INTECH
open science | open minds

OPEN ACCESS
ARTICLE

International Journal of Advanced Robotic Systems

# Game Theory Models for Multi-Robot Patrolling of Infrastructures

Regular Paper

Erik Hernández[1,*], Jaime del Cerro[1] and Antonio Barrientos[1]

1 Center for Robotis and Automation, UPM-CSIC, Madrid. Spain
* Corresponding author E-mail: ehernandez@industriales.upm.es

Received 14 Jun 2012; Accepted 17 Dec 2012

DOI: 10.5772/55535



Abstract This work is focused on the problem of performing multi-robot patrolling for infrastructure security applications in order to protect a known environment at critical facilities. Thus, given a set of robots and a set of points of interest, the patrolling task consists of constantly visiting these points at irregular time intervals for security purposes. Current existing solutions for these types of applications are predictable and inflexible. Moreover, most of the previous work has tackled the patrolling problem with centralized and deterministic solutions and only few efforts have been made to integrate dynamic methods. Therefore, one of the main contributions of this work is the development of new dynamic and decentralized collaborative approaches in order to solve the aforementioned problem by implementing learning models from *Game Theory*. The model selected in this work that includes belief-based and reinforcement models as special cases is called *Experience-Weighted Attraction*. The problem has been defined using concepts of *Graph Theory* to represent the environment in order to work with such *Game Theory* techniques. Finally, the proposed methods have been evaluated experimentally by using a patrolling simulator. The results obtained have been compared with previous available approaches.

Keywords Multi-Robot Patrolling, Game Theory, Experience-Weighted Attraction, Security Application

## 1. Introduction

Domains, where distributed surveillance, inspection or control are required, are candidates for being secured by performing patrolling tasks, usually by walking throughout at regular intervals (Abate, 1996), (Almeida et al., 2004). Currently, security system solutions are mostly predictable and inflexible. Additionally, since they are controlled by human operators, their performance can be affected by limitations such as boredom, distraction, or fatigue. Furthermore, in some environments, people must deal with hazardous conditions. As a consequence, it is important to improve the security elements used in these types of systems, which assist human beings in dangerous scenarios such as mine clearing or search and rescue operations. They are then able to perform other type of high level tasks, i.e., monitoring the system from a safe location (Oates et al., 2009). Recently, new research efforts have arisen trying to solve some challenges related to security tasks automation by using mobile robots (Everett, 2003). Thus, mobile robots aim to perform some useful task that a human either cannot, or

would prefer not to do. Moreover, the robot should hopefully do it better, cheaper, safer, and more reliably.
Security systems that utilize mobile robots in these types of applications have a great deal of advantages, i.e., they do not experience human limitations. However, some tasks are too complex that a single robot cannot achieve good results, especially in the presence of uncertainties, incomplete information, distributed control, and so forth. To overcome these challenges, *Multi-Robot Systems* can be used. They are characterized as a set of homogeneous or heterogeneous robots operating in the same environment using cooperative behaviors (Farinelli et al., 2004).

In this paper, new collaborative multi-robot approaches for infrastructure security applications at critical facilities are explored. The work is focused on *area patrol*, i.e., the activity of going throughout an area. Thus, given a set of robots and a set of points of interest, the patrolling problem consists of constantly visiting these points at irregular time intervals for security purposes. This problem has been formulated using concepts of *Graph Theory* to represent an environment where nodes stand for specific location of interest and edges for possible paths. By using this representation, each path has a cost that represents the time required to go from one node to another. The main advantage of this representation is its application. It can be easily used in other domains, i.e., computer networks, distributed coverage, and so forth. Additionally, there is a wide variety of problems that may be reformulated as particular patrolling task such as cleaning or surveillance. Since the patrolling problem seeks to maximize the number of visits to each node in a given environment, a good patrolling strategy must reduce the time lag between two visits to the same location (Chevaleyre, 2004).

The main contributions of this work are summarized as follows: An analysis of the behavior of game theory models in the multi-robot patrolling problem context is presented. A dynamic and distributed solution has been developed in order to solve the aforementioned problem. A novel decision-making rule has been defined. This rule attempts to allow robot dispersion, i.e., at each point of interest, each robot chooses a different available set of actions. A demonstration of how multiple robot interaction arises with the definition of multiple games at each point of interest has been defied. Finally, a detailed study of the behavior of the implemented model parameters has been described.

The rest of this paper is organized as follows. Section 2 briefly describes related work. Section 3 gives definitions of game theory and introduces the problem. Section 4 shows the implemented models in order to solve the patrolling problem. Section 5 presents the evaluation and experimental results. Finally, section 6 summarizes the obtained results.

## 2. Related work

The multi-robot patrolling problem has received much attention in recent years, specially in works that develop algorithms to coordinate decision-making among robots, (Portugal and Rocha, 2011). These works have implemented different principles such as reinforcement learning (Santana et al., 2004); negotiations methods (Hwang, 2009); swarm optimization (Glad and Buffet, 2009); cycle and partitioning strategies (Chevaleyre, 2004); and adaptive solutions (Sempé and Drogoul, 2003). A description of all of them can be found on a recent survey by (Portugal and Rocha, 2011). Beyond this survey, the multi-robot patrolling problem was tackled in (Ahmadi and Stone, 2006). In that work, the problem was called *Continuous Area Sweeping*, which is solved with a partitioning area method. Moreover, in (Aguirre et al., 2011), the multi-robot patrolling is applied to patrol national borders. In that work, elements of game theory as well as Monte Carlo simulation are used to solve the problem via genetic algorithms. Another work that utilizes game theory principles is described in (An et al., 2012). In that work, solutions to solve competitive or zero-sum games for the protection of critical infrastructure via *Stackelberg Games* are presented.

