A Game-Theoretic Approach to Determining Efficient Patrolling Strategies for Mobile Robots

Francesco Amigoni Nicola Gatti Antonio Ippedico
Dipartimento di Elettronica e Informazione
Politecnico di Milano
Piazza Leonardo da Vinci 32, 20133 Milano, Italy
{francesco.amigoni, nicola.gatti}@polimi.it, antonio.ippedico@gmail.com

Abstract

Use of game-theoretic models to address patrolling applications has gained increasing interest in the very last years. The patrolling agent is considered playing a game against an hostile possible intruder. Game-theoretic approaches allow the development of patrolling strategies that take into account a model of the intruder. In this paper, we adopt a game-theoretic approach for finding an efficient strategy for a patrolling mobile robot. We consider, differently from most recent approaches presented in literature, patrolling as an extended-form game that can be solved, in an approximated way, by reduction to a sequence of repeated strategic-form games, each one modeling the decision of where the robot should go in the next step. The solution of these games, found with mathematical programming tools, constitutes the patrolling strategy for the robot. Experimental results show the effectiveness of our approach.

1. Introduction

The very last years have seen an increasing number of attempts to approach patrolling applications from the point of view of game theory [9, 10]. The patrolling agent is considered playing a game against an hostile adversary (e.g., a possible intruder). The main advantage of game-theoretic approaches to patrolling is the possibility to take into account a model of the intruder in developing a *patrolling strategy*, namely in finding where the patrolling agent should move. Works presented in literature so far do not fully address the case in which the patrolling agent is a mobile robot.

In this paper, we propose a game-theoretic approach for finding an efficient strategy for a patrolling mobile robot. In particular, we present: a model of the game played by the patrolling robot and the intruder, a mathematical programming formulation for calculating the strategy that the patrolling robot should follow, and an extensive set of experimental activities to validate our approach.

Specifically, we consider the following situation. There are n places of interest to be patrolled by a robot that is supposed to be able to move between any pair of places and that is equipped with an omnidirectional sensor able to perfectly detect intruders within its field of view. Distances between places and sensor range are such that only a place at a time can be covered by the sensor; thus the robot can look for possible intruders in only a place at a time. We also suppose that the environment can dynamically change during the operation of the patrolling robot. In Section 2, we show that the most recent game-theoretic model for patrolling, presented in [9, 10], is not completely adequate for calculating the patrolling strategy for a mobile robot in the considered situation [5]. This is because it is a strategic-form game, namely a game in which the players take their actions simultaneously (like in rock-paper-scissors game), while strategic patrolling intrinsically presents a sequential structure. Moreover, the model in [9, 10] does not deal with possible dynamic changes in the environment. In Section 3, we propose an alternative model, that is an extensive-form game, namely a game where the players act sequentially (like in chess). In Section 4, we present an approach to study this game in an approximated way, in presence of possible unpredictable dynamic changes. The proposed approximation can be solved by exploiting mathematical programming tools. Finally, experimental results of Section 5 show that our approach is more efficient (i.e., it guarantees larger utility to the robot) than alternative approaches.

2. State of the Art

In the last years, several methods for developing autonomous mobile robots that patrol environments looking for intruders have been proposed, based on (partially observable) Markov decision processes [11, 12], on chaos theory [8], and on frequency based approach [2]. These methods do not usually consider any model for the possible intruders. Therefore, they usually show good covering of the environment and good unpredictability of the trajectory followed by the robots, but not always a good efficiency in patrolling (e.g., measured according to the utility gained by the robots). Conversely, in the approach we propose in this paper we take into account a model of the intruder in order to develop efficient patrolling strategies for the robots.

