A Multiagent Planning Approach for Cooperative Patrolling with Non-Stationary Adversaries

Aurélie Beynier

A Multiagent Planning Approach for Cooperative Patrolling with Non-Stationary Adversaries

Aurélie Beynier

Sorbonne Universités, UPMC Univ Paris 06
CNRS, UMR 7606, LIP6
F-75005, Paris, France

aurelie.beynier@lip6.fr

Multiagent patrolling is the problem faced by a set of agents that have to visit a set of sites to prevent or detect some threats or illegal actions. Although it is commonly assumed that patrollers share a common objective, the issue of cooperation between the patrollers has received little attention. Over the last years, the focus has been put on patrolling strategies to prevent a one-shot attack from an adversary. This adversary is usually assumed to be fully rational and to have full observability of the system. Most approaches are then based on game theory and consists in computing a best response strategy. Nonetheless, when patrolling frontiers, detecting illegal fishing or poaching; patrollers face multiple adversaries with limited observability and rationality. Moreover, adversaries can perform multiple illegal actions over time and space and may change their strategies as time passes. In this paper, we propose a multiagent planning approach that enables effective cooperation between a team of patrollers in uncertain environments. Patrolling agents are assumed to have partial observability of the system. Our approach allows the patrollers to learn a generic and stochastic model of the adversaries based on the history of observations. A wide variety of adversaries can thus be considered with strategies ranging from random behaviors to fully rational and informed behaviors. We show that the multiagent planning problem can be formalized by a non-stationary DEC-POMDP. In order to deal with the non-stationary, we introduce the notion of context. We then describe an evolutionary algorithm to compute patrolling strategies on-line, and we propose methods to improve the patrollers' performance.

Keywords: Multiagent patrolling; Markov Decision Processes; Distributed decision making; Planning under uncertainty.

1. Introduction

Multiagent patrolling is the problem faced by a set of agents that have to periodically visit a set of sites. In adversarial domains, the agents have to prevent some threats or illegal actions performed by a set of adversaries. A large amount of recent works have been dedicated to Security Games where a single adversary tries to perform a one-shot attack on one of the sites to patrol [1]. Wider settings have also been investigated [2,3,4,5]. They consider several defenders that have to face multiple adversaries performing frequently and repeatedly illegal actions. These works have been applied to domains such as preventing crime in urban areas [3], avoiding intrusions on frontiers [2], or detecting illegal fishing or poaching [4].

When considering several patrolling agents repeatedly facing multiple adversaries, it is natural to expect that the agents will be able to coordinate their individual strategies to maximize adversary detection. Shieh et al. [6] have shown that, in the context of a one-shot attack, defender effective teamwork significantly improves security. Although there has been an increasing focus on handling multiple adversaries and frequent illegal actions [5,4,7], the issue of effective cooperation between the patrollers have received little attention.

In this paper[a], we propose a new framework allowing several patrolling agents to effectively cooperate in order to face several defenders performing multiple illegal actions over time and space. Patrollers will have to secure a set of sites dispatched over the environment. At each decision step, each patroller will thus have to decide which site to visit next. Obviously, in order to compute valuable strategies for real settings, agents should be able to handle uncertainty on action outcomes and partial observability of the system. For instance, when fighting illegal fishing, cost guards' moves between two different spots rely on weather conditions and are thus uncertain. Furthermore, cost guards have limited observability and cannot observe all fishing spots at any time of their patrol. In this paper, we describe a multiagent planning approach computing cooperative patrolling strategies under uncertainty. Given her strategy, an agent will thus be able to make individual but cooperative decisions in partially observable and uncertain environments.

In the context of multiagent patrolling with multiple adversaries, the patrollers' objective consists in detecting as much illegal actions as possible. A model of the adversaries should be used to anticipate possible threats at each time-step. It is commonly assumed in the literature that adversaries are fully rational and fully observe the patrolling strategy. Tools from Game Theory can then be used to compute the best response strategy of the adversaries and to deduce the optimal strategy of the defenders. In fact, assuming full observability and rationality may not reflect reality and the optimal patrolling strategy (in theory) may not be so efficient in practice. Indeed, adversaries are often unable to fully observe the patrolling strategy because of their limited observation capacities. Moreover, observing the patrolling strategy may be risky (the adversary may be detected) and costly (it takes time and consumes resources the adversary may not have) [9,4]. In addition, adversaries may not be fully rational since they have limited reasoning capacities or they may not comply to the fully rational strategy (for instance, humans may deviate from the fully rational strategy).

Since adversaries have limited observability and bounded rationality, their strategy may evolve over time as they obtain more knowledge about the patrolling strategies or about the environment. As the adversarial behavior will thus be non-stationary, the patrollers will have to detect these changes and adapt their strategies. For instance, illegal fishermen can change their fishing spots based on their accumulated knowledge about cost guards' patrols.

[a]This paper is an extended version of [8] presented in IEEE ICTAI 2016.

Our framework proposes to consider a generic setting where several patrolling agents face multiple adversaries with non-stationary strategies. No specific assumption is made about the rationality nor the profile of the adversary. Instead, patrollers will learn a model of the adversary as they obtain more and more information and adapt their behaviors to the non-stationarity of the adversaries.

Our work contributes to the domain in several directions:

- A new formalization of the defenders' decision problem is proposed to allow for effective cooperation while facing multiple adversaries performing repeated illegal actions over time and space.
- Patrolling strategies handles uncertainty on action outcomes and partial observability of the environment.
- A generic model of the adversaries is learnt and updated as the agents make new observations about the adversaries. The multiagent planning approach copes with non-stationary behaviors of the adversaries.
- A distributed planning algorithm is proposed to compute online patrolling strategies and update them to the non-stationarity of the adversaries. A mathematical method is described to better identify changes in the adversaries strategy.

Section 2 presents an overview of related works. Section 3 formalizes the multiagent patrolling problem with multiple adversaries. Section 4 introduces the necessary background on Markovian decision models and shows how the problem can be formalized as a Decentralized Partially Observable Markov Decision Process (DEC-POMDP). Section 5 presents an evolutionary algorithm to compute patrolling strategies. Methods are proposed to update the model of the adversaries and adapt patrolling strategies to possible changes in adversary strategies. Finally, Section 6 describes experimental results about the efficiency of our approach.

2. State of the art

Earliest multiagent patrolling approaches have focused on computing strategies that minimize the time lag (called "idleness") between two visits of a same target [10]. These works compute deterministic strategies that consist in a sequence of targets. Chevaleyre [10] has shown that this problem is closely related to the Traveling Salesman Problem (TSP) and cyclic strategies computed using TSP solvers give good results in non-adversarial domains. However, such strategies fail to anticipate possible adversary strategies and consider action outcomes to be deterministic.

More recently, some research works have been interested with patrolling in adversarial settings. Such settings consider patrolling agents that have to prevent attacks from an intruder (i.e. the adversary). Different settings have been investigated, leading to different kinds of solutions. Most approaches assume a strong adversary performing extensive surveillance of the patrollers and then conducting a one-shot attack [11,12,13,14,9]. The adversary is thus able to obtain full knowledge

of the patrolling strategy. Developed approaches then formalize the problem as a leader-follower decision problem as described in Game Theory. Such settings, usually refered to as "security games", are well suited to develop frameworks to prevent punctual threats such as terrorism attacks. However, in many domains, adversaries only have limited observability [15] and considering a strong adversary is not realistic nor optimal. Agmon et al. [2,16] have studied the impact of adversarial knowledge on the patrolling strategies in the specific setting of perimeter patrols. They demonstrated that if the adversary is not a strong one and has no knowledge about the patrol scheme, an optimal deterministic strategy exists. Computing mixed strategies is thus not required.

When patrolling frontiers or fighting against illegal fishing or poaching, defenders certainly have to face multiple adversaries [4]. The defenders then try to maximize the number of detected illegal actions instead of preventing a one-shot attack. Furthermore, while considering such real systems, the full observability assumption does not hold. Instead, adversaries have partially and noisy observations about the patrollers strategies. From the defenders point of view, computing a best response strategy to strong adversaries may not be an optimal approach. Qian et al. [17] relaxed the assumption of a one-shot attack following a prior extensive surveillance. A single protector has to prevent several illegal actions performed by a single adversary. Both agents (the patroller and the adversary) are assumed to fully observe the actions of the other one.