Among all these works, three of them are directly related with this work. The pioneer work in the multi-robot patrolling problem was carried out by (Machado et al., 2003). In that work, authors defined an evaluation criterion based on idleness. Idleness is the time that a place remains unvisited. Thus, *Total Idleness* is defined as the average of the idleness of all places of a given environment. Since this criterion is widely used in literature, it was used to measure the performance of the methods proposed in this work. Moreover, the problem of generating a patrol path inside a target area was tackled in (Elmaliach et al., 2007). The algorithm applied to generate this patrol path is called *Cycle* and it guarantees that each point is covered with the same optimal frequency. The solution presented in that work uses Spanning Tree Coverage method to find a minimal Hamilton path of minimal costs. Once a path is obtained, robots are uniformly distributed along this path and follow the same patrol route over and over. Thus, uniform frequency of the multi-robot patrolling task is achieved as long as one robot continues working properly. Moreover, authors present criteria based on frequency optimization in order to evaluate multi-robot patrolling algorithms. Finally, in (Portugal and Rocha, 2010) is presented an algorithm called *MPS*. Such an algorithm divides the environment into regions with the same dimension by using a balanced graph partitioning approach. Each of these regions is assigned to a robot that follows a local patrolling route. The procedure to

obtain this patrolling route mainly seeks Euler and Hamilton circuits and paths. However, if such circuits and paths do not exist, the procedure seeks longest paths and Non-Hamiltonian cycles. Non-Hamiltonian cycles are selected only when they have at least half of the vertices of a graph; if not, the patrolling route remains the longest path. Since the longest path and the Non-Hamiltonian cycle do not contain all vertices of the graph, the procedure includes such vertices to complete the patrolling route. Then, ultimately inverse path procedure is used to return to the starting vertex of the route when is required.

Previous literature has demonstrated the effectiveness of methods that implement solutions based on cycles and paths (Chevaleyre, 2004). The good performance of these approaches could be explained by their centralized and explicit coordinator scheme, (Almeida et al., 2004). However, a centralized solution has several disadvantages such as lack of scalability in the number of places to protect and susceptibility to single-point failure, due to its unique, and hence vulnerable, control point. In addition, these approaches are deterministic, and therefore not suitable for security purposes due to their predictability.

The present work differs from others on the manner in which the patrolling problem was solved by implementing learning models from *Game Theory*. The theory of learning in games defines equilibrium as the result of dynamic adjustment processes in which players interact for optimality over time in repeated normal-form games. Thus, they compute their *myopic* best response based on the accumulated experience achieved by tracking previous plays history of other players. The learning model selected in this work to patrol throughout an environment was proposed by Camerer and is called *Experience-Weighted Attraction* (Camerer, 1999). Implementing such adaptive models allows developing dynamic and distributed solutions similar to (Santana et al., 2004) in contrast to several literature works.

## 3. Concepts from game theory

A brief overview of concepts as well as some definitions of game theory (Fudenberg, 1998) are given in order to clarify the description in the following sections. In this work, an abstract representation of the environment as an undirected weighted graph $G$ has been adopted. This graph is an ordered pair consisting of a set $E(G)$ of edges and a set $N(G)$ of nodes. Each node is a special point of interest that needs to be observed in search of intruders, but it is assumed that such observation is instantaneous. Each edge represents a path by using a number corresponding to the cost proportional to its length.

Thus, given such graph and a set of robots, the patrolling task consists of visiting at each time step as many nodes as possible in order to minimize the time lag between two visits at the same node. Therefore, each node not only is an environment point of interest to be inspected, but also a point where interaction among agents arises, i.e., each robot in graph node $n \in N(G)$ must select, based on other robots selections, an appropriate action in order to choose the next node to visit. Taking into account this interaction, normal-form games at each graph node $n \in N(G)$ have been defined.

**Definition 1 (Normal-Form Game)** Formally, a finite n-robot normal-form game $\Gamma$ is made of:

- A finite set $M$ of *robots* $i = 1,\ldots,n$.
- A finite set $A = A_1 \times \cdots \times A_n$, where $A_i = \left\{a_i^1,\ldots,a_i^k\right\}$ is a finite set of actions for *robot* $i = 1,\ldots,n$. Each vector $a = \left\{\left\{a_1^{j_1},\ldots,a_n^{j_n}\right\} \in A \middle| j_{1,\ldots,n} \in \left\{1,\ldots,k\right\}\right\}$ is called *action profile* for the game $\Gamma$. Each action is related to an edge $e \in E(G)$.
- A finite set $S = S_1 \times \cdots \times S_n$, where $S_i = \left\{s_i^1,\ldots,s_i^k\right\}$ is a finite set of strategies for *robot* $i = 1,\ldots,n$. Each vector $s = \left\{\left\{s_1^{j_1},\ldots,s_n^{j_n}\right\} \in S \middle| j_{1,\ldots,n} \in \left\{1,\ldots,k\right\}\right\}$ is called *strategy profile* for the game $\Gamma$. A strategy is the criterion taken into account to determine the action to be selected.
- A payoff function $\pi_i(s) \mapsto \Re$ for *robot* $i = 1,\ldots,n$, where $S$ is the set of strategy profiles.
- Therefore, $\pi_i(s)$ is the payoff of robot $i$ when strategy profile $s$ is chosen.