Strategic models for patrolling find a "natural" origin in von Neumann's *hide-and-seek game* [4]. This is a zero-sum strategic-form game, wherein a hider agent $\mathbf{h}$ chooses one cell of a two-dimensional grid in which to hide itself and a seeker agent $\mathbf{s}$ chooses a subset of cells of the grid (usually one row and one column) in which to seek the hider. If the agent $\mathbf{s}$ seeks the cell chosen by the agent $\mathbf{h}$, then the hider is caught. When the hider is caught, its payoff is -1 and the seeker's payoff is 1, and, when the hider is not caught, agents' payoffs are the inverse. As is customary for strategic-form games, the appropriate solution concept for the hide-and-seek game is Nash equilibrium: a pair of strategies $\boldsymbol{\sigma}^* = (\sigma^*_{\mathbf{h}}, \sigma^*_{\mathbf{s}})$, where $\sigma^*_{\mathbf{h}}$ and $\sigma^*_{\mathbf{s}}$ are the strategy of the hider and of the seeker, respectively, such that, when the seeker employs the strategy $\sigma^*_{\mathbf{s}}$, the hider cannot employ any strategy better than $\sigma^*_{\mathbf{h}}$, and vice versa.

The most significant recent extension to strategic patrolling applications of the von Neumann's hide-and-seek game has been proposed by Paruchuri *et al.* in [10]. They study the situation in which a guard (i.e., the seeker) patrols some areas and a robber (i.e., the hider) can observe the guard's strategy before undertaking its actions. There are n areas and time is discretized in turns, such that the robber spends $1 \leq d < n$ turns to rob an area and the guard in one turn can move between any pair of areas and patrol the destination area. Although the authors make explicit use of turns in the game, the model they study holds to be in strategic-form: the robber chooses an area to rob and simultaneously the guard chooses a route of d areas, e.g., $\langle 1, 2, \ldots, d \rangle$. The game is general-sum and agents can assign each area different evaluations (e.g., the robber could prefer robbing area 1 than robbing area 2). The possibility for the robber to observe the actions of the guard before undertaking its actions radically influences the analysis of the game. The authors implicitly assume that the guard will continuously repeat its strategy every d turns and, therefore, by observation the robber can derive a correct belief over the guard's strategy. Given this belief, the robber essentially behaves as a best responser and its strategy will be pure. By common knowledge, the guard is aware that the robber will observe its actions before acting and can take advantage from that. Technically speaking, the guard can (non strictly) increase its expected utility by employing a *commitment-based* strategy [13] (also called *Stackelberg* strategy and *leader-follower* strategy). This strategy maximizes the expected utility of the committer, in this case the guard, when it can commit to a strategy first and then a follower, in this case the robber, selfishly optimizes its own reward, considering the actions chosen by the committer [13]. In any two-player game, commitment-based equilibria are never worst than the best Nash equilibrium for the committer. The problem of computing commitment-based equilibria in any two-player game is a linear programming problem [3]. The result will usually prescribe a fully mixed strategy for the guard. Finally, robber can be of different *types*, with a given probability distribution known by the guard. Robber types differ in the payoffs, modeling the fact that different types of robbers can have different motivations. With multiple types, the problem of computing commitment-based equilibria can be computationally hard, although some efficient algorithms have been proposed [9, 10].

The above model by Paruchuri *et al.* presents some game theoretical inconsistencies due to the discrepancies between the model and the real setting it aims at capturing [5]. In the model, the guard first selects a realization, in this case a route, according to its mixed strategy and then executes it entirely, while the robber repeatedly observes a number of guard's routes and, once it has a correct belief over the guard mixed strategy, chooses what area to rob. By construction, the model implicitly assumes that the robber enters into an area exactly in the turn in which the guard starts a route. In a real setting, the robber could enter an area while the guard is already patrolling a given route, and not only when it starts a new route. In [5], it is provided a simple example in which the robber's optimal strategy is to wait for one turn to observe the area patrolled by the guard and then to rob exactly that area. This strategy is shown to increase the robber's expected utility wrt the strategy prescribed by the algorithms provided in [9, 10]. It easily follows that the model by Paruchuri *et al.* is not adequate to capture the setting it aims at and, in particular, it is not adequate to capture the setting we are considering in which the guard is a mobile robot. As discussed in [5], the inconsistencies are due to the fact that the game, although modeled as a repeated strategic-form game, is actually an extensive-form game.

For the sake of completeness, we also cite the very recent work in [1]. Considering a cyclic patrol path, the authors determine the probability for the guard to switch the patrolling direction in order to maximize the probability of detecting penetrations in the weakest spots of the environment. Their approach differs from ours since, although we both consider the presence of adversaries, we also consider the robber's preferences. Hence, we look for an equilibrium solution and not for a maximin solution, as in [1].

