Cooperative Multiagent Patrolling for Detecting Multiple Illegal Actions Under Uncertainty

Aurélie Beynier

Cooperative Multiagent Patrolling for Detecting Multiple Illegal Actions Under Uncertainty

Aurélie Beynier
Sorbonne Universités, UPMC Univ Paris 06
CNRS, UMR 7606, LIP6
F-75005, Paris, France
aurelie.beynier@lip6.fr

***Abstract*—Multiagent patrolling in adversarial domains has been widely studied in recent years. However, little attention has been paid to cooperation issues between patrolling agents. Moreover, most existing works focus on one-shot attacks and assume full rationality of the adversaries. Nonetheless, when patrolling frontiers, detecting illegal fishing or poaching; security forces face several adversaries with limited observability and rationality, that perform multiple illegal actions spread in time and space. In this paper, we develop a cooperative approach to improve defenders efficiency in such settings. We propose a new formalization of multiagent patrolling problems allowing for effective cooperation between the defenders. Our work accounts for uncertainty on action outcomes and partial observability of the system. Unlike existing security games, a generic model of the opponents is considered thus handling limited observability and bounded rationality of the adversaries. We then describe a learning mechanism allowing the defenders to take advantage of their observations about the adversaries and to compute cooperative patrolling strategies consequently.**

I. Introduction

During the past years an increasing focus has been put on multiagent patrolling and threat detection. Security Games have mainly been interested in detecting a single adversary trying to perform a one-shot attack such as a terrorism attack [1]. More recently, some works have investigated a wider variety of patrolling domains like preventing crime in urban areas [2], avoiding intrusions on frontiers [3], detecting illegal fishing or poaching [4]. In such settings, the defenders face multiple adversaries performing frequently and repeatedly illegal actions. For instance, illegal fishing may occur simultaneously on different sites and is a daily problem for cost guards. When patrolling frontiers, police forces have to repeatedly face multiple intrusions on various crossing points. In these contexts, deploying several patrollers able to effectively coordinate and plan their strategies, provide promising opportunities to reinforce security.

Indeed, as shown by Shieh et al. [5] in the context of a one-shot attack, defender teamwork leads to significant improvement over other approaches. Nonetheless, even if there has been an increasing focus on approaches handling multiple adversaries and frequent illegal actions [6], [4], existing works do not allow for an effective teamwork among defenders. In this paper, we introduce an approach allowing for effective cooperation among several defenders to cope with multiple illegal actions spread in time and space. We consider mobile patrollers that are able to move across their environment. Patrollers will thus have to cooperate in order to choose which sites to visit next. Obviously, in real settings, patrollers will have to cope with partial observability of their environment and uncertainty on action outcomes. For instance, when fighting illegal fishing, cost guards' moves depend on weather conditions. Move duration between fishing spots is thus uncertain. Moreover, cost guards cannot observe all fishing spots at any time of their patrol. In order to compute valuable strategies, there is a need to handle uncertainty on action outcomes and partial observability of the system.

We thus aim at developing cooperative patrolling strategies that detect as much illegal actions as possible, under uncertainty. Patrollers will have to anticipate possible actions of the adversaries at each time-step. To deal with this issue, most existing approaches assume that adversaries are fully rational and fully observe the patrolling strategy. A best response can then be anticipated and a defender strategy is computed using Game Theory. In fact, it is often difficult for the adversaries to obtain full knowledge of the patrolling strategy. Indeed, adversaries are often unable to fully observe the patrolling strategy since they have limited observation capacities, observing the patrolling strategy is risky (the adversary may be detected) and costly (it takes time and consumes resources) [7], [4]. Moreover, adversaries (mostly humans) often have bounded rationality. Existing approaches assuming full observability and rationality of the adversary thus fail to develop efficient strategies in many real-world settings. Besides that adversaries may have bounded rationality and do not act optimally, their strategies may evolve over time. For instance, illegal fishermen can change their fishing spots based on their accumulated knowledge about cost guard patrols. It is thus difficult for the defenders to anticipate the response of the adversaries to their patrolling actions.