In this implementation, each node $n \in N(G)$ has a fixed number of normal-form games which depends on the edges connected to it. At time step zero, the robots are randomly positioned on the set $N(G)$.

Thus, at every time step, the robot $i \in M$ reaches a node $n \in N(G)$ and plays its corresponding normal-form game $\Gamma$. As a consequence, the robot chooses its individual strategy $s_i^{j_i} \in S_i$ considering the strategies selected by all other robots. The action related to the strategy chosen leads the robot to the next node. Finally, the interaction among robots arises when each robot sends a message indicating the strategy selected. A robot can select an action with probability one or by randomizing over the set of available actions according to some probability distribution. Such strategies are called *pure* and *mixed*, respectively.

**Definition 2 (Pure Strategy)** Given a set of available actions $A_i = \{a_i^1, \ldots, a_i^k\}$ for robot $i = 1, \ldots, n$, a pure strategy $s_i^{l_i} = \{a_1^{j_1} \mid j_i \in \{1, \ldots, k\}\}, l_i \in \{1, \ldots, k\}$. Thus, given a game $\Gamma$, a *pure strategy profile* $s = \{\{s_1^{j_1}, \ldots, s_n^{j_n}\} \mid j_{1,\ldots,n} \in \{1, \ldots, k\}\}$ for the game is the joint strategies selection where $s_n^{j_n}$ is the choice of strategy $j_n$ by robot $n \in M$

**Definition 3 (Mixed Strategy)** A mixed strategy $\sigma_i$ for robot $i = 1, \ldots, n$ is a probability distribution over the set of available actions $A_i = \{a_i^1, \ldots, a_i^k\}$ for such robot; i.e., a *mixed strategy* has the form $\sigma_i = (p_i^1 a_i^1 + \cdots + p_i^k a_i^k)$ where $p_i^{j_i} \geq \{0 \mid j_i = 1, \ldots, k\}$ and $\sum_{j_i=1}^{k} p_i^{j_i} = 1$. Thus, robot $i \in M$ selects $a_i^{j_i}$ with probability $p_i^{j_i}$. If $p_i^{j_i} = 0, \forall j = 1, \ldots, k$ except for one, say $p_i^{l_i} = 1, l_i \in \{1, \ldots, k\}$, then $\sigma$ is a *pure strategy*. Thus, given a game $\Gamma$, a *mixed strategy profile* $\sigma = (\sigma_1, \ldots, \sigma_n)$ for the game is the joint selection of strategies where $\sigma_i$ is the strategy selected by robot $i \in M$.

A manner to represent a game $\Gamma$ is through the use of a *n × n matrix payoff game* in which each cell of this matrix indicates a strategy profile and contains the outcome or *mixed strategy payoffs* of the game when such profile is selected. Such outcome take the form $C(x, y) = (\pi_1, \ldots, \pi_n), x, y = 1, \ldots, n$ where the entry $\pi_i$ is the payoff of robot $i \in M$

**Definition 4 (Expected payoff of a mixed strategy)** Given a normal-form game $\Gamma$, the expected payoff for robot $i \in M$ playing the strategy profile for the game $s = (s_1^{j_1}, \ldots, s_n^{j_n}), j_{1,\ldots,n} \in \{1, \ldots, k\}$ is defined as

$$\pi_i(s) = \left\{ \sum_{a \in A} \pi_i(a) \cdot \prod_{i=1}^{n} \Pr(a_i^{j_i}) \mid j_i \in \{1, \ldots, k\} \right\} \quad (1)$$

where $\Pr(a_i^{j_i})$ is the probability that action $j_i$ will be played by robot $i \in M$.

Thus, the robots that interact in these types of games choose an action that maximize its expected payoff considering the actions selected by all other robots. This is called *best response* and it leads to the central solution concept of game theory, the *Nash equilibrium*. From now on, the robots other than robot $i \in M$ are specified by $-i = \{(1, \ldots, i-1, i+1, \ldots, n) \in M\}$. Moreover, the strategy profile without the strategy of robot $i \in M$ is defined by

$$s_{-1} = \left\{ (s_1^{j_1}, \ldots, s_{i-1}^{j_{i-1}}, s_{i+1}^{j_{i+1}}, \ldots, s_n^{j_n}) \in S_i \mid j_{1,\ldots,i-1,i+1,\ldots,n} \in \{1, \ldots, k\} \right\}.$$

**Definition 5 (Best Response)** The best response of robot $i \in M$ to the strategy profile $s_{-1}$ is a strategy $s_i^* \in S_i$ such that $\pi_i(s_i^*, s_{-i}) \geq \pi_i(s_i^{j_i}, s_{-i}), j_i \in \{1, \ldots, k\}$ for all strategies $s_i^{j_i} \in S_i$.

**Definition 6 (Nash equilibrium)** A strategy profile $s = \{(s_1^{j_1}, \ldots, s_n^{j_n}) \mid j_{1,\ldots,n} \in \{1, \ldots, k\}\}$ for the game $\Gamma$ is a Nash Equilibrium if, for all robots $i \in M, s_i^{j_i}$ is a best response to $s_{-i}$.

In the games played in this implementation, the robots do not have conflicting interests and their sole challenge is to coordinate on actions that are maximally beneficial to all. These types of games are called *team games*.