Despite the wide interest put in multiagent patrolling over the last years, little attention has been paid to the issue of cooperation between multiple patrollers. In fact, most existing works consider only one patrolling agent or assume that the same strategy is executed by all patrollers [16]. Munoz de Cote et al. [18] introduced alarms to provide some information about intruders' presence and improve the efficiency of the patroller. Recently, Shieh et al. [6] combined security games and Decentralized MDPs to enable effective cooperation between several patrollers under uncertainty. A single fully rational adversary is considered. This adversary is assumed to perform a prior extensive surveillance phase and to attack the target with the lowest coverage. Nguyen et al. [19] tackled the issue of multiple adversaries performing frequent and repeated illegal actions. The problem is formalized as a repeated game where defense resources have to be deployed on targets at each turn. While this work accounts for learning adversarial strategies, it does not consider effective cooperation between defenders nor uncertainty on action outcomes.

3. Multiagent patrolling setting

We consider a set of m heterogeneous defenders (agents i with $i \in [1, m]$) patrolling a set of n (with $m \ll n$) target sites t_j (with $j \in [1, n]$) to detect illegal actions. Note that if $m \geq n$, one patroller can be allocated to each target. However, we consider the more difficult case where $m \ll n$ and more elaborated strategies must be computed to schedule patrolling resources among the targets to protect. The

environment topology is represented as a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ where $\mathcal{N} = \{t_1, \cdots, t_n\}$ is the set of targets and $\mathcal{E}$ denotes the set of possible routes between the targets.

Figure 1 illustrates a multi-agent patrolling problem where 3 patrollers (boats with a red flag) have to patrol 6 target sites. Two adversaries (blue fishing boats) are fishing illegally on two different target sites. The adversary at the bottom right of the figure is detected since a patroller is simultaneously on the same target.

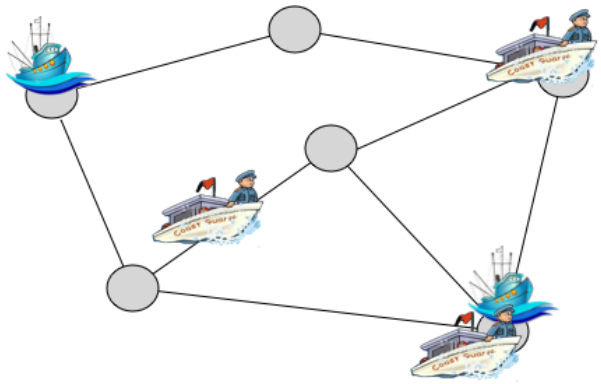

Fig. 1. Illustrative example

Because of the dynamicity of the environment and of the partial observability of the agents, it should be considered that action outcomes are stochastic. In fact, uncertainty may arise from the agent itself or from external events. For example, human patrollers may deviate from their recommended patrolling strategy, or the actuators of a robot may be imperfect leading to deviations from the expected behavior. External events can also lead to uncertain action durations or outcomes. In order to guarantee robust solutions, it is essential to take into account the uncertainty on action execution while computing patrolling strategies [20].

Each edge $e = (t_k, t_j) \in \mathcal{E}$ of the graph is thus assigned a probability distribution $C_{k,j}$ on possible travel durations. Moving from one target t_k to another target t_j can therefore takes different amounts of time.

Like previous works dealing with patrolling in adversarial domains, performing an illegal action is assumed to take time[2]. Here, each illegal action lasts Δ_{int} time step.

As explained above, we consider partial observability of the patrollers and of the intruders. Each patroller is assumed to observe her own location and illegal actions on the target she is currently patrolling. Adversaries performing illegal actions on another target are not observed. We do not make any assumption on the observability model of the adversaries.

Note that illegal actions can be performed several times on a same target and several illegal actions can be performed on different targets at the same time. Our approach is thus generic and does not make any assumption on the number ad-

versaries. Furthermore, the number of adversaries is unknown and may evolve over time. We do not make any assumption on the full rationality of the adversaries nor on their possible cooperation. Instead, defenders will try to anticipate the adversaries strategy from their observations.

4. The cooperative multiagent patrolling decision problem

Our approach addresses the problem of computing patrolling strategies for the defenders in order to maximize their performance. Cooperative strategies will be computed allowing for effective coordination of the patrollers while executing their action in uncertain and partially observable environments.

Patrolling systems are inherently distributed: at each decision step, each patrolling agent must decide in an autonomous but cooperative way, which target to visit next in order to maximize the global performance of the defenders. Each agent makes her decisions given her local knowledge about the adversaries and about the environment (graph topology and uncertainty model).

In this paper, we show that this distributed and cooperative decision problem can be formalized as a Decentralized Partially Observable Markov Decision Process (DEC-POMDP)[21]. In fact, DEC-POMDPs describe a mathematical framework for modeling and solving sequential multiagent decision problems under uncertainty where a set of cooperative agents have to decide, in a distributed way, how to act given partial observations about the system.

Because of limited observability and bounded rationality, we argue that it is not realistic to assume that the patrollers are able to build a full model of the adversaries (and thus to include it in their model of the environment). In fact, patrolling agents cannot observe all the actions of the adversaries over the whole environment. Patrollers then make decisions based on limited knowledge about the adversaries behavior. Moreover, adversaries may have bounded rationality and not always commit to an optimal policy. They can also keep on adapting their strategy from their past observations about the patrollers. In such settings, assuming a strong rational adversary and anticipating the best response of the adversaries is no more optimal for the patrollers. In this paper, we introduce the notion of *context* formalizing the knowledge of the patrollers about the adversary policy. Patrolling strategies will then be computed based on the current context. In order to cope with the non-stationarity of the adversary strategies, our framework allows patrolling agents to be able to detect policy changes. The current context is then updated and new patrolling strategies are computed on-line (ie. during the patrol).

4.1. *Background on DEC-POMDPs*

As demonstrated by the wide range of recent works dealing with Decentralized Partially Observable Markov Decision Problems (DEC-POMDPs), this mathematical model is especially suited for formalizing cooperative distributed decision-making problems under uncertainty.

A DEC-POMDP [21] is defined as a tuple $\langle \mathcal{A}g, S, A, T, O, \Omega, R \rangle$ where:

- $\mathcal{A}g = \{1, ..., m\}$ is a set of m agents,
- S is the set of world states s,
- $A = \{A_1 \times \cdots \times A_m\}$ is the set of possible joint actions $a = \{a_1, ..., a_m\}$ such as a_i is the action of agent i,
- T is the transition function giving the probability $T(s'|s, a)$ that the system moves to state s' while executing the joint action a from state s,
- $O = \{O_1 \times \cdots \times O_m\}$ is the set of joint observations $o = \{o_1, ..., o_m\}$ where o_i is the individual observation of agent i,
- Ω is the observation function giving the probability $\Omega(o|s, a)$ of observing o when executing the joint action a from state s,
- $R(s'|s, a)$ is the reward obtained when executing the joint action a from state s and moving to state s'.

Optimally solving a DEC-POMDP consists in finding a joint policy $\pi = \{\pi_1, ..., \pi_m\}$ that maximizes the common performance measure of the agents where π_i is the individual cooperative policy of agent i. An individual policy for an agent i maps each possible history of observations of i to an individual action a_i. It has been proved that optimally solving a DEC-POMDP is NEXP-Complete [21].

Because of the high complexity of DEC-POMDPs, many works have focused on developing efficient solving methods. Mechanisms for improving the efficiency of the optimal solving have been investigated such as: dynamic programming [22], multiagent A* [23] or formalization of the problem as a deterministic MDP with continuous states [24]. Despite major advances for improving of the optimal algorithms, the scalability of optimal approaches remains quite limited [24]. In fact, approximate approaches often better scale to large number of agents and long planning horizon [25,26,27].

4.2. *Formalization of the decision problem as a DEC-POMDP*

The following observations support the fact that DEC-POMDPs can formalize cooperative patrolling problems : 1) agents have to make distributed sequential decisions in uncertain and partial observable environments, 2) agents are cooperative since they want to maximize a common measure of performance (ie. the number of detected illegal actions). Nonetheless, patrollers face adversaries with non-stationary strategies. The transition and reward functions of the patrollers thus change over time. Moreover, patrollers may have little information on the adversaries. In our settings, the only observed information about the adversary strategies consists in detected illegal actions.

Non-stationary model of the adversaries

We propose a generic approach to model the knowledge of the patrollers about the adversaries. This approach only exploits the history of observations made by the

patrollers and is independent of the number of adversaries, the rationality of the adversaries or their observability.

