

Leader-Follower Strategies for Robotic Patrolling in Environments with Arbitrary Topologies

Nicola Basilico
DEI, Politecnico di Milano,
Milano, Italy
basilico@elet.polimi.it

Nicola Gatti
DEI, Politecnico di Milano,
Milano, Italy
ngatti@elet.polimi.it

Francesco Amigoni
DEI, Politecnico di Milano,
Milano, Italy
amigoni@elet.polimi.it

ABSTRACT

Game theoretic approaches to patrolling have become a topic of increasing interest in the very last years. They mainly refer to a patrolling mobile robot that preserves an environment from intrusions. These approaches allow for the development of patrolling strategies that consider the possible actions of the intruder in deciding where the robot should move. Usually, it is supposed that the intruder can hide and observe the actions of the patroller before intervening. This leads to the adoption of a leader-follower solution concept. In this paper, mostly theoretical in its nature, we propose an approach to determine optimal leader-follower strategies for a mobile robot patrolling an environment. Differently from previous works in literature, our approach can be applied to environments with arbitrary topologies.

Categories and Subject Descriptors

I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence—*Intelligent agents*

General Terms

Algorithms

Keywords

Leader-follower strategies, robotic patrolling

1. INTRODUCTION

Game theoretic approaches to patrolling have become a topic of increasing interest in the last few years [1, 2, 12, 13]. The basic setting considers a patrolling mobile robot with the goal to preserve an environment from intrusions. The robot has some ability to detect the intruder and the intruder can hide and observe the robot patrolling the environment before attempting to intrude. Use of game theory enables the development of patrolling strategies that consider the possible actions of the intruder in deciding where the robot should move. This usually grants the patrolling robot a larger expected utility than adopting a purely random strategy [2]. The fact that the intruder can observe

the patroller before acting leads to the adoption of a leader-follower solution concept [19] for the game that models the interaction between the patroller and the intruder.

In this paper, we propose an approach to determine optimal leader-follower strategies for a mobile robot patrolling an environment. Extending recent works in literature [1, 12, 13], our approach can be applied to environments with arbitrary topologies. In particular, we represent the environment 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 main original contributions of this paper are the following ones.

- We model patrolling in these environments as a two-player (i.e., the patroller and the intruder) extensive-form game with imperfect information and infinite horizon. The patroller's actions are movements between connected cells, while the intruder can wait hidden and observe the patroller or can attempt to intrude in a cell. Payoffs of the game are calculated according to the values of the cells for the two agents. We show that the leader-follower solution of this game is the *optimal patrolling strategy* for the mobile robot, giving it the maximum expected utility.
- Since to the best of our knowledge literature does not provide any method for finding a leader-follower strategy in extensive-form infinite-horizon games, we propose an approach that introduces symmetries in the patroller's strategies. Symmetries come down to linking the strategy of the patroller to the history H of its last $|H|$ actions.
- We formulate a bilinear mathematical programming problem to find the optimal patroller's strategy when $|H| = 1$, namely when the patroller operates under a Markov hypothesis.
- We provide some ways to reduce the complexity of finding a solution by simplifying the mathematical programming problem to be solved.
- Finally, we discuss the relation between the optimal value for $|H|$ and the topology of the environment.

This paper is structured as follows. The next section reviews the relevant related work. Section 3 presents our game model and our approach to its solution. Sections 4 and 5 describe some ways to reduce the complexity of solving our model and some of its extensions, respectively.

Cite as: Leader-Follower Strategies for Robotic Patrolling in Environments with Arbitrary Topologies, Nicola Basilico, Nicola Gatti, Francesco Amigoni, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 57–64
Copyright © 2009, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org), All rights reserved.

2. GAMES FOR SECURITY WITH MOBILE ROBOTS

A patrolling situation is characterized by one or more *patrollers* and by some *targets* to be patrolled. The interest in studying patrolling situations outside purely geometrical approaches (like those for the art gallery problem) emerges when, due to the characteristics of the setting (e.g., sensor range of the patrollers and time needed by an intruder to penetrate an area), the patrollers cannot employ a deterministic strategy, otherwise the intruder will surely succeed in attacking a target. As a result, patrollers should adopt an unpredictable patrolling strategy, randomizing over the targets and trying to reduce the intrusion probability. Some patrolling strategies of this type have been developed in mobile robotics [11, 14], but they usually do not consider any explicit model of the adversary, i.e., the intruder. Only very recently, adversaries have been taken into account in the development of patrolling strategies for mobile robots. As shown in [2], strategies that consider models of adversaries give the patrolling robot a larger expected utility than strategies that do not. Two are the main methods proposed in literature for robotic patrolling with adversaries: one does not explicitly model the preferences of the adversaries [1], whereas the other one does [12, 13]. Before briefly reviewing these methods, we note that similar strategic problems have been addressed in the pursuit-evasion field (e.g., [8, 18]). 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 usually knows only the current position of the pursuer but not its strategy, make the pursuit-evasion strategies not directly applicable to our patrolling scenario.