3. An Extensive-Form Model

In this section we attempt to provide an appropriate game model for strategic patrolling when the guard is a robot. First we extend the model [10] described in the previous section to an extensive-form game. Then, we refine the proposed model to make it closer to real settings.

We formulate the game as an extensive-form game wherein at each step the robot $\mathbf{r}$ chooses the next place to visit and simultaneously the intruder $\mathbf{i}$ (e.g., a robber) chooses whether or not to intervene and, in the former case, what place to enter. Time needed for the intruder to enter a place is d. Then, if the intruder decides to enter a place at time $\bar{t}$, it cannot take any action until $\bar{t}+d$, whereas, if the intruder decides to wait at $\bar{t}$, at $\bar{t}+1$ it can take any possible action. While the intruder is waiting, it can observe the actions performed by the robot. On the other hand, the robot cannot observe the actions undertaken by the intruder unless it visits a place in which the intruder is present or that has been previously entered. If the intruder enters a place at $\bar{t}$ and the robot visits this place at least one time before $\bar{t}+d+1$, then the intruder will be caught, otherwise it will have success. The game ends when the intruder either is caught or has success. The game is thus an infinite horizon extensive-form game with imperfect information. The infinite horizon is due to the fact that the robot continuously patrols the places and the intruder can wait forever. The information imperfectness is related to the non-perfect observability of the robot over the intruder's actions: whenever the robot is in a place where the intruder is not present, it cannot know whether the intruder is waiting or has entered another place. Similarly, if the intruder decides to enter a place, it does not know the place currently patrolled by the robot. A portion of the game tree in a situation with 3 places of interest is reported in Fig. 1. The dotted line connecting different intruder's decision nodes means that the intruder cannot discriminate between such decision nodes due to information imperfectness. The dotted-line box will be explained in the next section. Agents' payoffs are defined as follows: the intruder and the robot assign each place i a value $v_i^{\mathbf{i}}$ and $v_i^{\mathbf{r}}$, respectively, the intruder evaluates its catch with a value t (usually, $t<0$) and the robot evaluates the catch of the intruder with a value u. Given an outcome (represented by a path from the root to a leaf of the game tree), the utility of the robot will be the sum of the values of the preserved places and of u if the intruder was caught, whereas the utility of the intruder will be the value of the entered place, if it was not caught, and t otherwise. As in [10], the intruder can be of more types, making the robot not knowing with certainty the intruder's payoffs.

Now we refine the model to make it closer to concrete settings with patrolling robots. The first refinement we introduce captures robot's costs. The robot will incur in

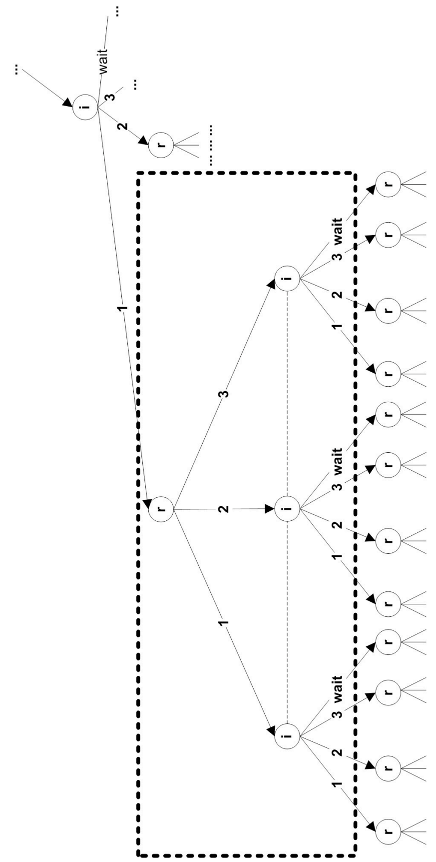

Figure 1. A portion of the game tree

some costs to move between places, due to energy and time needed for movements. We call $C_{i,j}$ the cost for the robot to reach place j starting from place i. This cost negatively affects the payoff received by the robot. Formally, the costs of the movements performed by the robot will be subtracted from the values of the preserved places. The second refinement we introduce captures the dynamicity of the environment. During the game, several events can happen, e.g., costs could increase because the path between two places closes or is hardly traversable due to the presence of some obstacle. We capture this issue by considering values $v_i^{\mathbf{r}}$ and $v_i^{\mathbf{i}}$ and costs $C_{i,j}$ as time dependent.