The knowledge about the adversaries is thus formalized, for each target t_i by a probability $PI_i.PI_i(t)$ stands for the probability that the adversaries initiate an intrusion on target t_i at step t. The probability that none adversary initiates an intrusion on target t_i at t is then given by $1 - PI_i(t)$. These probabilities PI_i can be refined over time as patrollers get more and more observations about the adversaries. Moreover, variations of PI over time reflect changes in adversaries strategies (non-stationarity). As explained below, PI values can be used to define the current transition and reward functions of the patrolling decision-problem.

DEC-POMDP model for a current context

Following previous works dealing with non-stationary mono-agent POMDPs [28], we decompose the non-stationary decision problem as a series of stationary decision problems. Each stationary phase is then referred to as a *mode* or a *context*. In the patrolling problem, a mode (or a context) refers to a profile of the adversaries strategy. In this paper, we exploit the periods of stability and the variations of PI probabilities to define stationary contexts and transitions between these contexts. For each stationary context, the multiagent patrolling problem is formalized as a DEC-POMDP. As probabilities PI evolve, the current context changes over time. For each new context, a new DEC-POMDP formalization of the decision problem is defined and patrolling strategies are updated consequently.

We now describe how to formalize the patrolling decision problem as a DEC-POMDP for a stationary context. We detail the definition of each component of the tuple $\langle \mathcal{A}g, S, A, T, O, \Omega, R \rangle$.

Agents ($\mathcal{A}g$): the set of agents involved in the multiagent decision problem consists of all the patrolling agents.

Actions (A): Each agent has to make decisions about the next target to patrol. An individual action a_i for an agent i thus consists in *moving to target* t_j ($t_j \in \mathcal{N}$) connected to the current location of agent i (t_j must be directly connected to the current target in the graph). We consider the more realistic setting where moving from a target to another may take different durations. At each time step, some agents will have to make new decisions while others will keep on executing their current moves. Agents may not all make decisions at each time step. Since DEC-POMDPs consider one time unit action duration, individual moves are decomposed into a set of consecutive unitary actions. In fact, if moving from a target t_k to a target t_j takes $c_{kj} \in C_{k,j}$ time units, the move is decomposed into c_{kj} successive unitary moves. Note that even if agents have probabilistic knowledge on the possible duration of an action, the effective duration of a move is actually known only once it has been fully completed. It is assumed that an agent does not change her mind during the execution of a move and always stays focused on the target to reach.

States (S): Let s_t denote the state of the system at time t. It is defined as: the position of each agent, the list of targets where an illegal action has been currently observed, the idleness of each target, the elapsed time of each current move. A state s_t is thus defined as a tuple $\langle p = \langle p_1, ..., p_m \rangle, int, idle = \langle idle_1, ..., idle_n \rangle, \delta = \langle \delta_1, \cdots, \delta_m \rangle \rangle$ where:

• The position p_i of each agent i is defined as a target or an edge of the graph, i.e. $p_i \in \{\mathcal{N} \cup \mathcal{E}\}$. The tuple of positions of the agents is denoted by p with $p = \langle p_1, ..., p_m \rangle$.

• For each target currently patrolled (t_i such as $t_i \in p$), the state indicates whether an illegal action is currently observed on this target. The state thus contains the list int of targets where illegal actions have been observed at t.

• *Idleness* refers to the time elapsed since the last visit of a target [10]. Each target is assigned an idleness value thus leading to the tuple $idle$ with $idle = \langle idle_1, ..., idle_n \rangle$.

• δ_i denotes the time elapsed since each patrolling agent has left her last visited target, leading to the tuple $\delta = \langle \delta_1, ..., \delta_m \rangle$. The highest possible travel time between two targets gives an upper bound on possible values for δ_i.

To avoid overloading equations, we will denote this state by $s_t = \langle p, int, idle, \delta \rangle$.

Observations (O): The individual observation of an agent i first consists in observing her current location (an edge or a target). If the agent has reached a target, she observes whether an illegal action is currently performed on that target. When an agent is moving from a target to another, she is assumed not to observe adversaries (we limit illegal actions to be performed on the nodes of the graph).

Transition function (T): The probability to move from one state to another while executing a joint action relies on : 1) probabilities on move durations and 2) probabilities on detection of illegal actions. As explained before, we define a DEC-POMDP for a given stationary context. We thus consider probabilities PI related to the current context and derive transition probabilities.

At each decision step, some agents are reaching new targets to visit whereas other agents are still moving between two targets. Move durations are independent of the other agents. However, idleness values rely on all the agents' actions and positions. From a state s_t, if an agent i reaches a new target t_j, the system moves to a state $s'_{t'}$ where the idleness of t_j is 0, $\delta_i = 0$ and $p_i = t_j$. Otherwise (i does not reach her next target), δ_i is incremented by 1, p_i corresponds to the current edge of the agent and the agent does not reset any idleness value (although other agents could change these values). The probability that an agent reaches her target is defined from probability distributions $C_{k,j}$.

The current context PI is used to compute probabilities on the detection of illegal actions. When the agent i reaches a target t_j, she may observe an illegal action on this target. The probability w_j of observing an illegal action on t_j at t is in fact the probability that an adversary initiated such an action within the last Δ_{int} time steps and it has not been detected yet. w_j is then defined as:

$$w_j(t) = \mathbb{P}(\bigcup_{w=0}^{min(\Delta_{int}, idle_j)} \mathcal{I}_j(t-x))$$

where $\mathcal{I}_j(t-x)$ denotes the event "an illegal action is initiated at $t-x$ on t_j". Using the inclusion - exclusion principle applied to probabilities, $w_j(t)$ can be rewritten as a sum of probabilities on conjunctions of events $\mathcal{I}_j$. The probability of such an event is then given by PI_j.

$$w_j(t) = \sum_{k=1}^{n}((-1)^{k-1}\sum_{i \leq i_1 < i_2 < ... < i_k \leq n} \mathbb{P}(I_j(t-1) \cap I_j(t-2) \cap ... \cap I_j(t-k)))$$

Observation function (Ω): In this paper, we assume deterministic observations. In fact, when an agent observes her current target, illegal actions that may be performed are always detected. There is no noise nor uncertainty on observations.

Reward function (R): The reward function formalizes the objectives of the patrolling agents. In a DEC-POMDP, the agents try to maximize their expected discounted sum of rewards. In order to maximize the number of detected illegal actions, a reward must be given for each detected adversary. We assume that only one agent is required to detect an adversary. A reward is perceived only once when several agents detect the same adversary at the same time. The reward obtained when executing action a from a state $s_t = \langle p, int, idle, \delta \rangle$ and moving to $s'_t = \langle p', int', idle', \delta' \rangle$ is defined as:

$$R(s'_t|a, s_t) = \sum_{t_i \in int'} R^D(t_i) + \sum_{t_i \in p' \text{ and } \notin int'} \cdot R^P(t_i, idle_i) \quad (1)$$

where $R^D(t_i) \in \mathbb{R}^*_+$ denotes the reward for detecting an illegal action on the target t_i and $R^P(t_i, idle_i) \in \mathbb{R}^*_+$ is the reward for patrolling target t_i without detecting any illegal action. In order to guarantee patrolling all targets, $R^P(t_i, idle_i)$ is proportional to the idleness of the target before being patrolled. This part of the reward encourages agents to patrol targets with a high idleness. It prevents patrolling behaviors where the agents focus on a restricted set of targets.

Targets can be rewarded with different values formalizing the relative significance of the targets. One can imagine to identify sensitive locations (areas populated with endangered species for instance) and attached highest rewards to these target sites.

Related models could be considered to formalize our decision problem but they do not fulfill all the requirements of our settings. Interactive POMDPs (I-POMDPs) [29] include a model of the other agents in the belief state of each agent. However, I-POMDPs assume a fixed set of adversarial models known beforehand. Stackelberg

Security Games [30] consider a known model of the adversary and do not account for effective cooperation during action execution.

4.3. *From observations to the current context definition*

The DEC-POMDP formalization described above is given for a fixed current context represented by probabilities PI. These probabilities formalize the current knowledge of the patrollers about the adversaries strategy. As patrolling agents execute their actions, they make more and more observations and obtain more and more information about the profile of the adversaries.