The work proposed in [13] explicitly considers the preferences of the adversary, as we do in this paper. The authors deal 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. They 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) [6]. 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 [19]. (A slight variation of this approach has been applied to the problem of patrolling n access points with $m < n$ static checkpoints at the Los Angeles International Airport [15].) As discussed in [7], the approach in [13] presents two drawbacks. First, since it does not consider the time spent by the robot to move between two areas, the model is applicable only in environments with fully connected topologies. Second, to avoid game theoretical inconsistencies, the decisions of the patroller must be over the next area, and not over the next route, to patrol. A comparison between our model and that in [13] is presented in Section 5.

A different method, where the preferences of the adver-

sary are not taken into account, is employed 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 max-minimizes 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 produced strategy is optimal when also the intruder has no preferences over the cells.

The attempt we make in this paper is to combine and improve the strategic approach proposed in [13] and refined in [7] and the approach proposed in [1]. Specifically, we enrich the models presented in [7, 13] by modeling the movement of the patroller in environments with arbitrary topologies. We also enrich the model presented in [1] by allowing a generic environment topology, considering agents' preferences, and generalizing the patroller's sensing capabilities.

3. AN EXTENSIVE-FORM GAME FOR ROBOTIC PATROLLING

3.1 Scenario, Assumptions, and Objective

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

- time is discretized in turns (as in [1, 13]);
- there is a single patrolling robot equipped with a sensor (e.g., a camera) able to detect intruders (as in [12, 13]);
- the environment is discretized in cells and its topology is represented by a directed graph (as in [1]);
- 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 [1, 13]).

The patroller's goal is to detect the intruder. If this happens, we say that the intruder has been "captured" by the patroller. The final goal of the proposed game-theoretic approach is to find the optimal strategy the patrolling robot should follow to detect effectively the intruder.

3.2 Robotic Patrolling Game Model

In this section we introduce the proposed model. In turn, we formally describe the environment where the patroller and the intruder act, the patroller's movement and sensing capabilities, and the game mechanism.

The environment is composed of a set C of n cells to be patrolled, whose topology is given by a directed graph \mathcal{G} . We represent \mathcal{G} by a matrix $T(n \times n)$, where $t_{i,j} = 1$ means that cells i and j are adjacent (the patroller can go from i to j with one action) and $t_{i,j} = 0$ means that they are not. In the former case, we say that cells i and j are directly connected. A cell may represent an access point to an area with some value (e.g., a door as in [1]) or an area with some value (e.g., an house as in [13]). Finally, each cell i requires the intruder $d_i > 0$ turns to enter.

We assume a simple movement model for the patroller: it spends one turn to move between two cells directly connected in \mathcal{G} and patrol the arrival cell. This model could be

easily extended to capture the time spent by the patroller to reverse the movement direction, as done in [1]. The sensing capabilities of the robot are captured by introducing 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. Given that the patroller is in cell i , we say that it senses all the cells j for which $v_{i,j} = 1$. Matrix V allows one to combine the sensing capabilities of the patroller with the topology of the environment. When the patroller can sense only its current cell, V is the identity matrix. The model could be easily extended to account for the uncertainty of the sensors, by letting V represent the probability that the patroller can sense the intruder from a given cell.

The game we employ to model the above scenario is a two-player extensive-form game with imperfect information. In particular, we use a two-player dynamic repeated game [6], where the players are the patroller and the intruder. (The game can be represented also as a partially-observable stochastic game with infinite states.) At each turn, a strategic-form game is repeated in which the players act simultaneously. The patroller chooses the next cell to move to among those directly connected to its current cell; formally, called i the current cell of the robot, its actions are $move(j)$, 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, its actions are $wait$ and $enter(i)$. If, instead, the intruder has previously attempted to enter a cell i , it cannot take any action for d_i turns after decision. This repeated game is dynamic since it changes at each 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. The game is with imperfect information since, when the patroller acts, it does not know whether the intruder is currently within a cell or is waiting to attack. The game has an infinite horizon, since the intruder is allowed to wait indefinitely outside the environment. The possible outcomes of the game are: *intruder-capture*, when the intruder attempts to enter a cell i at t and the patroller senses 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 sense 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. However, other situations, including when the intruder has an incentive to enter, could be easily captured.) Consistency constraints over these values are: $X_i \leq X_0$ and $Y_0 \leq 0 \leq Y_i$ for all $i \in \{1, 2, \dots, n\}$. Furthermore, our model can capture the possibility that the intruder's payoffs are uncertain to the patroller. According to the Harsanyi transformation [6], the intruder i can be of different types $\theta_i(k)$, each one characterized by a particular set of payoffs. Each type $\theta_i(k)$ is associated to a probability $\omega_i(k)$. However, in the following, we will consider the single type case.