The appropriate solution concept for the game we are considering is the Kreps and Wilson sequential equilibrium [7]. A sequential equilibrium is a pair $\langle \beta, \boldsymbol{\sigma}^* \rangle$, where β is a system of beliefs that is consistent with $\boldsymbol{\sigma}^*$, and $\boldsymbol{\sigma}^*$ are the agents' strategies that are sequentially rational given β. The strategies $\boldsymbol{\sigma}^*$ define the optimal agents' actions in each possible decision node.

Although our extensive-form model satisfactorily captures the robot patrolling domain, its solution is not easy. The first reason is the implicit assumption that the robot is clairvoyant about the events that will happen in the environment. This assumption is not reasonable in concrete settings, since some events will be absolutely unpredictable.[1] In practice, the robot can just compute a solution on the basis of the information it has actually gathered. It follows that the robot must compute a new solution for the game whenever it knows that the game has changed. This requires that the robot must be able to find solutions online. The second reason concerns the process of solving an imperfect information extensive-form game with infinite horizon. To the best of our knowledge no algorithm can efficiently solve such games. Just finite horizon imperfect information extensive-form games find satisfactory solvers in literature (e.g., [6]). Moreover, these algorithms do not scale in presence of a large number of places and intruder's types. These issues push towards the definition of an approximated approach to solve the extensive-form game and find the robot's patrolling strategy.

4. An Approximated Strategic-Form Model

The main idea underlying our approximated model is to consider a *slice* of the extensive-form game at a time. The solution to the overall problem will be obtained by solving sequentially all the single slices.

A slice of the extensive-form game is defined as a decision node of the guard and all the immediate successors where the intruder acts. In Fig. 1 the dotted-line box delimits a slice. A slice is thus a step of the extensive-form game in which the robot and the intruder simultaneously make one action, the robot choosing the next place to patrol, while the intruder choosing whether or not to enter a place and, in the first case, what place to enter. A single slice can therefore be intended as a two-player strategic-form game, and all the slices are strategically equivalent, meaning that the actions available to players are the same. The slices can differ in terms of payoffs: movement costs and place values can change slice by slice due to events happening in the environment (this is equivalent to say that $v_i^{\mathbf{r}}$, $v_i^{\mathbf{i}}$, and $C_{i,j}$ are time-dependent in the extensive-form game). The robot's payoffs in a given slice should be (the sum of) the values of the places it has preserved at that slice, minus the cost of the movement performed in that slice, plus u if the intruder is caught. However, the robot cannot know the global values of the preserved places in a given slice, since it cannot perfectly observe the actions of its opponent. For instance, if the robot is visiting place 1 and the intruder is not there, the robot cannot know whether or not it has entered some other place. Our idea is the following: the only place that is preserved for sure is the place currently patrolled by the robot. (We suppose that the robot has no way to obtain information about other places.) Thus, if, in a slice, the robot chooses to patrol place j starting from place i, then the robot's payoff will be $v_j^{\mathbf{r}} + C_{i,j}(+u)$. The definition of the intruder's payoffs is straightforward. If the intruder does not enter any place, then it will receive 0. If the intruder enters place i, it will receive $v_i^{\mathbf{i}}$ in the case it is not caught and t otherwise. An example of payoff matrix of the approximated strategic-form model (i.e., of a slice) in a situation with three places $1, 2$, and 3 is reported in Tab. 1 (patrolling robot actions are on the columns and intruder actions are on the rows, l is the place in which the robot is). The intruder can be of different types, according to a given probability distribution over types. Types differ in terms of values $v_i^{\mathbf{i}}$ and t.