Since patrolling agents only observe detected illegal actions, they cannot have prefect knowledge about the profile of the adversaries. However, PI probabilities can be estimated from the number of detected illegal actions over the last $\mathcal{H}$ time steps. Let $NI_i(t-\mathcal{H},t)$ be the number of detected adversaries on target t_i (defined for all t_i in $\mathcal{N}$) between $t-\mathcal{H}$ and t. We estimate the probability PI on target t_i at time t as:

$$PI_i(t) = \frac{NI_i(t-\mathcal{H},t)}{\sum_{t_k \in \mathcal{N}} NI_k(t-\mathcal{H},t)} \tag{2}$$

This definition may appear as a rough estimate but it guarantees the relation: $NI_i(t-\mathcal{H},t) > NI_k(t-\mathcal{H},t) \Rightarrow PI_i(t) > PI_k(t)$. In addition, this estimate is consistent with our limited observability assumption and fits nicely with the objective of patrolling agents to detect as many illegal actions as possible. As described in the experiments (see Section 6), this estimation allows for high detection ration. However, our DEC-POMDP definition is not restricted to this formalization of the current context. One can imagine a more sophisticated model of the adversaries provided that the agents have a higher degree of observability or obtain external knowledge about the adversaries.

Updating the DEC-POMDP model and the patrolling strategies at each time step from the new current context has an important computational cost and may lead to poor performance. We thus introduce the notion of *context horizon* (denoted by $\mathcal{T}$) that sets the period of validity of the current context. Once a context is defined at t by probabilities $PI(t)$, it is assumed to be valid for the next $\mathcal{T}$ time steps. A DEC-POMDP model is then also defined at t and strategies are computed for the next $\mathcal{T}$ time steps. Note that $\mathcal{T}$ is the period of validity of a context and the planning horizon of the DEC-POMDP whereas $\mathcal{H}$ sets the length of the observation history used to define PI probabilities related to the current context.

5. Computation of non-stationary patrolling strategies

Optimally solving a DEC-POMDP consists in computing a joint policy that maximizes the global expected reward derived from Equation 1. Since the reward function of a DEC-POMDP is defined over joint actions, individual optimal policies maximize the global reward of the agents and are thus cooperative policies. Note that

agents usually end with different individual policies. In the context of multiagent patrolling, the nodes of the graph will be dispatched among the agents taking into account uncertainty on moves between the target, target rewards and threat levels. Agents will tend to spread in the graph in order to provide a high coverage of the targets.

Various approaches have been proposed to solve DEC-POMDPs [24,31,32]. They can be applied to our DEC-POMDP model to compute a joint cooperative patrolling strategy. In the DEC-POMDP literature, it is commonly assumed that planning can be performed off-line (before the execution) in a centralized way: a joint policy is first computed by a central entity that then sends to each agent her individual policy. Individual policies are finally executed in a distributed way: each agent is able to make her own coordinated decisions from her local observations.

In the context of multiagent patrolling with non-stationary adversaries, the use of existing algorithms present some difficulties. In fact, each time a new context is considered, a new joint policy must be computed. If a centralized algorithm is used, a central entity would have to collect all the observations made by the patrolling agents to deduce the new context, update the DEC-POMDP model and compute a new strategy. This strategy will then be communicated to the patrolling agents. Such an approach obviously creates a bottleneck in the system and would result in high communication cost. Furthermore, DEC-POMDP algorithms use to compute the joint policy *from scratch*. They have not been designed to update joint strategies during the execution. They cannot re-use previously computed strategies to speed up the computation of a new joint strategy. In non-stationary environments, it would be useful to update on-line the strategies to the changes of dynamics.

In this paper, we propose to use a distributed evolutionary algorithm to compute patrolling strategies. Evolutionary algorithms have been used previously to compute approximate solutions for DEC-POMDPs [33,34] and showed significance improvement in the size of the horizon that can be handled. Evolutionary algorithms thus open promising directions to solve large problems. Moreover, our evolutionary algorithm allows for exploiting strategies of previous contexts when computing a new strategy for a new context. In order to improve the agents' efficiency we also investigate the relevance of communicating. Finally, since the policies of the adversaries may evolve over time, we propose a new procedure to detect policy variations.

5.1. *Evolutionary Algorithm for policy computation*

Based on the (1+1) evolutionary algorithm [35], we propose a method for planning the patrolling strategy over an horizon $\mathcal{T}$. In this evolutionary context, an individual is defined as a joint policy over the horizon $\mathcal{T}$. The algorithm (see Algorithm 1) starts from an initial solution (called `champion`) and then iterates to improve the `champion` until a computation deadline is reached. At each iteration, a mutation

is performed on the current `champion` to obtain a new solution referred to as the `challenger`. The challenger is evaluated and compared with the current `champion`. The highest rewarding solution is kept and becomes the new current `champion`.

Algorithm 1 (1+1) evolutionary algorithm

```
champion = RamdomIndividual()
championValue = Evaluate(champion)
while deadline non reached do
  challenger = Mutation(champion)
  challengerValue = Evaluate(challenger)
  if challengerValue > championValue then
    champion = challenger
    championValue = challengerValue
  end if
end while
```

The set of all possible individuals consists in the set of joint policies that comply with temporal and spatial constraints of the problem. Spatial constraints formalize existing paths between two targets in the graph $\mathcal{G}$. A policy of an agent i fulfills spatial constraints if the agent always decides to move to a target directly connected to her current target. Temporal constraints arise from uncertainty on action durations: an agent cannot decide for a new target to visit until she has reached her current target (ie. an agent cannot change her direction while moving on an edge of the graph).

The initial solution is defined considering the likelihood of an illegal action on each site. These probabilities can be estimated using probabilities PI defining the current context. In fact, the higher the probability of an illegal action on a target t_k, the higher the probability of selecting t_k.

In our setting, we define the mutation operator such as it increases the visit frequency of the weakest targets and decreases the visit frequency of the most visited targets: move to targets with low probabilities of threats are replaced by moves to targets with higher probabilities of threats.

Our evolutionary algorithm has the advantage of being anytime. Moreover, thanks to its low complexity, it scales well to large numbers of agents and long planning horizon $\mathcal{T}$. However, no guarantee on the quality of the solution can be given. In fact, the algorithm does not provide any bound regarding the distance to the optimal solution and can be trapped in local optima. This issue will be discussed in the experimental section.

5.2. *Communication models*

In order to guarantee coherent individual patrolling strategies, the evolutionary algorithm must be executed in a centralized way or in a distributed way with each

agent having the same PI distribution as her teammates. Centralized execution is possible using a central entity to whom each agent notifies detected intrusions. This central entity thus maintains a coherent estimation of PI probabilities and executes the evolutionary algorithm. Strategies can then be broadcasted to the patrolling agents.

However, using such a central entity introduces a weak point in the multiagent system that could be exploited by the adversaries. One solution consists in broadcasting only useful information to all the agents. Each time an agent detects an intrusion, she notifies all her teammates. Such communication can be encrypted to ensure more security. Each agent can then execute the evolutionary algorithm and computes coherent cooperative joint policies. However, such an approach may lead to a large number of messages.

We propose to limit communication between the agents by measuring the relevance of the communicated information. If an information is considered as relevant, it is communicated. Otherwise, it is not sent for the moment. We propose to measure the relevance of an information as the distance between the current probability distribution PI and the new probability distribution PI' obtained by exploiting the information to communicate. This distance is computed by the Kullback-Leibler divergence [36] which evaluates the difference between two probability distributions P and Q.

The divergence from P to Q is thus defined as:

$$\sum_i P(i) log \frac{P(i)}{Q(i)}$$

In order to preserve the symmetry property, we use the pseudo-distance:

$$D(P,Q) = \sum_i P(i) log \frac{P(i)}{Q(i)} + \sum_i Q(i) log \frac{Q(i)}{P(i)}$$

We then use a small β parameter value as a threshold value. If the Kullback-Leibler distance is greater than β , the information is considered as being relevant and it is communicated to all teammates. Otherwise, the information is not communicated but may be sent later if, combined with new observations, it becomes significant. We have also investigated communication models based on the Bhattacharyya distance [37] or the Hellinger distance [38]. We did not notice any significant differences in the values obtained to measure the relevance of an information.

5.3. *On-line detection of context changes*

As described in Section 4.3, the current context is updated every every $\mathcal{T}$ time steps considering the observations about detected illegal actions over the last $\mathcal{H}$ time steps. However, adversaries may change their strategy during the $\mathcal{T}$ time steps of a context. Patrollers should be able to detect such changes and adapt their strategies consequently instead of waiting for the end of lifetime of the current context.

We describe an approach allowing the agents to detect adversary policy changes during the lifetime of a context (the $\mathcal{T}$ time steps of the context). We define a mathematical method monitoring the variations of the number of detected adversaries *det* over the $\mathcal{H}$ last time steps considered in Equation 2. In fact, empirical studies on the variations of *det* showed that this quantity significantly decreases when adversaries change their strategy. Our purpose is thus to efficiently detect such decreases to update the context as soon as possible after the adversaries changed their policy.