In this paper, we attempt to improve the applicability and efficiency of patrolling teams while facing multiple adversaries that perform multiple illegal actions. To reach that aim, our work makes the following contributions:

- We propose a new formalization of the defenders' decision problem allowing for effective cooperation between the defenders while facing multiple adversaries performing repeated illegal actions over time and space.
- Our approach handles uncertainty on action outcomes and partial observability of the environment. Moreover,

we do not make specific assumptions on the observation capacities and on the rationality of the adversaries.
- We propose a distributed solving algorithm able to compute online and update patrolling strategies.
- We develop a mechanism allowing defenders to build online a model of the adversaries and to detect changes in adversary strategies. Our approach is thus able to handle non-stationary adversary strategies i.e. strategies of the adversaries that evolve over time.

This paper first reviews related works and introduces our problem settings. Second, it provides a new formalization of the multiagent patrolling problem with adversaries based on Decentralized Partially Observable Markov Decision Processes (DEC-POMDPs). Third, methods are proposed for the patrollers to compute their action policies and to adapt them to possible changes in adversary strategies. Fourth, experiments on the efficiency of our approach are presented.

II. Related Work

Several approaches have recently been proposed to deal with patrolling in adversarial settings. They involve two kinds of agents: the patroller and the intruder (i.e. the adversary). Existing works make different assumptions about the agents' observability of the system, leading to different kinds of solutions. Many approaches assume that the adversary is able to perform extensive observation of the patrollers and then conducts a one-shot attack [8], [9], [10], [7]. Such a strong adversary has thus full knowledge of the patrolling strategy and the problem can be represented as a leader-follower decision problem. These approaches refer to "security games". However, in many domains, considering such a strong adversary is not realistic nor optimal. Agmon et al. [3], [11] 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.

Most multiagent patrolling approaches do not tackle the issue of cooperation between multiple patrollers since they assume only one patrolling agent or they consider that the same strategy is executed by all patrollers [11]. Recently, Shieh et al. [5] proposed to combine security games and Decentralized MDPs to enable effective cooperation between several patrollers under uncertainty. The approach computes a defender's team strategy facing a single adversary that performed a prior extensive surveillance phase. In Shieh et al.'s work, the adversary is assumed to be fully rational and chooses to attack the target with the lowest coverage.

When patrolling frontiers or fighting against illegal fishing or poaching, defenders face multiple intruders [4]. The objective of the defenders is then slightly different as it consists in maximizing the number of detected illegal actions instead of preventing a one-shot attack. Moreover, the observability and knowledge of the system are often limited: attackers do not have enough time nor resources to fully observe the patrolling strategies. Defenders can thus not count for a best response strategy from the intruders. The defenders have then to consider bounded rationality and limited obsevability of the intruders. CAPTURE framework [12] has been proposed to prevent poaching while facing an uncertain number of poachers and limited observability of the poachers. However, the approach again considers a one-shot attack and does not allow for effective teamwork between the patrollers. Qian et al. [13] relaxed the assumption of a one-shot attack following a prior extensive surveillance and considered several illegal actions while a single protector and an adversary fully observe their opponent's actions. Recently, Nguyen et al. [12] 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.

III. Problem setting

We address the problem of computing patrolling strategies for m heterogeneous defenders (agents i with $i \in [1, m]$) that have to coordinate to patrol a set of n (with $m \ll n$) target sites t_j (with $j \in [1, n]$) to detect illegal actions. Defenders consist of mobile agents able to conduct surveillance on the targets. The environment topology is represented as a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ formalizing possible routes between the targets. $\mathcal{N} = \{t_1, \cdots, t_n\}$ denotes the set of targets and $\mathcal{E}$ is the set of possible routes between the targets.

We consider that uncertainty arises from imperfect action execution and limited observation. A probability distribution $C_{k,j}$ on possible travel durations is thus assigned to each edge $e = (t_k, t_j) \in \mathcal{E}$ of the graph. Since moving from one target to another can take several time steps, it should be noticed that a patrolling agent does not make a new decision at each time step and agents are not fully synchronized.