An example of a patrolling setting captured by our model is shown in Fig. 1. The bold numbers identify the cells; black blocks are obstacles; the values reported in cell i are (X_i, Y_i) and d_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 05 (this is equivalent to say that cell 04 contains "less value" than cell 05). Note also that cells 04, 05, and 10 have some interest for the intruder (for them, $Y_i > 0$). We call them *targets*.

01 (1,0) $d_{01} = 1$	02 (1,0) $d_{02} = 1$	03 (1,0) $d_{03} = 1$	04 (.8,.4) $d_{04} = 6$
05 (.7,.5) $d_{05} = 4$		06 (1,0) $d_{06} = 1$	
07 (1,0) $d_{07} = 1$	08 (1,0) $d_{08} = 1$	09 (1,0) $d_{09} = 1$	10 (.8,.4) $d_{10} = 5$

Figure 1: A patrolling setting

Finally, note that, although the model presented here shares some similarities with that in [2], the approaches to their solutions are completely different.

3.3 Leader-Follower Equilibrium

The intruder's ability to observe the patroller's strategy and act on the basis of such observation "naturally" induces a leader-follower situation, where the patroller is the leader and the intruder is the follower. The peculiarity of leader-follower equilibrium is that the leader commits to a strategy and the intruder acts as a best responder given such commitment.¹ In [19] the authors show that in a two-player strategic-form game the leader never gets worse by committing to a leader-follower strategy than by playing a Nash equilibrium strategy. However, to the best of our knowledge, there is not any similar result for extensive-form games such as the one we are dealing with. In this section we extend the result presented in [19] to our context.

First, we consider the patroller's strategy in absence of any commitment, later we will show that the patroller never gets worse when it commits to a leader-follower strategy. The appropriate solution concept for an extensive-form game with imperfect-information is the *sequential equilibrium* [10]. This is a refinement of the Nash equilibrium where, given the sequential structure of the game, the strategies are guaranteed to be rational (sequential rationality) and the beliefs to be consistent with the agents' optimal strategies (Kreps and Wilson's consistency). The presence of an infinite horizon complicates the study of the game. Typically, in presence of an infinite horizon, a game is studied by introducing symmetries, e.g., an agent will repeat a given strategy each K turns. (A classical example can be found [16].)

In our specific case, we introduce the history $H = \langle a_1, a_2, \dots, a_{|H|} \rangle$, defined as the sequence of the last $|H|$ actions taken by the patroller. With a slight notation overload, we will use H also to denote the sequence of last $|H|$ cells visited by the patroller; namely $H = \langle c_1, c_2, \dots, c_{|H|} \rangle$, where c_i is the cell reached by the patroller with action $a_i = move(c_i)$, starting from cell c_{i-1} . Note that $c_{|H|}$ is the current cell

¹Rigorously speaking, the follower is not just a best responder: in order to have an equilibrium, if it is indifferent between some actions, it should choose the one that maximizes the patroller's expected utility.

of the patroller. Being our game with infinite horizon, the length of H can be infinite. Introducing symmetries amounts to consider that, in the patroller's strategies, the next action is selected on the basis of the last $|H|$ actions, with $|H|$ finite and constant during all the game. For instance, when $|H| = 0$, each action in the patroller's strategy does not depend on previously taken actions. Namely, the probability to visit a cell does not depend on the (adjacent) cell where the patroller currently is. When $|H| = 1$ the patroller chooses its next action on the basis of its last action and then its strategy is Markovian. In this case, the selection of the next cell to visit depends on the current cell of the patroller. Reasonably, when increasing the value of $|H|$, the expected utility of the patroller never decreases. Furthermore, there exists an upper bound for $|H|$, say $\overline{|H|}$, such that for any $|H| \geq \overline{|H|}$ the patroller's expected utility keeps constant. We show some preliminary results on the relation between $\overline{|H|}$ and the topology of the environment in Section 5. On the other hand, when increasing the value of $|H|$, the computational complexity for finding a patrolling strategy increases too. This imposes a trade-off between expected utility and computational effort in selecting $|H|$.

Given a value for $|H|$, the game can be reduced to a strategic-form game. This is because the game repeats every $|H|$ turns. Therefore, we can consider a reduced game that is $|H|$ -turn long and constrain agents' possible strategies to be indefinitely repeated. In our case, the patroller's strategies will be of the form $\alpha_{H,move(j)}$, i.e., the probability to execute action $move(j)$ given an history H . The intruder's strategies during the game can be conveniently represented by using the following macro-actions: *enter-when*(H, j) and *stay-out*, where j is the cell to enter and H is the history of the patroller. Action *enter-when*(H, j) corresponds to make *wait* while it observes that the patroller has not followed history H and then to make *enter*(j); *stay-out* corresponds to make *wait* forever.