Finally, in order to visit as many nodes of a graph as possible, a set of robots must disperse throughout the environment. Based on this requirement, the payoffs are defined as follow: let $\tau_i(s_{-i}^j)$ be the times that robots other than robot $i \in M$ select the strategy $j \in \{1, \ldots, k\}$, where $s_{-i}^j = \{(s_1^j, \ldots, s_{i-1}^j, s_{i+1}^j, \ldots, s_n^j)\}$. Thus, the payoff for robot $i \in M$ playing such strategy is defined as $\pi_i(s_i^j, s_{-i}^j) = |M| - \tau_i(s_{-i}^j)$. Therefore, the payoff is low when several robots choose the same strategy. Henceforth, at each time step that a robot $i \in M$ reaches a node $n \in N(G)$, plays a normal-form game $\Gamma$ against $-i \in M$, chooses an available action of its set of actions $A_i$ and goes to the next node $n \in N(G)$.

## 4. Experience-weighted attraction learning model

In the *Experience-Weighted Attraction* or *EWA* learning model each strategy $s_i^{j_i} \in S_i$ has a numerical value called *attraction*, which specifies the probability of choosing that strategy. Each attraction has an initial value, which is updated each period through the use of two rules that update two variables. The first variable $\Lambda_i^{j_i}(t)$, correspond to the level of attraction of the robot $i \in M$ by the strategy $s_i^{j_i} \in S_i$ after period t. The second variable $\psi(t)$ represents the amount of experience and indicates the number of observation-equivalents of past experience. These variables begin with some prior values, $\psi(0)$ and $\Lambda_i^{j_i}(0)$.

Thus, the first rule updates the level of attraction according to

$$\Lambda_i^{j_i}(t) = \frac{\phi \cdot \psi(t-1) \cdot \Lambda_i^{j_i}(t-1) + \left[\delta + (1-\delta) \cdot \xi(S_i, s_i^{j_i}(t))\right] \cdot \pi(s_i^{j_i}(t), s_{-i})}{\psi(t)} \quad (2)$$

The decay rate $\phi$ depreciates previous attraction $\Lambda_i^{j_i}(t-1)$ and represents a combination of forgetting and level of knowledge about the adaptation of other robots. When $\phi$

is lower, old attractions are decayed more quickly, whereas the most recent attractions are more important. Moreover, the parameter $\delta$ weights hypothetical payoffs that unchosen strategies would have earned, whereas $(1-\delta)$ weights payoffs received from the chosen strategy $s_i^{j_i}(t)$. Finally, the indicator function $\xi\left(S_i, s_i^{j_i}(t)\right)$ is equal to 1 when the strategy selected $s_i^{j_i}(t)$ at period t is equal to some strategy of the set of strategies $S_i$. This indicator function is utilized in all the models of this section.

The second rule updates the amount of experience according to

$$\psi(t) = (1-\nu)\cdot\phi\cdot\psi(t-1) + 1, t \geq 1 \quad (3)$$

where $(1-\nu)\cdot\phi$ is the rate of decay for experience, which measures the impact of previous experience. Moreover, the parameter $\nu$ specifies how quickly robot $i \in M$ lock in to a strategy. When $\nu = 0$, attractions are weighted averages of past attractions and payoffs. Past attractions are weighted by $\frac{\phi\cdot\psi(t-1)+1}{\phi\cdot\psi(t-1)+1}$, whereas payoffs are weighted by $\frac{1}{\phi\cdot\psi(t-1)+1}$. When $\nu = 1$ the attractions cumulate because $\psi(t)$ becomes one.

The initial attraction level of each strategy $s_i^{j_i} \in S_i$ is denoted by $\Lambda_i^{j_i}(0) = \frac{1}{|S_i|}, \forall j_i = 1,\ldots,k$, while the initial value for the number of *observation-equivalents* $\psi(0) = 1$.

Thus, an EWA robot $i \in M$ using these rules shapes a set of attractions which specify preference for a specific strategy $s_i^{j_i} \in S_i$. Such preference is given as a choice probability $P_i^{j_i}$ in time step $t+1$ through a logistic stochastic response function defined by

$$P_i^{j_i}(t+1) = \frac{e^{\lambda\cdot\Lambda_i^{j_i}(t)}}{\sum_{j_i=1}^{k} e^{\lambda\cdot\Lambda_i^{j_i}(t)}} \quad (4)$$

where $\lambda$ is the response sensitivity. With $\lambda = 0$ the choice is stochastic while $\lambda = \infty$ is best response.

Beyond these rules, specific values of $\psi(0), \delta$ and $\rho$ reduce this general model to special cases such as *reinforcement* and *belief-based* models.

### 4.1 Reinforcement Model

In the *reinforcement* model of EWA, every time step that a robot $i \in M$ reaches a graph node $n \in N(G)$, it performs three steps. In the first step, the robot selects one of the strategies available at such node. This selection is based on a logistic stochastic response function defined by

$$P_i^{j_i}(t+1) = \frac{e^{\lambda\cdot R_i^{j_i}(t)}}{\sum_{j_i=1}^{k} e^{\lambda\cdot R_i^{j_i}(t)}} \quad (5)$$

where $\lambda$ is the response sensitivity. With $\lambda = 0$ the choice is stochastic, whereas $\lambda = \infty$ is best response. The response function $P_i^{j_i}$ specifies the selection probability of the strategy $s_i^{j_i}$

Each strategy $s_i^{j_i}$ is related to a reinforcement value $R_i^{j_i}$. When this reinforcement value is updated, its related strategy is reinforced. Thus, in the second step, once a strategy is selected, only this strategy is reinforced by previous received payoffs according to

$$R_i^{j_i}(t) = \phi\cdot R_i^{j_i}(t-1) + \xi\left(S_i, s_i^{j_i}(t)\right)\cdot\pi_i\left(s_i^{j_i}(t), s_{-i}\right) \quad (6)$$