Like previous works dealing with patrolling in adversarial domains, it is assumed that performing an illegal action is not instantaneous and takes Δ_{int} time steps [3].

As mentioned previously, we are interested in realistic settings where patrollers and intruders have partial observability of each other. Thus, each patrolling agent is assumed to know her own location and only observes adversaries performing illegal actions on the target she is currently patrolling. 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 framework is thus generic and does not assume a fixed known number of adversaries. The number of adversaries can then evolve over time and is unknown to the patrollers. We do not make any assumption on the full rationality of the adversaries nor their possible cooperation. Instead, defenders will try to anticipate the adversary behavior from the observations made during the patrol.

IV. Decision problem formalization

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's decision is based on her local knowledge about the adversaries and about the environment. In the following, we show that this distributed and cooperative decision problem can be formalized as a Decentralized Partially Observable Markov Decision Process (DEC-POMDP) [14]. In fact, DEC-POMDPs allow for formalizing sequential multiagent decision problems under uncertainty where a set of cooperative agents has to make decisions in a distributed way with partial observabilty of the system.

In real settings, because of limited observability and bounded rationality, it is very difficult for the patrollers 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 and have to make decisions based on limited knowledge about adversary behaviors. On the other hand, the adversaries have bounded rationality and may not always commit to an optimal policy. They also continuously adapt their strategy from their local observations. One of the main difficulty to overcome is that patrolling agents cannot compute a best response of the adversaries to anticipate their behavior. 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 dynamics of the adversary strategy, our approach also proposes mechanisms for the patrolling agents to be able to detect policy changes as soon as possible, to update the context and to adapt patrolling strategies consequently.

A. Background on DEC-POMDPs

A DEC-POMDP 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, $A = \{A_1 \times \cdots \times A_m\}$ 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 s' while executing a from s, $O = \{O_1 \times \cdots \times O_m\}$ is the set of joint observations $o = \{o_1, ..., o_m\}$, Ω is the observation function giving the probability $\Omega(o|s, a)$ of observing o when executing a from s, $R(s'|s, a)$ is the reward obtained when executing action a from s and moving to 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. It has been proved that optimally solving a DEC-POMDP is NEXP-Complete [14]. Many works have thus focused on developing efficient solving methods. Despite major advances in the scalability of the algorithms, the size of the problems that can be optimally solved remains limited [15]. In fact, approximate approaches often better scale to large number of agents and planning horizon. Moreover, it has to be noticed that most existing approaches perform centralized off-line planning i.e. planning is performed in a centralized way before the execution. Strategy computation can then not be distributed among the agents. Finally, the issues of learning or re-planning during the execution have received little attention so far.

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) [16] 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 [17] assume a known model of the adversary and do not account for effective cooperation during action execution.

B. Patrollers' DEC-POMDP formulation

We now describe how the cooperative patrolling problem can be represented as a DEC-POMDP. Of course, patrolling strategies rely on the current knowledge about the adversaries. However, in our settings, the only observed information about the adversary strategies consists in detected illegal actions. To formalize the information the patrollers have about the adversaries, we propose to use a probability distribution PI defined for each target. $PI_i(t)$ is 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 $1 - PI_i(t)$. The probabilities PI_i may evolve over time as the patrollers get more and more observations about the adversaries. Moreover, the adversaries may change their strategy. PI is thus non-stationary over time.

Inspired by the notion of modes used in mono-agent POMDP [18], our approach consists in defining a DEC-POMDP for a given distribution PI which will be referred as the current context of the decision making. As discussed later in the paper, the DEC-POMDP definition will have to be updated as the context evolves.

Actions: At each decision step, an agent must decide for the next target to patrol. An individual action a_i thus consists in *moving to target* t_j ($t_j \in \mathcal{N}$). Since the execution of an action may last over several time steps, agents may not be synchronized. DEC-POMDPs consider one time unit action duration so, we propose to decompose individual moves from a target to another 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. Although an agent has prior knowledge on the possible durations c_{kj}, she actually knows the effective duration of the move only once it has been fully executed (the agent has reached t_j). During the execution of the unitary moves, the agent does not change her decision and keeps executing the action *moving to target* t_j.