In the following we show that the appropriate solution concept for our game is the leader-follower equilibrium. Indeed, when the leader (in our case the patroller) commits to a leader-follower equilibrium it cannot obtain a worse expected utility than the one it would obtain from a sequential equilibrium. We state it in the following theorem.

THEOREM 3.1. *Given the finite-horizon game described above with a fixed $|H|$, the leader never gets worse when committing to a leader-follower equilibrium strategy.*

Proof. The finite-horizon game we derive by introducing symmetries can be easily translated into a strategic-form game [6]. If the leader does not commit to a strategy, it receives the utility prescribed by a sequential equilibrium of the game. This equilibrium is a specific Nash equilibrium of the strategic-form game. By von Stengel and Zamir [19], in any two-player strategic-form game, the worst leader-follower equilibrium is not worse than the best Nash equilibrium, and therefore, in our case, the worst leader-follower equilibrium is not worse than any sequential equilibrium. The thesis of the theorem follows. \square

Hence, we can say that the leader-follower equilibrium is the appropriate solution concept for our finite-horizon game and that the corresponding strategy for the patroller is the optimal patrolling strategy for the setting we consider. Given a value for $|H|$, the leader-follower strategy gives the patroller the maximum expected utility.

3.4 Mathematical Programming Formulation under Markov Hypothesis

In this section, we formulate the problem of determining the optimal patrolling strategy as a mathematical programming problem. Its solution can be obtained using optimization software tools, e.g., [17]. According to game theory, a solution for the game we have defined for a given $|H|$ is a strategy profile $\sigma^* = (\sigma_p^*, \sigma_i^*)$ where σ_p^* is the strategy of the patroller (playing as the leader) and σ_i^* is the strategy of the intruder (playing as the follower), that are in a leader-follower equilibrium. The literature provides algorithms for finding leader-follower equilibria only in strategic-form games by solving a multi-linear programming problem [4]. At the equilibrium, the follower employs pure strategies [19]. More precisely, the follower will play the best response for the strategy the leader committed to. Thus, for each pure strategy $\sigma_i = a$, with $a \in \{\textit{enter-when}(H, j), \textit{stay-out}\}$, it is possible to compute the patroller's strategy that maximizes $EU_p(BR_i = a)$, assuming that $\sigma_i = a$ is the best response for the intruder. With this method, there are as many optimization problems as the pure strategies of the intruder and each single optimization problem is linear. The patroller will induce the intruder to follow the strategy a such that $EU_p(BR_i = a)$ is maximum. In [12], the authors proposed an alternative mathematical programming formulation based on mixed integer linear programming that is more efficient with more intruder's types. Since we study single intruder's type settings, we do not employ this formulation.

In what follows we provide a mathematical programming formulation for our model when $|H| = 1$. (This is because, as we will show, formulations with $|H| = 0$ are not generally applicable to realistic environment topologies and formulations with $|H| > 1$ can be obtained easily by extending the case with $|H| = 1$.) In this case, the Markov hypothesis holds and the patroller's strategy $\{\alpha_{(i),move(j)}\}$ can be compactly represented by the set $\{\alpha_{i,j}\} \forall i, j \in C$, where each $\alpha_{i,j}$ denotes the probability for the patroller to move from cell i to cell j . The mathematical formulation we provide is a multi-bilinear problem [3]. More precisely, given a pure strategy $\sigma_i = a$, the maximization of the patroller's expected utility is linear in the objective and bilinear in the constraints. The complexity of the optimization problem is due to the non linearity introduced by considering symmetries in our game model. More precisely, non-linearity comes from the need to constrain behavioral strategies² to be equal in the same state, i.e., $\alpha_{i,j}$ is fixed for all the decision nodes for which the patroller's current cell is i . (The formulation we provide is inspired to the sequence-form proposed in [9].) 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 . Extension to the general case is straightforward. Our solving algorithm develops into two stages.

In the first stage we check whether 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

²We recall that a behavioral strategy of an agent in a given decision node is the strategy conditioned by the agent being at such node.

problem in which $\alpha_{i,j,s}$ are the decision variables ($C \setminus i$ is the set obtained by removing the element i from C):

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

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

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

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

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

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

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 cells; constraints (4)-(5) express the Markov hypothesis over the patroller's decision policy; constraints (6) express that no action *enter-when*(z, w) gives to the intruder an expected utility larger than that of *stay-out*. (Note that, with $|H| = 1$, intruder's action *enter-when*(H, j) reduces to *enter-when*(i, j), where i is the current cell of the patroller.) The non-linearity is due to constraints (5). If the above problem admits a solution, the resulting $\alpha_{i,j,s}$ are the optimal patrolling strategy. When the above problem is unfeasible, we pass to the second stage of the algorithm.