Table 1. Payoff matrix

	1	2	3
1	$t, v_1^{\mathbf{r}} + u - C_{l,1}$	$v_1^{\mathbf{i}}, v_2^{\mathbf{r}} - C_{l,2}$	$v_1^{\mathbf{i}}, v_3^{\mathbf{r}} - C_{l,3}$
2	$v_2^{\mathbf{i}}, v_1^{\mathbf{r}} - C_{l,1}$	$t, v_2^{\mathbf{r}} + u - C_{l,2}$	$v_2^{\mathbf{i}}, v_3^{\mathbf{r}} - C_{l,3}$
3	$v_3^{\mathbf{i}}, v_1^{\mathbf{r}} - C_{l,1}$	$v_3^{\mathbf{i}}, v_2^{\mathbf{r}} - C_{l,2}$	$t, v_3^{\mathbf{r}} + u - C_{l,3}$
wait	$0, v_1^{\mathbf{r}} - C_{l,1}$	$0, v_2^{\mathbf{r}} - C_{l,2}$	$0, v_3^{\mathbf{r}} - C_{l,3}$

It is reasonable to consider the commitment-based approach in our model, because the robot can repeat its strategy over the slices and the intruder can observe it. Each slice is solved by looking for commitment-based equilibria. A solution is a mixed strategy for the patrolling robot that maximizes its expected utility in the slice. The patrolling strategy of the robot, that should come from the solution of the extended-form game, is thus approximated by a sequence of steps that come from the solution of the strategic-form games (slices). Note that both our approximated model and the model in [9, 10] are strategic-form games, the main difference being that, in our case, the robot's actions refer the *next* place to patrol, while, in the case of [9, 10], the guard's actions are multi-step patrolling routes.

In what follows we formulate the problem of searching for commitment-based equilibria in a single slice similarly to the mathematical programming formulations presented in [10] and in [9], respectively. The two formulations are equivalent; we will evaluate their performance in finding the solution of a slice in the next section. For the sake of simplicity, we report only the case with one type of intruder; with multiple types, formulations are easy extensions (refer to [9] for more details).

We first provide a multi-LP (multi linear programming) formulation. We call $\gamma_i \in [0,1]$ with $\sum_i \gamma_i = 1$ the probability for the robot to perform the action of going to check place i. The optimization problem to be solved can be formulated as: find the values of γ_i such that, for any pure strategy $j = \{1, 2, \ldots, n, \text{wait}\}$ of the intruder:

[1] Note that in some settings the intruder could produce events to alter the environment and takes benefit from that. For example, it could drop an obstacle to raise the cost of moving between two places.

$$\max \sum_{i=1,2,\ldots,n} \gamma_i R(i,j)$$

$$\sum_{i=1,2,\ldots,n} \gamma_i I(i,j) \geq \sum_{i=1,2,\ldots,n} \gamma_i I(i,j')$$

$$\forall j' \in \{1,2,\ldots,n,\text{wait}\}, j' \neq j$$

$$\sum_{i=1,2,\ldots,n} \gamma_i = 1 \quad \text{and} \quad \gamma_i \geq 0 \quad \forall i \in \{1,2,\ldots,n\}$$

where $i = 1, 2, \ldots, n$ are the possible actions of the patrolling robot and $R(i, j)$ and $I(i, j)$ are the payoffs of the patrolling robot and of the intruder, respectively, when the first one performs action i and the second one performs action j. The matrices R and I are based on $v_k^{\mathbf{r}}$, $v_l^{\mathbf{i}}$, t, and u defined above. For example, considering Tab. 1, $R(1, \text{wait}) = v_1^{\mathbf{r}} - C_{l,1}$ and $I(1, \text{wait}) = 0$, when l is the current place of the robot. We formulate $n + 1$ problems, one for each pure strategy j of the intruder $(1, 2, \ldots, n$, and wait). Each one of these problems is solved to find a set $\{\gamma_i\}$. Among these $n + 1$ sets $\{\gamma_i\}$, the one that gives the largest expected utility to the patrolling robot is the mixed strategy it should adopt in the slice. Specifically, the robot draws a realization from this mixed strategy by randomly selecting an action (a place where to go) according to probabilities γ_is.