States: A state s_t at time t 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. The position p_i of each agent i is 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 has been currently observed on this target. The state thus contains the list int of targets where intrusions have been observed at t. The time elapsed since the last visit of each target is called *idleness*. 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. A state s_t is thus a tuple of the form $\langle p = \langle p_1, ..., p_m \rangle, int, idle = \langle idle_1, ..., idle_n \rangle, \delta = \langle \delta_1, \cdots, \delta_m \rangle \rangle$. To avoid overloading equations, we will denote this state by $s_t = \langle p, int, idle, \delta \rangle$.

Observations: Each agent observes her current position. 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: Transition probabilities are defined from probabilities on move durations and from probabilities on detection of illegal actions. At each decision step, some agents reach new targets to visit whereas other agents are still on the road between two targets. 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 $C_{k,j}$.

Probabilities on the detection of illegal actions are estimated using the current context PI. When the agent i reaches her destination t_j, she may observe an illegal action on t_j. The probability w_j of observing an illegal action on t_j at t is the probability that an adversary initiated such an action within the last Δ_{int} time steps and it has not been detected yet:

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

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 re-written as a sum of probabilities on conjunctions of events $\mathcal{I}_j$. The probability of such an event is then given by PI_j.

Observation function: Since agents are assumed to always detect illegal actions on the targets they patrol, there is no uncertainty on observations.

Reward function: The reward function is defined in order to reward detected illegal actions. 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. Agents are thus encouraged to patrol the targets that were visited a long time ago. Different rewards over the sites can be defined so as to represent the relative significance of the targets.

Since agents have limited observability of the targets and of the adversaries, it has to be noticed that it is not possible to represent non-detected illegal actions in the DEC-POMDP and to take them into account in the reward function.

C. Computation of the current context

Our DEC-POMDP formalization assumes a fixed model PI of the adversaries strategy. This model (also referred as "context") is built by the patrolling agents during the execution of their actions. Of course, as patrolling agents make more and more observations, they obtain a more accurate model of the adversaries.

If patrolling agents are only aware of detected illegal actions, they cannot have prefect knowledge of the probability distribution PI. However, the number of detected illegal actions over the last $\mathcal{H}$ time steps can be used to compute an estimate of PI for each target. 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 define the following estimate:

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

This definition may appear as a rough estimate but it guarantees the relation: $NI_i(t-\mathcal{H}, t) > NI_k(t-\mathcal{H}, t) \Rightarrow PI_i(t) > PI_k(t)$. Moreover, this estimate meets our assumption on limited observability and fits nicely to the patrolling agents' objective consisting in detecting as much illegal actions as possible. However, our DEC-POMDP formalization allows for other definitions of PI exploiting possible higher degrees of observability or external knowledge about the adversaries.

Since updating the DEC-POMDP model and the patrolling strategies at each time step from the new current context is too costly (in terms of time and computational resources), we introduce a context horizon $\mathcal{T}$ that sets the period of validity of the current context. The DEC-POMDP model is then updated every $\mathcal{T}$ steps using the new context computed from the observations made by the patrolling agents over the last $\mathcal{H}$ steps. New patrolling strategies will then be executed. Note that $\mathcal{T}$ is the planning horizon whereas $\mathcal{H}$ is the length of the observation history used to define a context PI.

It has to be pointed out that our approach allows patrolling agents to adapt online their strategies. From the point of view of the attackers, even if they had full observability of the patrollers, the patrolling strategy will then not seem deterministic all along the execution.