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

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

s.t.

constraints (1)-(5)

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

The objective function maximizes the patroller's expected utility. Constraints (7) express that no action *enter-when*(z, w) gives a larger value to the intruder than action *enter-when*(s, q). We can formulate n^2 above problems, for all the possible *enter-when*(s, q) actions with $s, q \in C$ (recall that $|C| = n$). If a problem is feasible, its solution is a set of probabilities $\alpha_{i,j,s}$, that define a possible patrolling strategy. From all the solutions of feasible problems, we pick out the one that gives the patroller the maximum expected utility. We stress that a bilinear programming problem can result to be unfeasible. This can happen when the assumption that action *enter-when*(s, q) be the best response is wrong. Indeed, it can be that for any possible patroller's strategy there exists an action, different from *enter-when*(s, q), which gives the intruder a better expected utility. However, as discussed in [19], there exists at least one action of the intruder such that the corresponding bilinear problem is feasible. Note that this second stage of the algorithm requires solving n^2 problems, each one with $O(n^3d)$ variables and constraints.

We report in Fig. 2 the optimal patroller's strategy for the setting of Fig. 1, as calculated with the algorithm described above, considering that the patroller can sense its current cell and adjacent cells. The expected utility for the patroller is 0.845 and the induced best response for the intruder is *enter-when*(01,10), namely to enter cell 10 when the patroller is in 01. Note that cells 04 and 10 are excluded from the optimal patrolling strategy. This makes sense, since the patroller, due to its sensing capabilities, is able to patrol them from adjacent cells that are more "central" (03 and 09, respectively). The patroller uses the strategy to select its actions. For example, whenever the patroller is in cell 01, it randomly chooses its next action between *move*(02) and *move*(05) with probability 0.49 and 0.51, respectively.

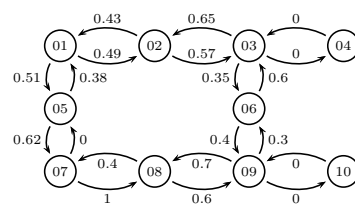


Figure 2: Optimal patrolling strategy for Fig. 1

4. IMPROVING EFFICIENCY

In this section we provide some ways to improve the efficiency of our solving algorithm. The basic idea is that in realistic settings the number of *targets* (i.e., cells over which the intruder has a strictly positive payoff) is usually much smaller than n (i.e., the number of cells in C). In this cases, we can reduce the searching space, in terms of the number of variables and constraints, of the programming problems stated in the previous section. In the following, we suppose that the patroller is able to sense only its current cell.

Our simplifying method develops into two steps. Given a game representing a patrolling scenario, as described in Section 3.2, in the first step, we check whether the patroller can capture the intruder by following a deterministic strategy. A deterministic strategy is a sequence of targets such that all the targets appear in the sequence and no target i is leaved uncovered for more than d_i turns when the sequence is cyclically repeated. We call it "deterministic", since such a strategy does not prescribe the patroller to randomize over the next cell to visit, but it says exactly what is the next cell to visit. Let us focus on the case wherein every target has the same penetration time, say d . In this case, searching for a deterministic strategy can be addressed in the following way. At first, we search for a minimum-length sequence visiting all the targets. This can be calculated by an integer linear programming problem. Then we check whether no target is leaved uncovered for more than d turns when the strategy is cyclically repeated. In the affirmative case, the found sequence assures the patroller to capture always the intruder. Otherwise, no deterministic strategy exists and we apply the second step in which we reduce the number of bilinear programming problems to solve, reducing the number of possible intruder's best responses and the number of constraints per bilinear programming problem. Let us discuss in more detail these two steps.

A deterministic patrolling strategy starts from a target, ends in the same target, and covers all the targets. This