Then, we provide a MILP (mixed integer linear programming) formulation:

$$\max \sum_{i=1,2,\ldots,n} \sum_{j=1,2,\ldots,n,wait} z_{i,j} R(i,j)$$

$$\sum_{i=1,2,\ldots,n} \sum_{j=1,2,\ldots,n,wait} z_{i,j} = 1$$

$$\sum_{j=1,2,\ldots,n,wait} z_{i,j} \leq 1, \forall i \quad \text{and} \quad \sum_{j=1,2,\ldots,n,wait} q_j = 1$$

$$0 \leq (a - \sum_{i=1,2,\ldots,n} (I(i,j) \sum_{j=1,2,\ldots,n,wait} z_{i,j})) \leq$$

$$\leq (1 - q_j)M$$

$$z_{i,j} \in [0,1] \qquad q_j \in \{0,1\} \qquad a \in \Re$$

Here, M is a large constant and a is the intruder's maximum expected utility; q_j is a binary variable such that $q_j = 1$ means that the intruder's optimal action is j; $z_{i,j}$ is defined as $z_{i,j} = \gamma_i q_j$.

It is interesting to compare the complexity of the MILP formulation for our model and for the model in [9]. One of the most complex scenarios considered in [9] involves 7 places of interest and 8 types for the intruder (and routes with $d = 2$ steps). In this scenario, the MILP problem for the model [9] is composed of 1240 decisional variables $z_{i,j}$ and 576 constraints, while the MILP problem for the model proposed in this paper is composed of 520 decisional variables and 384 constraints. Our model, therefore, reduces the complexity of finding a solution, even when we consider that in [9] the actions of the patrolling agent are all the possible routes of two places (with no repetitions and no order), while in our case the actions of the patrolling robot are the next places to reach. The longer the route (i.e., the larger d) the more efficient our approach wrt to that in [9].

5. Experimental Results

In this section, we present some of the experimental activities we performed to validate our approach. Multi-LP problems have been solved with the *linprog* built-in function of Matlab 7.0, while MILP problems have been solved with Matlab 7.0, using GLPK[2] 4.9 integrated with GLPKMEX[3] 0.5.9. Experiments have been performed with a 2.8 GHz Pentium 4 computer with 512 MB and Windows XP. We evaluate the performance of our approach according to computational time for finding a patrolling strategy and according to coverage, unpredictability, and utility of the patrolling strategy. We also evaluate the ability of our approach to adapt to dynamic changes in the environment.

Let us initially evaluate the computational time required by our method for determining the mixed strategy for the patrolling robot. We consider a linear environment with n places of interest, evenly separated by 10 units (namely, their positions are $p_1 = 10, p_2 = 20, \ldots, p_n = 10n$), with values $v_i^{\mathbf{r}} = v_i^{\mathbf{i}}$ for all i randomly chosen in $[0, 10]$, the robot starting from origin 0, $u = 1000$, and $t = -100$. The cost $C_{i,j}$ for reaching place j from a place i is calculated as $C_{i,j} = 0.1|p_i - p_j|$ (this amounts to consider for the patrolling robot a translational cost of 0.1 per unit and a null rotational cost). When we consider a single type for the intruder, the computational time for finding the solution of a single slice (namely, for finding the next step to perform in patrolling) when formulating the problem as multi-LP and MILP are reported in Tab. 2, where results are averaged over 10 runs (runs differ in the random values of places). As expected, computational time increases with the number of places n to be patrolled, making our approach adequate for a use onboard a real robot in environments with up to some dozens of places. The results also show that the multi-LP formulation scales much better than the MILP formulation with n. When we consider multiple types for the intruders (each one giving different values to places of interest), the MILP formulation scales better than the multi-LP formulation. Tab. 3 compares the single step computational time for multi-LP and MILP (averaged over 10 runs, '-' shows that the algorithm has been manually stopped after 6000 s).

[2] GNU Linear Programming Kit, `http://www.gnu.org/software/glpk`.

[3] `http://control.ee.ethz.ch/~mpt/`.

Note that the results of Tab. 3 (bottom) are comparable to those reported in [9], even considering that, in that case, experiments have been performed with CPLEX, that is a much more efficient MILP solver than the one we used (unfortunately, we had no possibility to access CPLEX). When solving the model in [9] with our software, we noted that our model finds a patrolling strategy more efficiently. For example, when considering $n = 3$ places and 5 types of intruders, solving the model in [9] requires 0.152 s to find a route of $d = 2$ places, while solving our model requires 0.138 s and, when considering $n = 10$ and 2 types of intruders, the computational time is 2.967 s and 0.223 s, respectively, the difference growing when n increases.