As can be seen, this rule is the result of setting in the EWA model $\delta = 0, \nu = 1$, and $\psi(0) = 1$, therefore $\psi(t) = 1$. This is the simplest form of cumulative reinforcement. When $\delta = 0, \nu = 0$ and $\psi(0) = \frac{1}{(1-\phi)}$, *EWA* model becomes a form of averaged reinforcement governed by

$$R_i^{j_i}(t) = \phi\cdot R_i^{j_i}(t-1) + (1-\phi)\cdot\xi\left(S_i, s_i^{j_i}(t)\right)\cdot\pi_i\left(s_i^{j_i}(t), s_{-i}\right) \quad (7)$$

where reinforcements are averages of previous attractions and incremental reinforcement. The initial reinforcement value of the strategies available at each node is defined by $R_i^{j_i}(0) = \frac{1}{|S_i|}, \forall j_i = 1,\ldots,k$.

Finally, in the last step, the robot $i \in M$ communicates the strategy selected to the other robots, so that they update the reinforcement value of the strategy selected by the robot $i \in M$ in the node $n \in N(G)$. Thus, similar to the behavior of attractions in EWA model, in the *reinforcement* case, each robot shapes the reinforcement of each strategy by utilizing the aforementioned rules. The Algorithm 1 describe the three steps accomplished in this model.

**Data**: M, N
**forall** $n \in N(G)$ **do**
  **forall** $j \in 1,\ldots,k$ **do**
    $R^j(0) \leftarrow \frac{1}{|S|}$
  **end**
**end**
**while** *true* **do**
  (1) Select a strategy based on $P^j$;
  (2) Reinforce the strategy selected through $R^j$;
  (3) Communicate the strategy selected to the other robots $-\, i \in M$;
**end**

**Algorithm 1**. Algorithm of the reinforcement model

*4.2 Belief-Based Model*

Belief-based models start with the premise that each robot $i \in M$ identifies that it is playing a game $\Gamma$ with other robots, and it forms beliefs about what these robots will play in the future based on its past observation. Then, it attempts to define dynamic processes that lead to a *Nash Equilibrium* by choosing a best response strategy that maximizes its expected payoff to its beliefs.

There are different iterative learning rules to form beliefs. One widely used model of learning is the process of *Weighted Fictitious Play* and its variants, such as *Cournot Best-Response Dynamics*, which looks back only one play, as opposed to *Fictitious Play* which looks back the t most recent plays, (Brown, 1951). At each time step in the model of *Weighted Fictitious Play*, each robot $i \in M$ chooses its strategies to maximize its expected payoff given its prediction about the distribution over strategies of other robots at that time step. Therefore, *Weighted Fictitious Play* is an instance of model-based learning in which a robot maintains beliefs $B_i^t(s_{-i})$ about the strategies of other robots $s_{-i} \in S_{-i}$. In the prediction of this learning rule, the initial prior belief that robot $i \in M$ assigns to strategies $s_{-i}$ of robots $-i \in M$ is governed by

$$B_i^0(s_{-i}) = \frac{Y_i^0(s_{-i})}{\sum_{\tilde{s}_{-i} \in S_{-i}} \left[ Y_i^0(\tilde{s}_{-i}) \right]} \tag{8}$$

where $Y_i^0(s_{-i}) : S_{-i} \to \Re_+$ is an exogenous initial weight function, which assigns a real value to each strategy of the robots other than robot $i \in M$. This value is assigned according to $Y_i^0(s_{-i}) : \frac{|S_{-i}| - (j_i - 1)}{|S_{-i}|}, \forall j_i = 1, \ldots, k,$ where $|S_{-i}|$ represents the cardinality of the set $S_{-i}$. Later, every time step that the strategy $s_{-i}$ is played, its weight is updated according to

$$Y_i^t(s_{-i}) = \phi \cdot Y_i^{t-1}(s_{-i}) + \begin{cases} 1 & \text{if } s_{-i}^{t-1} = s_{-i} \\ 0 & \text{if } s_{-i}^{t-1} \neq s_{-i} \end{cases} \tag{9}$$

The initial weight assigned is different for each strategy. This assignation permits that the updates performed by 9 do not lead to weights with the same value, which allows to avoid selection problems.

The belief that robot $i \in M$ assigns to the robots $-i \in M$ playing $s_{-i}$ at time step t is given by

$$B_i^t(s_{-i}) = \frac{Y_i^t(s_{-i})}{\sum_{\tilde{s}_{-i} \in S_{-i}} \left[ Y_i^t(\tilde{s}_{-i}) \right]} \tag{10}$$

The updating rule formulated by 10 can be defined in terms of previous-period beliefs by

$$B_i^t(s_{-i}) = \frac{\phi \cdot \sum_{\tilde{s}_{-i} \in S_{-i}} \left[ Y_i^t(s_{-i}) \right] \cdot B_i^{t-1}(s_{-i}) + \begin{cases} 1 & \text{if } s_{-i}^{t-1} = s_{-i} \\ 0 & \text{if } s_{-i}^{t-1} \neq s_{-i} \end{cases}}{\phi \cdot \sum_{\tilde{s}_{-i} \in S_{-i}} \left[ Y_i^t(\tilde{s}_{-i}) + 1 \right]} \tag{11}$$

In 11, $B_i^t(s_{-i})$ is expressed in terms of previous-period beliefs, similar to EWA model with previous-period attractions and reinforcement model with previous-period reinforcements.