strategy can be efficiently represented as a sequence of independent paths, each one connecting a pair of targets. The minimum-length sequence visiting all the targets is the one that minimizes the largest number of turns between two successive visits to a target. To determine it, we use the following model in which we distinguish the targets from “common” cells. We consider a triple (C, N, A) , where C is the set of cells, $N \subseteq C$ is the set of targets, and $A = \{A_{i,j}\} \forall i, j \in N, i \neq j$, where $A_{i,j} = \{a_{i,j}^k\}$ is the set of minimum-length paths connecting targets i and j ($a_{i,j}^k$ is the k -th minimum-length path that can be thought as a directed arc connecting i and j in N). The length of a path is the number of cells it traverses; for example, in Fig. 3, the length of the minimum path between targets 06 and 08 is 5. Minimum paths can be determined with the well-known Dijkstra’s algorithm. For every cell $c \in C$, we introduce a function $g_c(\cdot)$ such that $g_c(a_{i,j}^k) = 1$ if and only if c belongs to the path $a_{i,j}^k$ and $g_c(a_{i,j}^k) = 0$ otherwise. Given the above definitions, the problem of finding a deterministic strategy can be formulated as an integer linear program. The decision variables are $x_{i,j}^k$ s, with $x_{i,j}^k \in \{0, 1\}$ and $x_{i,j}^k = 1$ if and only if path $a_{i,j}^k$ is selected to be part of the deterministic patrolling strategy and $x_{i,j}^k = 0$ otherwise. According to this formulation, a deterministic strategy is given by the union of all the selected minimum-length paths connecting pairs of targets, namely $\{a_{i,j}^k | x_{i,j}^k = 1\}$. The problem of finding a deterministic strategy can be formulated as a modified version of the well-known traveling salesman problem [20]:

$$\min \sum_{c \in C} \sum_{i,j \in N} \sum_{k=1}^{|A_{i,j}|} x_{i,j}^k g_c(a_{i,j}^k)$$

s.t.

$$\sum_{j \in N} \sum_{k=1}^{|A_{i,j}|} x_{i,j}^k = 1 \quad \forall i \in N \quad (8)$$

$$\sum_{j \in N} \sum_{k=1}^{|A_{j,i}|} x_{j,i}^k = 1 \quad \forall i \in N \quad (9)$$

$$\sum_{i,j \in S} \sum_{k=1}^{|A_{i,j}|} x_{i,j}^k \leq |S| - 1 \quad \forall S \subset N \quad (10)$$

The objective is to minimize the total length of the patrol tour. Constraints (8) and (9) impose that every target is visited at least once and that for every pair of targets at most one connecting path can be selected; constraints (10) remove subtours. For example, in Fig. 3 a deterministic strategy for the represented setting is shown with gray arrows. Call u_i the maximum number of turns between two successive visits of target i when cyclically executing a deterministic strategy determined as above. If $\forall i \in N, u_i \leq d$ then a deterministic strategy is the optimal strategy, assuring that the patroller will always capture the intruder, if it attempts to enter.

We now discuss the second step of our method. The number n of cells that compose the environment significantly affects the computational time of our algorithm. The larger n , the more the time needed to solve the multi-bilinear problem, because the number of variables and constraints it involves grows. As it is reasonable to consider, some cells will be never patrolled at the equilibrium; hence these cells can be removed from the mathematical programming problem. This is the basic idea of the second step we propose. Starting from the integer linear problem above, we introduce decision

variables y_c s, with $y_c \in \{0, 1\}$ and $y_c = 1$ if and only if cell $c \in C$ belongs to at least one path $a_{i,j}^k$ of the best deterministic strategy and $y_c = 0$ otherwise. Then, the significant set of cells to concentrate on can be determined with the following integer linear problem:

$$\min \sum_{c \in C} y_c$$

s.t.

$$\sum_{k=1}^{|A_{i,j}|} x_{i,j}^k = 1 \quad \forall i, j \in N \quad (11)$$

$$y_c \geq \left(\sum_{i,j \in N} \sum_{k=1}^{|A_{i,j}|} x_{i,j}^k g_c(a_{i,j}^k) \right) / M \quad \forall c \in C \quad (12)$$

The objective is to minimize the number of cells over which the patroller will randomize. Constraints (11) impose that for every pair of targets (i, j) , one connecting path from i to j is selected; constraints (12) link the two sets of decision variables, imposing y_c to be equal to 1 if cell c is visited at least one time in the solution (M is an arbitrary very large value). The reduced set of cells we can consider is given by $R = \{c \in C | y_c = 1\}$. Every strategy randomizing over a cell $i \in C \setminus R$ is non-optimal. Therefore, considering only the cells in R , we can reduce n and consequently the computational time. In practice, we can now apply the algorithm of Section 3.4 only to the cells in R . For example, in the patrolling setting of Fig. 3, the set R is composed of the cells on the path of the deterministic strategy.

Moreover, we can further improve the efficiency of our algorithm by reducing the number of constraints per bilinear problem and the number of bilinear problems to be considered. At first, we can reduce the possible intruder’s best responses to *enter-when*(H, z) where z is a target. Then, we introduce the concept of action dominance for further reduction. Action *enter-when*(H', z) dominates action *enter-when*(H'', z) if and only if for every path, for which the patroller can reach target z starting from $c_{|H'|}$ within d_z steps, it passes through $c_{|H''|}$. If an action ξ is dominated by action ξ' , then the probability for the intruder of being captured when playing ξ is larger than playing ξ' . For example, in Fig. 3, the actions *enter-when*($H, 12$) with $c_{|H|} = 11$ are dominated by *enter-when*($H', 12$) with $c_{|H'|} = 06$ or with $c_{|H'|} = 18$. In this case, we can avoid to solve the bilinear problems associated to (in which the intruder’s best responses are supposed to be) *enter-when*($H, 12$) with $c_{|H|} = 11$.