V. Policy Computation and Updates

For a current context PI, solving the corresponding DEC-POMDP returns a joint policy $\pi = \{\pi_1, ..., \pi_m\}$ maximizing the global expected reward derived from Equation 1. An individual policy π_i allows agent i to decide, from her observations of the system, how to act in a cooperative way. Note that agents usually end with different strategies: the nodes of the graphs are dispatched among the agents taking into account action

uncertainty, target rewards and threat levels. Several algorithms have been developed to solve DEC-POMDPs [15], [19], [20]. They can be used to solve our DEC-POMDP formalization described in the previous section. An optimal policy can then be computed for a given context PI. Each time a new context is deduced from new observations, the patrolling strategy has to be computed again given the new context. However, in our settings, existing approaches suffer from several limitations. First, they perform centralized computation of the patrolling strategy. In fact, even if patrolling strategies can be executed in a distributed way, most existing approaches consider a central entity computing the individual strategies and then broadcasting them to the agents. Each time a new context is considered, such a central entity would thus have to collect all the observations of the patrollers, to compute a new strategy and to broadcast it. This would create a bottleneck in the system and would result in high communication cost. The second drawback of these algorithms is that they have not been developed to exploit learned information and to update strategies during the execution. They cannot re-use previously computed strategies to speed up the computation of a new joint strategy for a new context. Finally, despite recent advances, existing approaches fail to scale to large sizes of problems.

In this section, we investigate evolutionary algorithms to compute approximate patrolling strategies in a distributed way. Such algorithms have already been used to compute approximate solutions for DEC-POMDPs [21], [22] and showed significance improvement in the size of the horizon that can be considered. In addition, our algorithm allows for exploiting previously computed strategy when considering a new context. We then propose to reduce communication overhead and to improve patrolling performance.

A. Evolutionary Algorithm for policy computation

We propose to adapt the (1+1) evolutionary algorithm [23] to optimize the patrolling strategy over horizon $\mathcal{T}$. The evolutionary algorithm (see Algorithm 1) selects an initial solution (called `champion`) and then iterates to improve the `champion` until a computation deadline is reached. At each iteration, a mutation operator is applied to the current `champion` thus obtaining a `challenger`. This new solution is evaluated and becomes the new `champion` if its value is higher than the one of the 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 population of individuals (i.e. set of policies) consists in the set of joint policies that comply with temporal and spatial constraints of the problem. Spatial constraints are fulfilled if each agent always executes an admissible action at each decision step i.e. 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. The initial solution is built from a probability distribution over the nodes reflecting the likelihood of an illegal action on each site deduced from observations. In fact, the higher the probability of an illegal action on a target t_k, the higher the probability of selecting t_k. Probabilities are also weighted by the visit frequency of the target in the previous context. The mutation of the current champion strengthens the weakest targets: the targets with the lowest probabilities of threats are replaced by the targets with the highest probabilities of threats. This algorithm has the advantage of being anytime and better scales to large numbers of agents and long planning horizon $\mathcal{T}$ because of its low complexity. However, it does not provide any guarantee on the quality of the solution regarding the optimum. This issue will be investigated in our experiments.

B. Communication models

Executing our evolutionary algorithm in a distributed way requires the agents to communicate their observations about detected illegal actions. In fact, this information is necessary to deduce the new context. However, communication can be risky (adversaries may listen) and resource-consuming. In order to limit the number of messages, we propose to measure the relevance of the communicated information. An information is communicated to the other agents if it is considered as relevant. We thus measure the distance between the current probability distribution PI and the new probability distribution PI' obtained by exploiting the information not communicated so far. To compute the distance between two probability distributions, we use the Kullback-Leibler divergence [24] which measures the difference between two probability distributions P and Q. The divergence from P to Q is defined as: $\sum_i P(i) log \frac{P(i)}{Q(i)}$. In order to preserve the symmetry property, we use the pseudo-distance:

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

If the Kullback-Leibler distance is greater than a small β parameter value, the information is considered as being relevant and it is communicated to all teammates. Other measures have been investigated, like the Bhattacharyya distance or the Hellinger distance but, we did not notice any significant difference in the solutions we obtained.