Following this updating rule, most recent and previous-period beliefs are updated $\phi$ times. When $\phi = 0$ *Weighted Fictitious Play* becomes *Cournot Best-Response Dynamics*, and when $\phi = 1$ it becomes *Fictitious Play*. Once beliefs are updated, expected payoff of robot $i \in M$ in period t, $E_i^t(s_i^{j_i})$ is defined according to

$$E_i^t(s_i^{j_i}) = \sum_{\tilde{s}_{-i} \in S_{-i}} \pi(s_i^{j_i}(t), \tilde{s}_{-i}) \cdot B_i^t(s_{-i}) \tag{12}$$

As in the case of beliefs, expected payoffs can be expressed as a function of previous-period expected payoffs which yields

$$E_i^t(s_i^{j_i}) = \frac{\phi \cdot \sum_{\tilde{s}_{-i} \in S_{-i}} \left[ Y_i^t(\tilde{s}_{-i}) \right] \cdot E_i^{t-1}(s_i^{j_i}) + \pi(s_i^{j_i}(t), s_{-i})}{\phi \cdot \sum_{\tilde{s}_{-i} \in S_{-i}} \left[ Y_i^{t-1}(\tilde{s}_{-i}) \right] + 1} \tag{13}$$

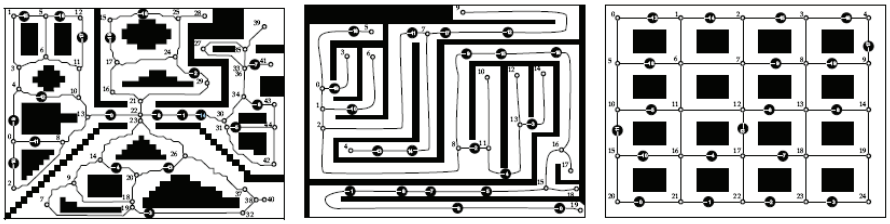
Finally, the best response of the robot $i \in M$ in *Weighted Fictitious Play* is given by

$$BR_i^t = \arg\max_{j_i} E_i^t(s_i^{j_i}) \tag{14}$$

## 5. Experiments and result

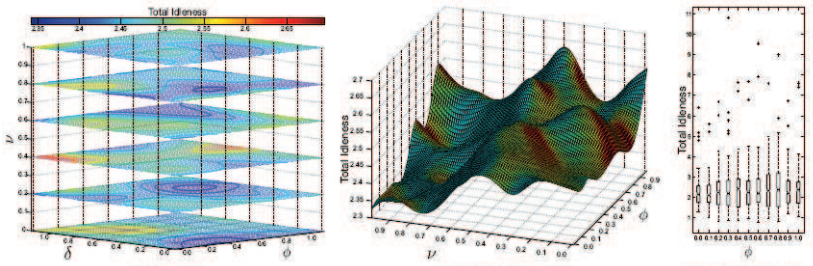
In order to evaluate and compare this implementation with other methods, a patrolling simulator developed from pioneers works (Machado et al., 2003) has been used.

Thus, the first experiments aim at analyzing the behavior of these models with different values of their parameters, namely, for *EWA* model $\phi, \delta$ and $\nu$, for reinforcement model $\phi$ and $\nu$, and for belief-based model $\phi$. In order to do so, the map shown in figure 1(a) was used. Where unfilled small circles stand for nodes or points of interest, lines stand for edges of a graph or paths that robots use to move throughout the map. Filled big circles stand for robots patrolling such map. In this set of experiments, a group of 20 robots started at node number 22 and patrol until each node had been visited 256 times.



(a) Strongly Connected Map (b) Maze Map (c) Grid Map

**Figure 1.** Maps used in order to evaluate performance of each algorithm




(a) *EWA* (b) Reinforcement (c) Belief-based

**Figure 2.** Performance of the models evaluated utilizing the map of figure 1(a).

Figure 2 shows the performance of the models evaluated. At each plot, color intensity, coordinates, and box plots represent the total idleness. Thus, figure 2(a) shows the behavior of *EWA* model using six slice planes at the axis $\nu$ through a volumetric data created with values of $0 \le \nu, \phi, \delta \le 1.0$. As it can be seen, the best behavior was obtained with the slice at $\nu = 0.2$. In this slice, the volumetric data shows that the idleness is, in general, approximately 2.5. The region around the points $\delta = 0.7$ and $\phi = 0.9$ presents the lowest idleness, whereas the idleness value hardly ever reaches values of 2.55. Figure 2(b) shows a surface expressing the behavior of the reinforcement model for values of $0 \le \nu, \phi, \delta \le 1.0$. It shows that the bigger the value of $\nu$ and the lower the value of $\phi$, the better the performance of the model. Indeed, the surface has a minimum at coordinate $(\nu = 0.9, \phi = 0.3)$. In general, $\nu = 0.9$ achieves the best behavior, which indicates that the form of cumulative reinforcement performs better than the average reinforcement. In the form of cumulative reinforcement, each strategy has a level of reinforcement, which is incremented cumulatively by received payoffs. Finally, figure 2(c) shows the performance of the belief-based model for values of $0.0 \le \phi \le 1.0$ by means of using box plots. The size of each box represents the spread of the data, whereas the symbols "+", called outliers, are values distant from the rest of the data. For security purpose, it is not suitable to have several outliers because they represent nodes with high idleness, which indicates points of weakness. In general, the nature of the belief-based model is almost the same in all cases. When $\phi = 0.1$ the median, 2.29, of the data is the smallest. Even though, the smallest median does not mean better performance, the size of the box when $\phi = 0.1$ is lower than the other options. This size indicates that the total idleness of all nodes is similar. Moreover, there are only two outliers, at nodes 28 and 40 with total idleness of 5.22 and 5.61, respectively. Although, there are options with two or less outliers, their values are bigger. Based on this information, values of $\nu = 0.2, \phi = 0.9$ and $\delta = 0.7$ for *EWA* model; $\nu = 0.9$ and $\phi = 0.3$ for reinforcement model, and $\phi = 0.1$ for belief-model model were selected.