Applying both the reduction of cells (i.e., using the set R) and the elimination of dominated actions, the number of bilinear problems to be solved for the setting of Fig. 3 falls from 29^2 to 18. Moreover, the computational time for solving a single bilinear problem for the same setting decreases from more than 30 minutes to 4.22 minutes on average, when we model our mathematical programming problems with AMPL [5] and we solve them by SNOPT 7.2 solver [17] on a Pentium R 3 GHz 1 GB RAM Linux computer.

5. ENVIRONMENT TOPOLOGY AND $|\overline{H}|$

We have seen that finding the solution for our patrolling game requires to set the length $|H|$ of the history and that histories longer than a threshold $|\overline{H}|$ give the patroller the maximum expected utility. In this section we report some results on the relation between $|\overline{H}|$ and the patrolling setting.

		01 (1,0) $d_{01} = 1$	02 (1,0) $d_{02} = 1$	03 (1,0) $d_{03} = 1$			
04 (1,0) $d_{04} = 1$	05 (1,0) $d_{05} = 1$	06 (2,6) $d_{06} = 7$		07 (1,0) $d_{07} = 1$	08 (3,8) $d_{08} = 7$		09 (1,0) $d_{09} = 1$
10 (1,0) $d_{10} = 1$		11 (1,0) $d_{11} = 1$	12 (1,2) $d_{12} = 9$		13 (1,0) $d_{13} = 1$	14 (2,7) $d_{14} = 9$	15 (1,0) $d_{15} = 1$
16 (1,0) $d_{16} = 1$	17 (1,0) $d_{17} = 1$	18 (2,4) $d_{18} = 7$			19 (1,0) $d_{19} = 1$		20 (1,0) $d_{20} = 1$
		21 (1,0) $d_{21} = 1$	22 (1,0) $d_{22} = 1$	23 (1,0) $d_{23} = 1$	24 (1,0) $d_{24} = 1$		25 (1,0) $d_{25} = 1$
				26 (1,0) $d_{26} = 1$	27 (1,0) $d_{27} = 1$	28 (1,0) $d_{28} = 1$	29 (1,0) $d_{29} = 1$

Figure 3: Another patrolling setting

We show that when the environment has a fully connected topology $|\overline{H}| = 0$ and that, for some topologies, $|\overline{H}| > 1$. We initially consider the situation when the topology is fully connected. We state the following theorem.

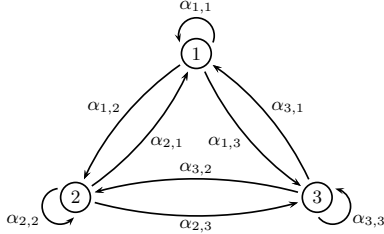


Figure 4: A setting with fully connected topology

THEOREM 5.1. For a fully connected topology, $|\overline{H}| = 0$.

Proof sketch. We prove that, in an environment with fully connected topology, our algorithm produces the same leader-follower equilibrium when $|H| = 0$ and $|H| = 1$. We consider the basic case with three cells, $d_i = 2$ for any cell i . The proof in the general case with more complex patrolling settings and $|H| > 1$ is an easy generalization and we omit it. Suppose $|H| = 1$. Fig. 4 shows an example of this setting, where 1, 2, 3 are cells and $\alpha_{i,j,s}$ are the patroller's strategy. Consider the bilinear programming problem of Section 3.4 in which the best response of the intruder is supposed to be *enter-when*(1,2). The objective function can be written as $X_2 \cdot (1 - p) + X_0 \cdot p$ where $p = \alpha_{2,1} + \alpha_{1,1}\alpha_{1,2} + \alpha_{1,3}\alpha_{3,2}$ is the probability to capture the intruder in cell 2. Since $X_0 \geq X_2$, the maximization of the objective function can be reduced to the maximization of p . The constraints are (a) $EU_i(\text{enter-when}(1,2)) \geq EU_i(\text{enter-when}(2,2))$ and $EU_i(\text{enter-when}(1,2)) \geq EU_i(\text{enter-when}(3,2))$, and (b) $EU_i(\text{enter-when}(1,2)) \geq EU_i(\text{enter-when}(i,j))$ for $i \in \{1,2,3\}$ and for $j \in \{1,3\}$. Consider the first constraint of (a). It can be written as (with the same reduction used for the objective function): $\alpha_{1,2} + \alpha_{1,1}\alpha_{1,2} + \alpha_{1,3}\alpha_{3,2} \leq \alpha_{2,2} + \alpha_{2,1}\alpha_{1,2} + \alpha_{2,3}\alpha_{3,2}$. The second constraint of (a) can be written analogously. Since the objective of the patroller is to maximize $\alpha_{1,2} + \alpha_{1,1}\alpha_{1,2} + \alpha_{1,3}\alpha_{3,2}$, we have that the maximum is when either $\alpha_{1,2} = \alpha_{2,2} = \alpha_{3,2} = 0$ or $\alpha_{i,j} = \alpha_{k,j}$ for all i, j, k . The first option is not possible, since it prescribes that the patroller never patrols cell 2 knowing that the best response of the intruder