C. Detection of context changes

Although adversaries are assumed to partially observe the patrollers, they may adapt their strategy from their observations of the system. As explained in Section IV-C, observations about the adversaries over the last $\mathcal{H}$ steps are used to define a new context every $\mathcal{T}$ time steps. In this section, we propose to improve patrolling performance by detecting adversary policy changes during the $\mathcal{T}$ time steps of a context. We thus

define a mathematical method that studies the variations of the number of detected adversaries det over the $\mathcal{H}$ last time steps considered in Equation 2. Empirical studies of the variations of det showed that this quantity significantly decreases when the adversaries change their strategy. The objective of the following method is thus to better detect such decreases by applying the four successive processing operations:

1) Compute a moving average of $det_t(\mathcal{H})$ values over the last $\mathcal{H}$ time steps for each time step t.
2) Decompose det_t values using a finite adaptation of Stieltjes decomposition[1]. This decomposition allows us to identify decreasing components det^- of det.
3) Apply a backward finite difference operator:
$$\nabla_{det^-}[t] = det_t^-(\mathcal{H}) - det_{t-1}^-(\mathcal{H})$$
to quantify variations.
4) Threshold the values obtained in the previous step to detect adversarial policy changes.

When an adversarial policy change is detected, the current context PI is updated even if the deadline $\mathcal{T}$ has not been reached. Note that this method can be applied irrespectively of the solving algorithm. The threshold of the procedure has to be tuned considering the DEC-POMDP formalization of the problem. Small thresholds lead to more sensitive detection but could lead to "false" detection of strategy changes. High thresholds might miss some strategy changes. Anyway, the context will be updated and new policies will be computed once the horizon $\mathcal{T}$ is reached.

VI. Experiments

We experimented our approach on different sizes of randomly generated graphs. Graph connections and probabilities on action durations were randomly generated. Each node is connected in average to 2/3 of the other nodes. Action durations were randomly drawn in the interval [1,5]. Initial adversarial strategies 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. For each experiment, the system was executed over at least 400 time steps and the adversaries changed their policies at least once during the execution. Performing an illegal action was assumed to take 10 time steps. Experiments were performed on a computer equipped with an Intel(R) Core(TM)2 Duo processor, 2000 MHz, 8Gb.

A. Performances and scalability

We first studied the detection ratio (number_of_detected_illegal_actions / number_of_illegal_actions) over the patrolling mission. We compared the results obtained by our evolutionary algorithm with the optimal solution computed using the MADP toolbox [25]. Figure 1 gives the detection ratio for small scenarios (2 agents and 5, 6 or 7 targets) executed over 530 time steps. It can be seen that our evolutionary algorithm leads to high detection ratios. In fact, the detection ratio is decreased by only 3% over the optimal solution for 5 targets and by 6% for 7 targets. Note that even the optimal approach does not lead to full detection of illegal actions since agents cannot cover all targets within 10 time steps (duration of an illegal action).

[1]For space reasons, we do not detail this decomposition.

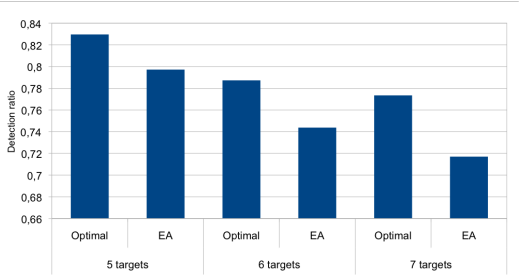

Fig. 1: Detection ratios of the executed strategies

We then studied the scalability of our approach and tested the performances of the solutions. We were not able to optimally solve problems larger than 2 agents and 7 targets. Nonetheless, our evolutionary algorithm successfully solved problems up to 50 targets and 7 agents (with a deadline of 10 seconds). Larger problems can even be solved by enlarging the deadline of the evolutionary algorithm.