Once all parameters of the models were determined, the next experiments aimed at evaluating the models in comparison to other methods available in the Patrolling Simulator, namely, the *Cycle* algorithm and the *MPS* algorithm, which were described in section 2. For these experiments, all the maps of figure 1 were used. For each map, sets *M* of 6, 8, 10, 15, 20, 25, 30 robots were evaluated. This combination generates 21 cases of study to experiment. Each experiment consists of visiting all nodes 256 times. At time $t = 0$, each robot was placed randomly at one node of the map.

Table 1 shows the results of these experiments for *Strongly Connected*, *Maze* and *Grid* maps. In these results, the lower the size of $\mu$, the better the performance of the algorithm and the lower the value of $\sigma$, the lower the variance. As it can be observed, if the value of M increases, the performance of all of the algorithms is better. Although this behavior seems obvious, if the coordination among robots is not appropriate, bigger values of M will not lead to better performance of the algorithms. Therefore, this nature is due to the suitable coordination.

| Map | M | Cycle | | MPS | | Belief-Based | | Reinforcement | | EWA | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Total Idleness | | | | | | | | | |
| | | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ |
| Strongly Connected | 30 | 1.711 | 0.829 | | | 1.663 | 0.649 | **1.647** | 0.614 | 1.674 | 0.673 |
| | 25 | 2.006 | 0.974 | | | **1.904** | 0.655 | 2.086 | 1.109 | 1.979 | 0.876 |
| | 20 | 2.471 | 1.195 | | | **2.445** | 1.151 | 2.460 | 1.173 | 2.453 | 1.101 |
| | 15 | 2.981 | 3.805 | | | 3.066 | 2.672 | **2.372** | 2.923 | 3.598 | 4.981 |
| | 10 | 5.119 | 6.534 | | | 3.941 | 5.090 | **3.148** | 2.640 | 5.587 | 5.165 |
| | 8 | **4.225** | 3.662 | 5.889 | 5.287 | 4.912 | 4.095 | 4.443 | 4.714 | 5.130 | 4.714 |
| | 6 | 8.139 | 10.81 | 6.783 | 6.689 | 8.474 | 9.503 | 8.078 | 10.47 | **6.746** | 6.477 |
| Maze | 30 | 3.304 | 1.604 | | | **3.114** | 1.490 | 3.240 | 1.720 | 3.264 | 1.639 |
| | 25 | 3.904 | 1.895 | | | **3.605** | 1.760 | 3.884 | 2.055 | 3.940 | 1.996 |
| | 20 | 4.877 | 2.361 | | | **4.477** | 2.251 | 4.689 | 2.446 | 4.931 | 2.489 |
| | 15 | 4.844 | 4.101 | | | **4.259** | 4.270 | 7.153 | 7.168 | 6.896 | 7.019 |
| | 10 | 7.674 | 7.737 | | | 8.203 | 7.404 | 6.865 | 5.571 | **6.008** | 9.869 |
| | 8 | 9.963 | 8.524 | | | **5.909** | 6.165 | 9.437 | 9.746 | 9.926 | 10.24 |
| | 6 | 13.58 | 11.56 | | | **10.30** | 7.604 | 19.72 | 19.53 | 18.64 | 27.91 |
| Grid | 30 | 0.921 | 0.415 | | | **0.881** | 0.155 | 0.919 | 0.274 | 0.900 | 0.230 |
| | 25 | 1.064 | 0.478 | | | 1.034 | 0.286 | 1.016 | 0.231 | **1.000** | 0.234 |
| | 20 | 1.319 | 0.594 | | | **1.218** | 0.252 | 1.235 | 0.299 | 1.226 | 0.290 |
| | 15 | 1.279 | 3.114 | | | 1.649 | 1.335 | 1.892 | 1.312 | **1.242** | 0.961 |
| | 10 | 2.080 | 3.326 | | | 2.027 | 1.967 | **1.753** | 1.705 | 1.989 | 1.682 |
| | 8 | 2.514 | 1.826 | 2.110 | 1.936 | 2.020 | 2.087 | 2.604 | 1.743 | **2.015** | 1.998 |
| | 6 | 2.883 | 5.238 | 2.947 | 2.513 | **1.843** | 1.471 | 3.517 | 2.479 | 2.852 | 2.397 |

**Table 1.** Performance of the algorithms with a different size of M in each map

Apart from having a more suitable architecture for security purposes, in as many as 95% of cases one of the methods presented in this work improves *Cycle* algorithm. The only case where *Cycle* algorithm performed better was in *Strongly Connected* map with $M = 8$. Most notably, these results indicate that regardless of which map is used, in all cases, at least one of the methods presented in this work improves *MSP* algorithm. Taking into account that both *Cycle* and *MPS* algorithms use a centralized and explicit coordinator scheme, this improvement in performance is significant. Finally, in 80% of cases *MPS* algorithm does not work due to partitioning problems. Portugal and Rocha (2010) describe the reasons of these problems. It is worth noting that the proposed solution does not have these problems.