Our method consists of four successive processing operations:

(1) Compute a moving average $det_t(\mathcal{H})$ of the number of detected illegal actions over the last $\mathcal{H}$ time steps for each time step t.

$$det_t(\mathcal{H}) = \frac{1}{|\mathcal{H}|} \sum_{t_i \in \mathcal{N}} NI_i(t - \mathcal{H}, t)$$

This average allows for smoothing variations due to the stochasticity of the system.

(2) Decompose det_t values using a finite adaptation of Stieltjes decomposition (see Appendix for more details). $det_t(\mathcal{H})$ is decomposed into two functions $det_t^-(\mathcal{H})$ and $det_t^+(\mathcal{H})$ such as $det_t^-(\mathcal{H})$ corresponds to the negative variation of $det_t(\mathcal{H})$ whereas $det_t^+(\mathcal{H})$ corresponds to the positive variation of $det_t(\mathcal{H})$. The decreasing components det^- of *det* can thus be identified.

(3) Apply a backward finite difference operator to $det_t^-(\mathcal{H})$:

$$\nabla_{det^-}[t] = det_t^-(\mathcal{H}) - det_{t-1}^-(\mathcal{H})$$

to quantify decreasing variations.

(4) Threshold the values obtained in the previous step to detect adversarial policy changes. If a variation exceeds the threshold, it is assumed that the adversaries have changed their strategy.

As soon as an adversarial policy change is detected, the current context PI is updated even if the deadline $\mathcal{T}$ has not been reached. New policies are then computed based on the new context. Note that this method can be applied irrespectively of the solving algorithm.

The threshold of the procedure is a parameter of the method and has to be tuned considering the DEC-POMDP formalization of the problem. Low threshold values provide sensitive detection but could lead to "false" detection of strategy changes. On the other hand, high thresholds might miss some strategy changes.

If no change has occurred over the horizon $\mathcal{T}$, the context is updated at the end of the horizon and new policies are computed for the new context over the next $\mathcal{T}$ time steps.

It should be noted that our approach allows patrolling agents to adapt their strategies online. Patrolling strategies are thus non-stationary. From the point of view of a strong adversary (ie. adversary with full observability of the patrollers

and performing an extensive surveillance phase), the patrolling strategy will thus not seem deterministic all along the execution.

6. Experimental results

Our approach has been experimented on different sizes of randomly generated problems. Graphs were randomly build considering various numbers of nodes. Edges were defined by randomly picking two different nodes in $\mathcal{N}$. Each node is connected in average to 2/3 of the other nodes. Action durations were randomly drawn in the interval $[1, 5]$. Illegal actions were assumed to last over 10 time steps.

Initial strategies of the adversaries were also randomly defined assuming that intrusion probabilities belong to the interval $[0.1, 0.5]$ for a subset of the targets and are 0 elsewhere. Note that these strategies are initially unknown to the patrollers.

For each simulation, the system was executed over at least 400 time steps and the adversaries changed their policies at least once during the execution. Experiments were performed on a computer equipped with an Intel(R) Core(TM)2 Duo processor, 2000 MHz, 8Gb.

6.1. *Performances and scalability*

First experiments dealt with the performance of the patrollers. As the agents aim at maximizing the number of detected illegal actions, we studied the detection ratio (number_of_detected_illegal_actions / number_of_illegal_actions) along the patrolling mission. Results obtained by our evolutionary algorithm have been compared with the optimal solution computed using the MADP toolbox [39]. Figure 2 gives the detection ration for small scenarios (2 agents and 5, 6 or 7 targets) executed over 530 time steps. It can be observed that our DEC-POMDP based approach leads to high detection ratio: over 70% for both algorithms and over 76% for the optimal algorithm. Although the evolutionary algorithm is not guarantee to find an optimal solution, performances are closed to the optimal. In fact, the detection ratio is decreased by only 3% over the optimal solution for 5 targets and by 6% for 7 targets. It has to be noticed that even the optimal approach cannot lead to full detection of illegal actions since agents cannot cover all targets within 10 time steps (duration of an illegal action).

Figure 3 gives the reward obtained by the agents with the optimal and the evolutionary algorithm. It can be observed that the evolutionary algorithm is also closed to the optimal solution.

Number of targets: The scalability of the approach has then been studied. We increased the number of targets and tested the performances of the solutions. Problems over 2 agents and 7 targets could not be solved optimally. Nonetheless, our evolutionary algorithm successfully solved problems up to 50 targets and 7 agents (with a deadline of 10 seconds). Note that larger problems could be solved by enlarging the deadline of the evolutionary algorithm.

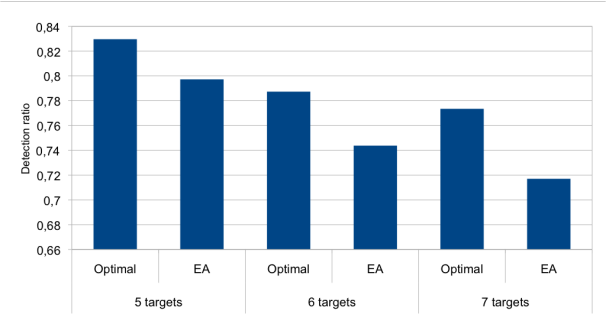

Fig. 2. Detection ratios of the executed strategies

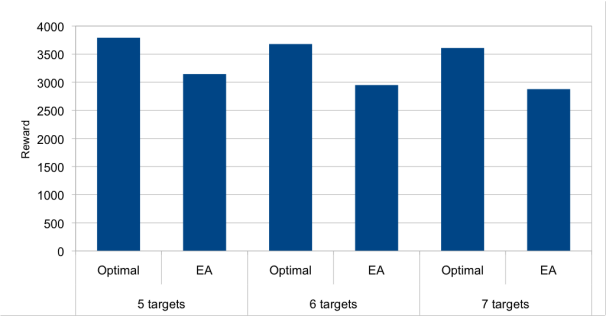

Fig. 3. Reward obatined by the patrollers

Moreover, for a fixed size of problems higher performances could be obtained by enlarging the deadline. In fact, our evolutionary algorithm is anytime and performances increase as more time is given to the algorithm.

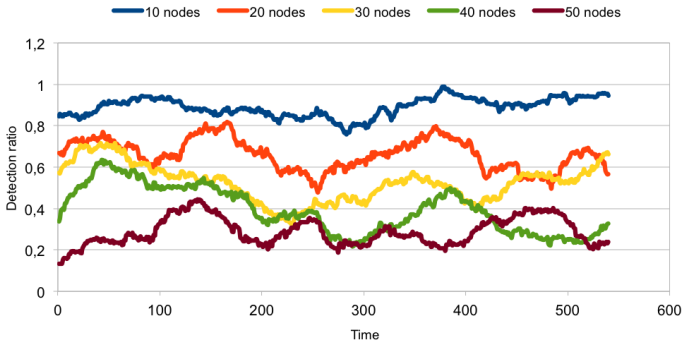

Fig. 4. Influence of the number of targets

Figure 4 gives the variations of the detection ratio over 530 time steps for differ-

ent sizes of graphs. The number of agents was fixed to 4 agents for all sizes of graphs. It can be observed that large numbers of targets lead to lower detection ratios. In fact, as the number of targets increases, it becomes more and more difficult for the 4 agents to cover all the targets and to obtain high detection ratios. For a fixed number of agents, these experiments show that the performances are closely related to the extent of the area to patrol. The extent of the area relates to the number of targets to patrol and to move durations between two targets. Figure 4 shows that good performances are obtained for graphs smaller or equal to 16 targets. Indeed, detection ratios of 80% and more were obtained for 4 agents by the evolutionary algorithm.

Number of agents: We then tested the influence of the number of agents m on the performances of the approach. Although all targets cannot be covered at each time step and moving from one target to another takes time (without the ability to make detection), good detection ratios are obtained even if $m \ll n$. Figure 5 gives the detection ratio obtained on a 16-target graph when the number of agents varies. When $m = n$, the detection ratio obviously equals to 1: each agent is assigned a single node of the graph and stays on this node. Nonetheless, it can be observed that full detection of illegal actions is almost obtained for 12 agents and more. Even considering 6 agents leads to high detection ratios (more than 0.9).