is to enter cell 2. Thus, the second option holds. Under this option, the mathematical problem reduces to the one with $|H| = 0$ and therefore they admit the same solution. In the situation in which constraints (b) are more strict than constraints (a), the corresponding problem with $|H| = 0$ results unfeasible and there is an action of the intruder such that the utility expected by \mathbf{p} is larger than that expected when *enter-when*(1,2). \square

Hence, when the topology is fully connected, the patroller's optimal strategy does not depend on the history. The patroller will repeat the same strategy at each turn.

Since by Theorem 5.1 we know that the patrolling strategy obtained with $|H| = 0$ is optimal for fully connected topologies, we can compare our model with that discussed in [12, 13]. For example, consider the setting of Fig. 4 with a single intruder's type and $d_1 = d_2 = d_3 = 2$, $X_0 = 1$, $X_1 = X_2 = X_3 = 0.1$, $Y_0 = -0.3$, $Y_1 = Y_2 = Y_3 = 0.9$. Adopting the model of [12, 13], the actions available to the patroller \mathbf{p} are the possible routes (sets) of two cells, i.e., $\{1,2\}$, $\{2,3\}$, and $\{3,1\}$, whereas the actions available to the intruder \mathbf{i} are the attempts to enter in the three cells, i.e., 1, 2, and 3. In this model the optimal strategies prescribe that \mathbf{p} randomizes uniformly over its three actions, whereas \mathbf{i} can make indifferently one of its three actions. The utilities expected by \mathbf{p} and \mathbf{i} are 0.7 and 0.1, respectively. However, in practice, these utilities are not obtainable, since \mathbf{i} can improve its utility by observing the specific realization of the patroller and then acting on the basis of this observation. More precisely, if \mathbf{i} waits for a turn observing the action taken by \mathbf{p} and then chooses to enter the cell just patrolled by \mathbf{p} , the utilities expected by \mathbf{p} and \mathbf{i} become 0.4 and 0.5, respectively. This leads a rational intruder to violate the rules prescribed by [12, 13]. When we adopt our model, the actions available to \mathbf{p} are *move*(1), *move*(2), and *move*(3), whereas the actions available to \mathbf{i} are *enter-when*(\cdot ,1), *enter-when*(\cdot ,2), and *enter-when*(\cdot ,3), plus *stay-out*. The strategies produced by our algorithm prescribe: \mathbf{p} randomizes uniformly over its three actions, whereas \mathbf{i} can make indifferently one of its three actions *enter-when*(\cdot , \cdot). With our approach, the utilities expected by \mathbf{p} and \mathbf{i} are 0.6 and 0.28, respectively. Hence, in practice, our approach gives the patroller a larger expected utility (0.6 vs. 0.4) than that given by the approach in [12, 13].

$|\overline{H}| = 0$ holds only for fully connected topologies. With other topologies, it is generally required that $|H| \geq 1$ and then $|\overline{H}| \geq 1$. In general, the value of $|\overline{H}|$ depends on the specific setting and, in particular, on the topology and the intruder's penetration times. We report an example to show that, for some topologies, $|\overline{H}| > 1$. Consider Fig. 5 and assume $X_0 = 1, Y_0 = -\varepsilon$ with $\varepsilon > 0$ and arbitrary small. Note that the targets for the intruder are cells 04, 05, and 10. We assume that the patroller can sense only its current cell. We show that, for this setting, $|\overline{H}| > 1$. The gray arrows in Fig. 5 denote the deterministic patrolling strategy calculated as shown in the previous section. Its length is 12. Being as long as the minimum intruder's penetration time in a target cell, such deterministic strategy is the optimal patroller's strategy. The utility expected by \mathbf{p} when following this strategy is 1. Our approach produces this strategy with a suitable $|H| > 1$. To prove it, we suppose $|H| = 1$ and show that, with this value, our approach does not find the optimal patrolling strategy. In general, it is possible to find a value for ε such that \mathbf{i} will always prefer to enter a cell belonging to