## 6. Conclusions

Several dynamic and distributed collaborative multi-robot approaches for security applications at critical facilities have been developed. Thus, a team of robots endowed with patrolling behaviors based on learning models from game theory as well as a thorough study of such models in the context of the patrolling problem has been presented. As shown in section 5, a significant improvement in performance was obtained by using the proposed methods with respect to *Cycle* and *MPS* algorithms. Moreover, the distributed characteristics of these models offer solutions with several advantages, such as scalability, modularity, and incremental expandability. Furthermore, the behavior of the robots patrolling that are using the techniques of this work is non-deterministic, which is suitable for security applications due to the fact that intelligent intruders can learn patrolling paths, and based on this information, perform attacks to the protected system. The evaluation to support this claim is not part of the scope of this work. However, results in (Sak and Wainer, 2008) demonstrate that system protection based on not static solutions is less susceptible to be attacked.

Despite the good performance achieved with the models implemented, there are significant remaining questions for future research. Firstly, interference among robots arises when more than one robot utilizes the same edge. In order to avoid this interference, it is necessary to evaluate if the edge selected is used or not by other robots. Future research consisting of studying the behavior of these methods including such aspect is necessary. Secondly, the metric of the patrolling simulator to evaluate the performance of the algorithms only includes the idleness of each node, however it does not take into account if one edge connected to such node (in the case that it has more than one) is used or not. Thus, an interesting future research consisting of evaluating the behavior of the algorithms including such a restriction because it allows to have a more secure system. Finally, even though the expected payoff matrix defined has achieved suitable results, new definition matrices should be explored.

## 7. Acknowledgements

This work has been supported by the Robotics and Cybernetics Research Group at Technique University of Madrid (Spain), and funded under the projects "ROTOS: Multi-Robot system for outdoor infrastructures protection", sponsored by Spain Ministry of Education and Science (DPI2010-17998), and the project ROBOCITY 2030 Project, sponsored by the Community of Madrid (S-0505/DPI/000235).

## 8. References

[1] Abate, F. (1996). The Oxford Dictionary and Thesaurus: The Ultimate Language Reference for American Readers, *Oxford University Press*.

[2] Aguirre, O., Llausas, R., Lucero, C., Taboada, H., Espiritu, J. and Kiekintveld, C. (2011). A multi-objective evolutionary algorithm for intelligent patrolling, *In Proceedings of the 41st International Conference on Computers & Industrial Engineering*.

[3] Ahmadi, M. and Stone, P. (2006). A multi-robot system for continuous area sweeping tasks, *In Proceedings of the IEEE International Conference on Robotics and Automation*.

[4] Almeida, A., Ramalho, G., Santana, H., Tedesco, P., Menezes, T., Corruble, V. and Chevaleyre, Y. (2004). Recent advances on multi-agent patrolling, *Advances in Artificial Intelligence,SBIA*, Springer.

[5] An, B., Kempe, D., Kiekintveld, C. and Shieh, E. (2012). Security Games with Limited Surveillance, *AAAI 2012*.

[6] Brown, G. (1951). Iterative solution of games by fictitious play, *Activity analysis of Production and allocation*.

[7] Camerer, C. (1999). Experience Weighted Attraction Learning in Normal Form Games, *Econometrica*.

[8] Chevaleyre, Y. (2004). Theoretical analysis of the multi-agent patrolling problem, *International Conference on Intelligent Agent Technology*, IEEE Computer Society.

[9] Elmaliach, Y., Agmon, N. and Kaminka, G. (2007). Multi-robot area patrol under frequency constraints, *Annals of Mathematics and Artificial Intelligence* .

[10] Everett, H. (2003). Robotic security systems, *Instrumentation & Measurement Magazine, IEEE*.

[11] Farinelli, A., Iocchi, L. and Nardi, D. (2004). Multi-robot systems: a classification focused on coordination, *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*.

[12] Fudenberg, D. (1998). *The theory of learning in games*, The MIT Press.

[13] Glad, A. and Buffet, O. (2009). Self-organization of patrolling-ant algorithms, *Proceedings of the 2009 Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, IEEE Computer Society.

[14] Hwang, K. (2009). Cooperative patrol planning of multi-robot systems by a competitive auction system, *ICCAS-SICE*.

[15] Machado, A., Ramalho, G., Zucker, J. and Drogoul, A. (2003). Multi-agent patrolling: An empirical analysis of alternative architectures, *Multi-Agent-Based Simulation II*.

[16] Oates, R., Milford, M. and Wyeth, G. (2009). The implementation of a novel, bio inspired, robotic security system, *Intelligent Robots and Systems (IROS)*.

[17] Portugal, D. and Rocha, R. (2010). Msp algorithm: multi-robot patrolling based on territory allocation using balanced graph partitioning, *Proceedings of the 2010 ACM Symposium on Applied Computing*, ACM.

[18] Portugal, D. and Rocha, R. (2011). A survey on multi-robot patrolling algorithms, *Technological Innovation for Sustainability*.

[19] Sak, T. and Wainer, J. (2008). Probabilistic multi-agent patrolling, *Advances in Artificial Intelligence-SBIA*.

[20] Santana, H., Ramalho, G., Corruble, V. and Ratitch, B. (2004). Multi-agent patrolling with reinforcement learning, *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-agent Systems-Volume 3*, IEEE Computer Society.

[21] Sempé, F. and Drogoul, A. (2003). Adaptive patrol for a group of robots, *Intelligent Robots and Systems (IROS)*.