Table 2. Computational time (in s) with a single type of intruder

n (# of places)	multi-LP	MILP
3	0.068	0.006
5	0.079	0.014
10	0.280	0.029
25	1.085	0.752
50	3.702	11.220
100	16.096	215.209

Table 3. Computational time (in s) with multiple types of intruders

multi-LP				
n (# of places)	2 types	3 types	4 types	5 types
3	0.342	1.278	6.854	62.972
5	0.796	13.227	544.481	-
10	16.095	-	-	-
15	146.665	-	-	-

MILP				
n (# of places)	2 types	3 types	4 types	5 types
3	0.015	0.031	0.068	0.138
5	0.033	0.103	0.357	1.128
10	0.223	1.615	4.320	-
15	1.351	8.128	-	-
20	4.706	14.834	-	-
25	12.955	-	-	-

We now turn to evaluate the performance of our approach in terms of coverage, unpredictability, and utility in different environments. Coverage is evaluated as the number of places that are never visited over 500 steps. Unpredictability is the average entropy of the strategy over 500 steps, where the entropy at each step is calculated as $E = \sum_{1,2,\ldots,n} -\gamma_i \ln(\gamma_i)$, where γ_is are the probabilities of selecting to go to place i at the given step. Utility is the average utility (over 500 steps) gained by the patrolling robot at each step. We compared our method with the following techniques:

- *uniform strategy*: given n places of interest, select one with uniform probability $1/n$;
- *proportional strategy*: given n places of interest, select one with probability proportional to the value of places minus the cost for reaching them (the larger the value $v_i^{\mathbf{r}} - C_{l,i}$ the larger the probability of selecting place i);
- *random strategy*: given n places of interest, randomly generate a probability distribution $\{\gamma_i\}$ over the places and then select a place according to this distribution.

We consider environments with $n = 25$ places of interest, whose values $v_i^{\mathbf{r}} = v_i^{\mathbf{i}}$ are randomly chosen in $[1, 100]$, $u = 1000$, and $t = -100$. Costs are calculated considering a translational cost of 0.1 per unit and null rotational cost. We employed the multi-LP formulation. We performed 10 runs of 500 steps each and we present the average values over the runs (runs differ in the random values assigned to places and in the selection of the place to visit according to the mixed strategy found at each step). We consider a linear environment with places evenly separated by 10 units (Tab. 4, top), a ring environment with places evenly separated by 10 units (Tab. 4, middle), and a star environment with any pair of places separated by 10 units (Tab. 4, bottom). From the results, it emerges that our approach guarantees average utility per step consistently larger than that guaranteed by other methods. In this sense, we claim that the patrolling strategy resulting from the approach presented in this paper is efficient. At the same time, our patrolling strategy provides a good coverage and a reasonable unpredictability, although its entropy is worst than that provided by other methods, which are of course “more random”.

Table 4. Performance of patrolling strategies

Linear environment			
method	# of never visited places	entropy	utility
multi-LP	15.4	2.081	78.95
uniform strategy	18.7	3.219	52.124
proportional strategy	12.2	3.014	69.528
random strategy	0.5	3.027	48.028
Ring environment			
method	# of never visited places	entropy	utility
multi-LP	15.4	2.117	82.911
uniform strategy	18.1	3.219	56.149
proportional strategy	10.6	3.019	66.450
random strategy	1.3	3.033	50.494
Star environment			
method	# of never visited places	entropy	utility
multi-LP	14.9	2.169	75.014
uniform strategy	19	3.219	34.831
proportional strategy	16.4	3.007	55.714
random strategy	1.2	3.031	40.444