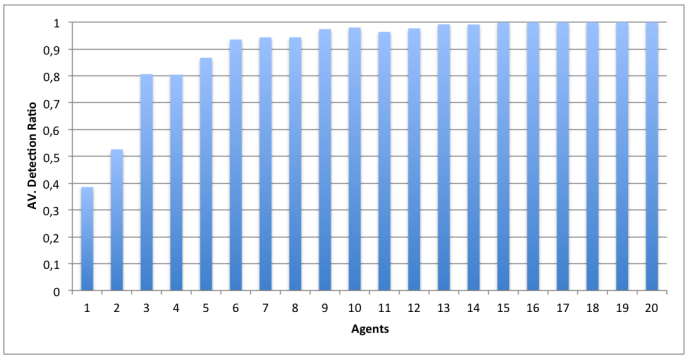

Fig. 5. Influence of the number of agents

Deadline and planning horizon of the evolutionary algorithm: The deadline of the evolutionary algorithm and the planning horizon both influence solution quality. We considered different values of these parameters and recorded the influence on the average detection ratio. Figure 6 gives the detection ratio for problems involving 5 agents and 16 targets. It can be seen that deadlines of the evolutionary algorithm over 1 second do not significantly improve solutions. In fact, good quality solutions are already obtained with shorter deadlines. Furthermore, increasing the

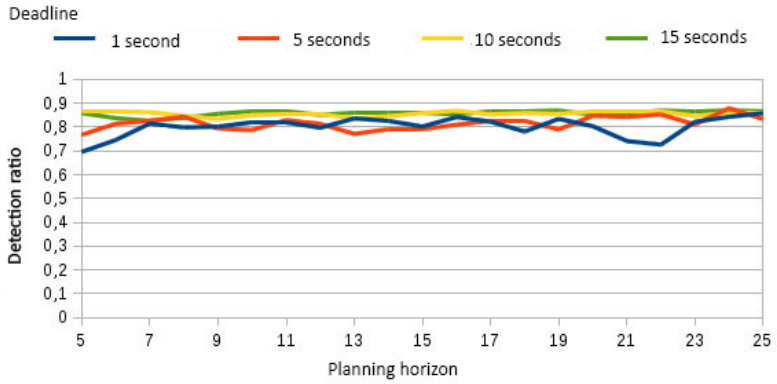

Fig. 6. Influence of the deadline and of the planning horizon

planning horizon $\mathcal{T}$ over 10 time steps does not improve the detection ratio. Setting the planning horizon results from a trade-off between the number of steps looked ahead in strategy computation and the frequency of updates. Longer planning horizon decreases the frequency of strategy updates (policy computation is done every $\mathcal{T}$ steps). On the other hand, shorter planning horizons increases the frequency of context updates and policy updates. More variations of the detection ratio are then observed since agents become more and more myopic and are unable to anticipate future action outcomes and opportunities.

Figure 7 illustrates how the value of the champion evolves as time passes (until the deadline of 10 seconds is reached). The value of the champion is monotonically increasing since our algorithm is anytime. Nevertheless, later time steps lead to less improvement since it becomes more difficult to find a variation that improves the value of the champion (more iterations do not change the current champion).

6.2. *Performances with limited communication*

We have then tested the relevance of limiting the number of messages using Kullback-Leibler divergence. Table 8 gives the detection ratio when limiting communication. Figure 9 describes the number of messages exchanged between the agents. Note that a logarithmic scale is used in Figure 9. We considered different number of agents and a graph of 16 targets. We varied the β threshold used to decide whether a message must be sent or not ($\beta \in \{5, 10, 15\}$). $KL05$ stands for the approach based on the Kullback-Leibler divergence with a threshold set to 5. The "Com" approach consists in always broadcasting observations about detected illegal actions.

As shown in Figure 9, our approach based on Kullback-Leibler divergence allows the agents to reduce significantly the number of messages sent. In the 5-agents case, the number of messages has been reduced from 574 to 16. The β value can be used as a parameter to tune the frequency of communication. Our approach could

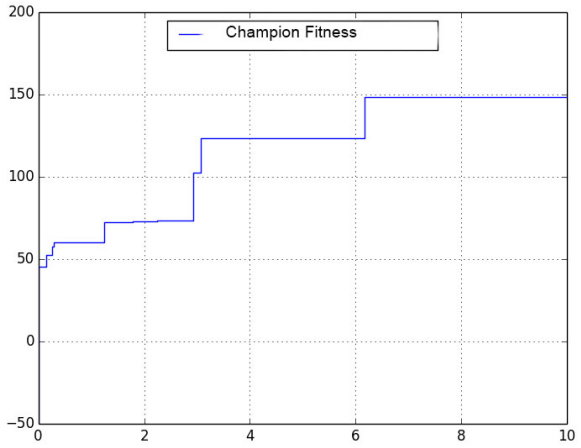

Fig. 7. Best champion among the execution of the EA

also handle dynamic values β where the value of β could vary along the execution. Settings without any communication have also been experimented. Detection ratios between 0.2 and 0.5 were obtained.

Table 8 shows that limiting the number of messages does not significantly decrease the performances of the agents since important information is still exchanged. In fact, new observations are only taken into account once they substantially change the current context.

	3 agents	4 agents	5 agents
Com	0.7375	0.8084	0.8645
KL05	0.7241	0.7966	0.8585
KL10	0.6850	0.7476	0.8036
KL15	0.6661	0.7383	0.7660

Fig. 8. Average detection ratio

6.3. *Adversary policy changes*

Finally, we performed experiments to give more insights about the different steps of our method for detecting changes of the adversaries' policy. Figure 10 describes the variations of *det* for each step of our detection method along the execution. The adversaries changed their policies around the 400th time step. The mobile average was computed over 50 times steps and the threshold was fixed to 0.05. This threshold remains constant during the whole execution. The curve entitled *threshold* on Figure

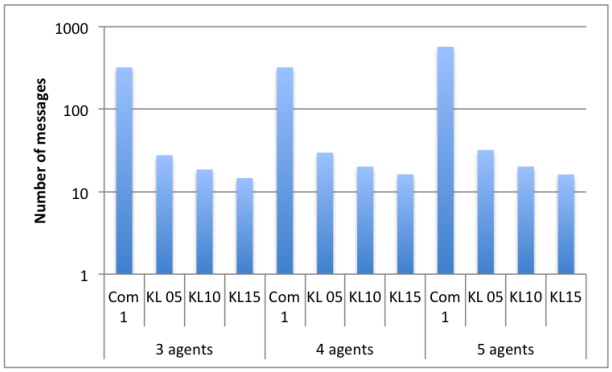

Fig. 9. Number of messages

10 allows for visualizing whether the decreasing component of Stieljes is under the threshold. It can be observed that the mobile average smooths the small variations of *det*. Stieljes decomposition allows for determining the decreasing component of the mobile average. One can checked that the value of the Stieljes decomposition falls under the threshold when the adversaries change their strategy.

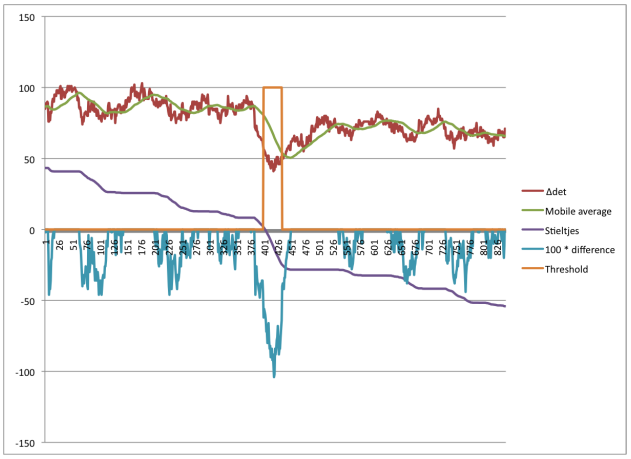

Fig. 10. Measures of the method for detecting policy changes

We also studied how the detection ratio evolves over time during the execution (Figure 11 assumes full communication between the agents). The detection ratio remains stable over the execution except when the adversaries change their strategy. In Figure 11, the sharp decrease in detection ratio around 270 is due to changes in the adversaries policy. This drop is successfully detected using the method described in Section 5.3 and the agents quickly adapt their strategy. The detection ratio thus returns to its previous level.

