receiving a lot of attention. The literature offers interesting models based on game theory able to take into account possible adversaries. However, these models provide a coarse description of the situations. In this paper we proposed two different extensions in the attempt to make patrolling models closer to real-world applications. We enriched a state-of-the-art model with the intruder’s movements, introducing access points from which the intruder can enter the environment and by constraining it to move along paths, and with the intruder’s observation capabilities, capturing the situations in which the intruder cannot perfectly observe the whole environment when it decides the target to attack. For both these extensions we provided mathematical programming formulations, reduction algorithms, and experimental evaluations. The experimental results show that our algorithms are very efficient in terms of computational time. In particular, our reduction algorithms allow one to save up to 97% of computational time.

We are interested in extending our work along several directions. At first, we shall develop a simulator for patrolling based on an off-the-shelf robot simulator to evaluate our model in quasi-real settings. At second, we shall extend our model to capture patroller’s augmented sensing capabilities (also uncertain) and to introduce a delay between the turn in which the intruder decides to enter and the turn in which it actually enters. Finally, we shall extend our model to capture settings where multiple mobile robots cooperate to patrol the environment.

REFERENCES

<table>
<thead>
<tr>
<th>BGA</th>
<th>BGA with partial observation</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>semi-reduced</td>
<td>fully-reduced</td>
<td></td>
</tr>
</tbody>
</table>