We finally show the ability of our approach to cope with dynamic changes in the environment. Consider the scenario described at the beginning of this section, with a single type of intruder, a linear environment with $n = 10$ places of interests whose values (equal for robot and intruder) are reported in the second column of Tab. 5 (top). We consider experiments lasting 500 steps. Let us suppose that the intruder does a robbery at step 100 in place 6, making the value of this place null after the event. The last two columns of Tab. 5 (top) report the average number of visits of the patrolling robot to the places before and after the robbery (the average is over 10 runs; we consider the robbery as instantaneous, so the intruder cannot be caught by the patrolling robot). The number of visits to each place increases after the robbery, as the robot does not go anymore to the robbed place 6 (because it has now a value of 0). Note that the number of visits before robbery is over 100 steps, while that after robbery is over 400 steps. This shows that our approach is able to cope with dynamic changes in the environment, in particular, in this case, with changes in the values of the places. A similar experiment is reported in Tab. 5 (bottom). We consider again a linear environment, with $n = 3$, with initial constant costs for moving between places ($C_{l,i} = 1$ for $i = 1, 2, 3$, independently from the starting position l), and with values for places (equal for robot and intruder) reported in the second column of Tab. 5 (bottom). At step 100 the costs of reaching places 1 and 2 change, becoming $C_1 = 1050$ and $C_2 = 1000$. As the average number of visits (over 10 runs of 500 steps each) reported in the table shows, the strategy of the robot changes accordingly, bringing the robot to patrol only places 1 and 3 (after the cost change, place 2 has a low value and a high cost), with slight preference for the latter (place 3 has less value than place 1, but also a much lower cost).

Table 5. Environment changes

A robbery happens			
place i	$v_i^{\mathbf{r}}$	# of visits before robbery	# of visits after robbery
1	95.0628	25.4	109.4
2	23.8827	0	0
3	61.0774	10.7	51.6
4	49.1123	3.6	23.9
5	89.2386	23.6	99.8
6	76.4476 (0)	16.1	0
7	46.1903	0.5	14.4
8	2.8319	0	0
9	82.3193	20.1	88.8
10	45.0256	0	12.1

Costs change			
place i	$v_i^{\mathbf{r}}$	# of visits before change	# of visits after change
1	90	47.9	171.2
2	9	8.5	0
3	30	43.6	228.8

6. Conclusions

In this paper we have presented an approach to determine the patrolling strategy for a mobile robot employed in surveillance tasks. Our approach, differently from others presented in literature, considers patrolling as an extended-form game that can be solved, in an approximated way, by reducing it to a sequence of repeated strategic-form games. Each strategic-form game models the decision of where to go in the next step. The solution of these games constitutes the patrolling strategy for the mobile robot. Experimental results have shown the effectiveness of our approach.

Our contribution constitutes a step towards the development of an autonomous patrolling robot, following a game-theoretic approach. More work needs to be done in order to reach this goal. In particular, we are developing a more complete extended-form model that takes into account some important issues for mobile robots (such as energy consumption and map-related problems). Moreover, we plan to move soon toward the implementation of our approach on a real robot to test its potential on the field.

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. A realistic model of frequency-based multi-robot polyline patrolling. In *Proc. AAMAS*, pages 63–70, 2008.

[3] V. Conitzer and T. Sandholm. Computing the optimal strategy to commit to. In *Proc. EC*, pages 82–90, 2006.

[4] M. Flood. The hide and seek game of von neumann. *Management Science*, 18(5):107–109, 1972.

[5] N. Gatti. Game theoretical insights in strategic patrolling: Model and algorithm in normal-form. In *Proc. ECAI*, 2008.

[6] D. Koller, N. Megiddo, and B. von Stengel. Efficient computation of equilibria for extensive two-person games. *Games and Economic Behavior*, 14(2):220–246, 1996.

[7] D. Kreps and R. Wilson. Sequential equilibria. *Econometrica*, 50(4):863–894, 1982.

[8] L. Martins-Filho and E. Macau. Patrol mobile robots and chaotic trajectories. In *Mathematical Problems in Engineering*. Hindawi, 2007.

[9] 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.

[10] 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.

[11] P. Paruchuri, M. Tambe, F. Ordonez, and S. Kraus. Security in multiagent systems by policy randomization. In *Proc. AAMAS*, pages 273–280, 2006.

[12] S. Ruan, C. Meirina, F. Yu, K. Pattipati, and R. Popp. Patrolling in a stochastic environment. In *Proc. 10th Int'l Command and Control Research Symposium*, 2005.

[13] B. von Stengel and S. Zamir. Leadership with commitment to mixed strategies. CDAM Research Report LSE-CDAM-2004-01, London School of Economics, 2004.