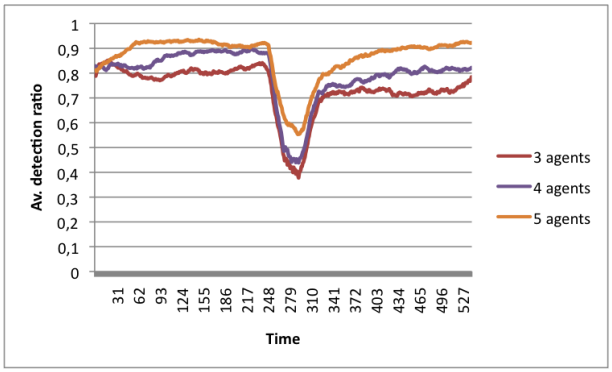

Fig. 11. Detection ratio over time (full communication)

Moreover, limiting the number of messages does not decrease the performances of the agents since significant information is still exchanged. Indeed, Figure 12 gives the evolution of the detection ratio over time if the agents limit the number of sent messages. However, it may be noticed that full communication leads to smoother curves since gathered information is continuously incorporated in the estimation of the adversarial strategies. On the other hand, under restricted communication new observations are only taken into account if they substantially change the current context.

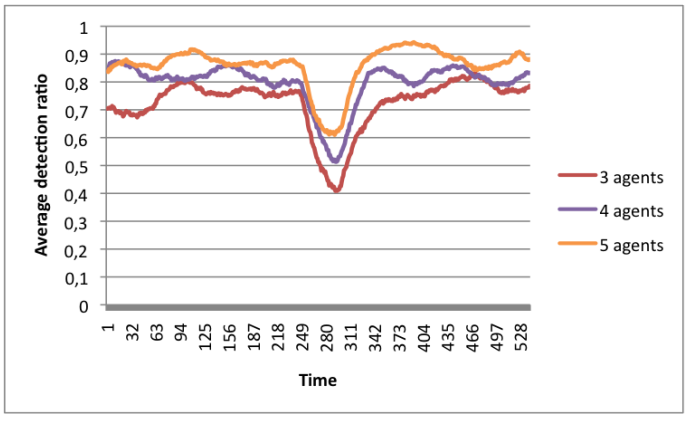

Fig. 12. Restricted communication

We also experimented the influence of the number of policy changes on the detection ratio (see Figure 13 for 4 agents and 16 targets). Drops in the detection ratio correspond to changes of the adversary strategies. Patrolling agents successfully detect context changes and adapt their strategy consequently. Thanks to our approach, patrolling performances quickly return to their previous level. Obviously,

the less the adversaries change their strategies, the higher the detection ratio. In fact, when the adversaries often change their strategy it becomes more and more difficult to deduce the current context. In highly dynamic problems, the detection ratio falls behind 0.6. Under such conditions, it would be recommended to increase the number of patrolling agents in the system.

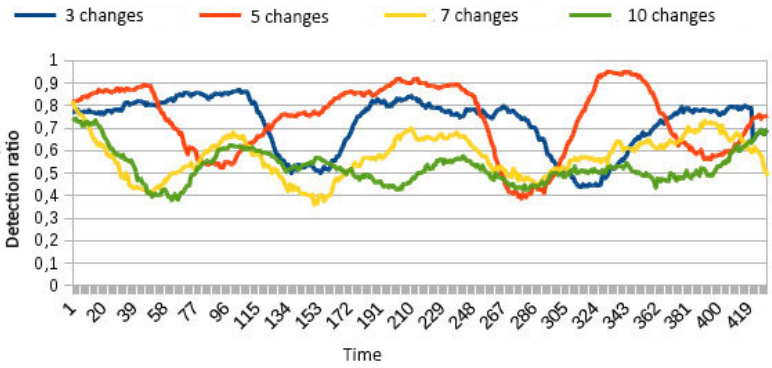

Fig. 13. Influence of the number of strategy changes

7. Conclusion

We introduced a new framework for effective cooperation between several patrolling agents acting in uncertain and partially observable environments. Our approach considers a generic model of the adversaries and does not make any restriction on the observability nor rationality of the opponents. Our approach thus provides new contributions to the domain along several dimensions by handling: multiple cooperative patrollers with limited observability, multiple adversaries performing multiple illegal actions over time and space, a generic probabilistic model of the adversaries, a more realistic model of actions formalizing uncertainty on action outcomes. To our knowledge, this framework is the first attempt to address all these issues together.

We proposed to formalize the multiagent patrolling problem as a DEC-POMDP. In order to cope with the non-stationarity of the adversaries behavior, we introduced the notion of *context* in multiagent Markovian models and we described a statistical approach to represent the profile of the adversaries and the current context. We hope that this approach could open the door to further works handling non-stationary in DEC-POMDPs.

We also presented a distributed solving approach based on evolutionary algorithms. Our algorithm computes patrolling strategies online and is able to exploit strategies of previous contexts to compute a new patrolling strategy for the current

context. Finally, we proposed approaches to improve the efficiency of the agents by allowing for a better detection of context changes and improving the quality of the information sent to the other agents.

Future work will explore more sophisticated models of the adversaries. We would like to include the temporal dimension in the context changes as it could have been done in HS3MDPs [28]. Models inspired from behavioral economics could also be useful to develop more accurate profile of the adversaries.

Appendix

In this section, we detail our decomposition of a series of variations y into two components y^+ and y^- such as $\forall t : y[t] = y^+[t] + y^-[t]$. As stated by Stieltjes in the continuous case, we demonstrate that y^+ is an increasing function and y^- is a decreasing function (i.e. $-y^-$ is an increasing function).

Proposition 1: For all series of variations y, the series y^+ defined as

$$y^+[0] = \tfrac{1}{2}y[0]$$

$$y^+[t] = \begin{cases} y^+[t-1] & \text{si } y[t] \leq y[t-1] \\ y^+[t-1] + y[t] - y[t-1] & \text{si } y[t] > y[t-1] \end{cases}$$

is increasing.

Proof: Let consider both cases separately:

- if $y[t] \leq y[t-1]$ then $y^+[t] = y^+[t-1]$ and $y^+[t] \geq y^+[t-1]$
- if $y[t] > y[t-1]$ then $y^+[t] = y^+[t-1] + y[t] - y[t-1]$ and $y^+[t] \geq y^+[t-1]$ since $y[t] - y[t-1] > 0$

In both cases, we thus have $y^+[t] \geq y^+[t-1]$ $\square$.

Proposition 2: For all series of variations y, the series y^- defined as

$$y^-[0] = \tfrac{1}{2}y[0]$$

$$y^-[t] = \begin{cases} y^-[t-1] & \text{si } y[t] \geq y[t-1] \\ y^-[t-1] + y[t] - y[t-1] & \text{si } y[t] < y[t-1] \end{cases}$$

is increasing.

Proof: Let consider both cases separately:

- if $y[t] \geq y[t-1]$ then $y^-[t] = y^-[t-1]$ and $y^-[t] \leq y^-[t-1]$
- if $y[t] < y[t-1]$ then $y^-[t] = y^-[t-1] + y[t] - y[t-1]$ and $y^-[t] \leq y^-[t-1]$ since $y[t] - y[t-1] < 0$

In both cases, we thus have $y^-[t] \leq y^-[t-1]$ □.

Proposition 3: For all series of variations y, y is the sum of y^+ and y^- with y^+ and y^- defined as described in Proposition 2 and Proposition 3.

Proof: We proceed by induction on the values t for which the series y is defined. We want to demonstrate that $y[t] = y^+[t] + y^-[t]$ for all t.

Base Case (t=0)
By definition, $y^+[0] = \frac{1}{2}y[0]$ and $y^-[0] = \frac{1}{2}y[0]$.
So, $y^+[0] + y^-[0] = \frac{1}{2}y[0] + \frac{1}{2}y[0] = y[0]$.
The equality holds for $t = 0$.

Induction Step We now assume that the equality holds for $t-1$ and show that it then holds for t:

- if $y[t] < y[t-1]$ then

$$y^+[t] + y^-[t] = y^+[t-1] + y^-[t-1] + y[t] - y[t-1]$$

Since (by assumption) $y[t-1] = y^+[t-1] + y^-[t-1]$ holds, we deduce

$$y^+[t] + y^-[t] = y[t-1] + y[t] - y[t-1] = y[n]$$

- if $y[t] \geq y[t-1]$ then

$$y^+[t] + y^-[t] = y^+[t-1] + y[t] - y[t-1] + y^-[t-1]$$

Since (by assumption) $y[t-1] = y^+[t-1] + y^-[t-1]$ holds, we deduce

$$y^+[t] + y^-[t] = y[t-1] + y[t] - y[t-1] = y[t]$$

The equality then also holds for t. □.

Example
Let consider the following series y of variations:

t	0	1	2	3	4	5
y[t]	0	3	5	2	6	4

This function can be decomposed into y^- and y^+ as follows:

$y^-[0] = 0$	$y^+[0] = 0$
$y^-[1] = 0$	$y^+[1] = 0 + 3 = 3$
$y^-[2] = 0$	$y^+[2] = 3 + 5 - 3 = 5$
$y^-[3] = 0 + 2 - 5 = -3$	$y^+[3] = 5$
$y^-[4] = -3$	$y^+[4] = 5 + 6 - 2 = 9$
$y^-[5] = -3 + 4 - 6$	$y^+[5] = 9$

It can be checked that for all t, $y[t] = y^+[t] + y^-[t]$.

In this paper, this decomposition is applied to the variations *det* on the number of detected illegal actions.

References

1. M. Jain, B. An and M. Tambe, An overview of recent application trends at the AAMAS conference: Security, sustainability and safety, *AI Magazine* **33**(3) (2012) 14–28.
2. N. Agmon, V. Sadov, G. A. Kaminka and S. Kraus, The impact of adversarial knowledge on adversarial planning in perimeter patrol, in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems (AAMAS '08)* **1**2008, pp. 55–62.
3. C. Zhang, A. Sinha and M. Tambe, Keeping pace with criminals: Designing patrol allocation against adaptive opportunistic criminals, in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS '15)*2015, pp. 1351–1359.
4. F. Fang, P. Stone and M. Tambe, When security games go green: Designing defender strategies to prevent poaching and illegal fishing, in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI '15)*2015.
5. F. Fang, T. H. Nguyen, R. Pickles, W. Y. Lam, G. R. Clements, B. An, A. Singh and M. Tambe, Deploying paws to combat poaching: Game-theoretic patrolling in areas with complex terrains, in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*2016.
6. E. A. Shieh, A. X. Jiang, A. Yadav, P. Varakantham and M. Tambe, An extended study on addressing defender teamwork while accounting for uncertainty in attacker defender games using iterative dec-mdps, *Multiagent and Grid Systems* **11** (2016) 189–226.
7. T. Flanagan, C. Thornton and J. Denzinger, Testing harbour patrol and interception policies using particle-swarm-based learning of cooperative behavior, in *Proceedings of the Second IEEE International Conference on Computational Intelligence for Security and Defense Applications CISDA'09*2009, pp. 215–222.
8. A. Beynier, Cooperative Multiagent Patrolling for Detecting Multiple Illegal Actions Under Uncertainty, in *International Conference on Tools with Artificial Intelligence (ICTAI)* (San José, United States, 2016). Best Paper.
9. B. An, M. Brown, Y. Vorobeychik and M. Tambe, Security games with surveillance cost and optimal timing of attack execution, in *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems (AAMAS '13)*2013, pp. 223–230.
10. Y. Chevaleyre, Theoretical analysis of the multi-agent patrolling problem, in *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT '04)*2004, pp. 302–308.
11. P. Paruchuri, J. P. Pearce, M. Tambe, F. Ordonez and S. Kraus, An efficient heuristic approach for security against multiple adversaries, in *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems (AAMAS '07)*2007, pp. 181:1–181:8.
12. N. Basilico, N. Gatti and F. Amigoni, Leader-follower strategies for robotic patrolling in environments with arbitrary topologies, in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '09)* **1**2009, pp. 57–64.

13. N. Basilico, N. Gatti and F. Amigoni, Developing a deterministic patrolling strategy for security agents, in *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 02 WI-IAT '09*, (IEEE Computer Society, 2009), pp. 565–572.
14. C. Kiekintveld, M. Jain, J. Tsai, J. Pita, F. Ordóñez and M. Tambe, Computing optimal randomized resource allocations for massive security games, in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '09)*2009, pp. 689–696.
15. N. Basilico, N. Gatti, T. Rossi, S. Ceppi and F. Amigoni, Extending algorithms for mobile robot patrolling in the presence of adversaries to more realistic settings, in *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 02 WI-IAT '09*, (IEEE Computer Society, 2009), pp. 557–564.
16. N. Agmon, S. Kraus, G. A. Kaminka and V. Sadov, Adversarial uncertainty in multi-robot patrol, in *Proceedings of the 21st international jont conference on Artifical intelligence (IJCAI'09)*2009, pp. 1811–1817.
17. Y. Qian, W. B. Haskell, A. X. Jiang and M. Tambe, Online planning for optimal protector strategies in resource conservation games, in *Proceedings of the 13th International Conference on Autonomous Agents and Multi-agent Systems (AAMAS '14)*2014, pp. 733–740.
18. E. Munoz de Cote, R. Stranders, N. Basilico, N. Gatti and N. Jennings, Introducing alarms in adversarial patrolling games: Extended abstract, in *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems AAMAS '13*, (International Foundation for Autonomous Agents and Multiagent Systems, 2013), pp. 1275–1276.
19. T. H. Nguyen, A. Sinha, S. Gholami, A. Plumptre, L. Joppa, M. Tambe, M. Driciru, F. Wanyama, A. Rwetsiba, R. Critchlow and C. Beale, Capture: A new predictive anti-poaching tool for wildlife protection, in *15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '16)*2016.
20. Z. Yin, M. Jain, M. Tambe and F. Ordóñez, Risk-averse strategies for security games with execution and observational uncertainty., in *AAAI*, eds. W. Burgard and D. Roth2011.
21. D. Bernstein, S. Zilberstein and N. Immerman, The complexity of decentralized control of mdps, in *Mathematics of Operations Research*2002, pp. 27(4):819–840.
22. E.A. Hansen, D. Bernstein and S. Zilberstein, Dynamic programming for partially observable stochastic games, in *Proceedings of the Nineteenth National Conference on Artificial Intelligence*2004.
23. D. Szer, F. Charpillet and S. Zilberstein, Maa*: A heuristic search algorithm for solving decentralized pomdps, *CoRR* **abs/1207.1359** (2012).
24. J. Dibangoye, C. Amato, O. Buffet and F. Charpillet, Exploiting separability in multi-agent planning with continuous-state MDPs, in *Proceedings of the 11th International Conference on Autonomous Agents and Multi-agent Systems (AAMAS '14)*2014, pp. 1281–1288.
25. R. Nair, M. Tambe, M. Yokoo, D. Pynadath and S. Marsella, Taming decentralized pomdps: Towards efficient policy computation for multiagent settings, in *Proceedings of the 18th International Joint Conference on Artificial Intelligence IJCAI'03*2003, pp. 705–711.
26. S. Seuken and S. Zilberstein, Memory-bounded dynamic programming for dec-pomdps, in *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI)*2007, pp. 2009–2015.

27. C. Amato, A. Carlin and S. Zilberstein, Bounded dynamic programming for decetralized pomdps, in *AAMAS 2007 Workshop on Multi-Agent Sequential Decision Making in Uncertain Domains*2007.
28. E. Hadoux, A. Beynier and P. Weng, Solving Hidden-Semi-Markov-Mode Markov Decision Problems, in *Scalable Uncertainty Management Lecture Notes in Computer Science* **8720**September 2014, pp. 176–189.
29. P. Doshi and P. J. Gmytrasiewicz, A framework for sequential planning in multi-agent settings, *CoRR* **abs/1109.2135** (2011).
30. M. Jain, B. An and M. Tambe, An overview of recent application trends at the aamas conference: Security, sustainability and safety, *AI Magazine* **33**(3) (2012) 14–28.
31. C. Amato, G. Chowdhary, A. Geramifard, N. K. Ure and M. J. Kochenderfer, Decentralized control of partially observable markov decision processes, in *IEEE Conference on Decision and Control* (Florence, Italy, 2013).
32. A. Kumar and S. Zilberstein, Point-based backup for decentralized POMDPs: Complexity and new algorithms, in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '10)* (Toronto, Canada, 2010), pp. 1315–1322.
33. M. Mazurowski and J. Zurada, Solving decentralized multi-agent control problems with genetic algorithms, in *IEEE Congress on Evolutionary Computation*2007.
34. B. Eker and H. L. Akın, Solving decentralized pomdp problems using genetic algorithms, *Autonomous Agents and Multi-Agent Systems* **27** (July 2013) 161–196.
35. S. Droste, T. Jansen and I. Wegener, On the analysis of the (1+1) evolutionary algorithm, *Theoretical Computer Science vol. 276* (2002).
36. S. Kullback and R. Leibler, On information and sufficiency, *Annals of Mathematical Statistics* (1951).
37. T. Kailath, The divergence and bhattacharyya distance measures in signal selection, in *IEEE Transactions on Communication Technology*1967.
38. A. Bhattacharyya, On a measure of divergence between two multinomial populations (1946) 401–406.
39. Madp toolbox.