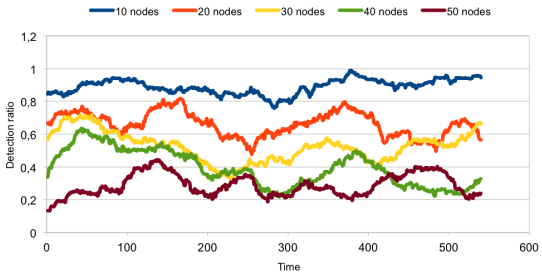

Fig. 2: Influence of the number of targets

We experimented our approach on several sizes of graphs (Figure 2) considering a fixed number of 4 agents. The larger the number of targets, the lower the detection ratio. 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. In order to guarantee good performances of patrolling agents, a minimum number of agents is required. This number is closely related to the number of targets to patrol and to move durations. In our 4-agent experimental setting, it can be observed that good performances are obtained for graphs smaller or equal to 16 targets. Indeed, our approach is thus able to provide detection ratio of 80% and more. We investigated deeper the influence of the number of agents on the detection ratio. 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), our approach provides good detection ratios for $m \ll n$. As shown in Figure 3 for a 16-target graph, if $m = n$, the detection ratio obviously equals to 1. Nonetheless, it can be observed that this value is almost reached with 12 agents. Moreover, considering 6 agents leads to a detection ratio greater than 0.9.

Since the deadline of the evolutionary algorithm and the planning horizon both influence solution quality, we varied

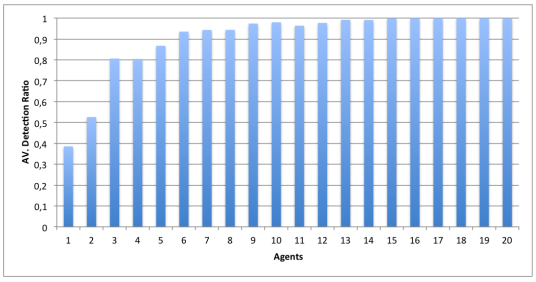

Fig. 3: Influence of the number of agents

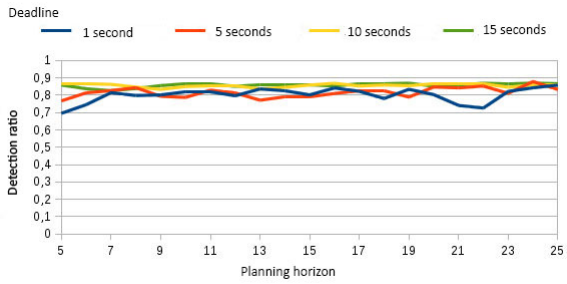

Fig. 4: Influence of the deadline and of the planning horizon

the value of these parameters and studied the average detection ratio. As it can be seen on Figure 4 for 5 agents and 16 targets, increasing the deadline of the evolutionary algorithm over 1 second does not significantly improve solutions. Indeed, good solutions are computed within a short delay. Furthermore, increasing the planning horizon $\mathcal{T}$ over 10 time steps does not improve the detection ratio neither. In fact, a good setting of 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 reduces the frequency of strategy updates (planning is performed every $\mathcal{T}$ steps). On the other hand, shorter planning horizons increases the frequency of context and policy updates. More variations are then observed in the detection ratio since agents become more and more myopic and are unable to anticipate future action outcomes and opportunities.

B. Communication model

We have then experimented the influence of using Kullback-Leibler divergence on the number of messages and on the detection ratio. Figure 6 and Table 5 illustrate the impact of limiting communication in a problem composed of 16 targets considering different number of agents. Note that a logarithmic scale is used in Figure 6. We varied the β threshold used to decide whether the message must be sent ($\beta \in \{5, 10, 15\}$) and we compared with the approach consisting in always broadcasting observations about the adversaries ("Com"). As shown in Table 5, 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. Nonetheless, we noticed a significant decrease in the number of sent messages (Figure 6). In the 5-agents case, the number of messages has been reduced from 574 to 16. β value can be used as a parameter to tune the frequency of communication. In the 5-agents case, while varying β between 5 and 15, the number of messages decreases from 32 to 16. Settings without any communication have also been experimented leading to detection ratios between 0.2 and 0.5.

	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. 5: Average detection ratio

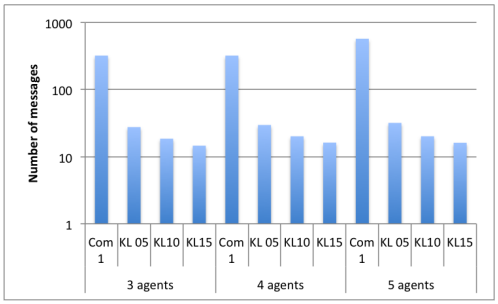

Fig. 6: Number of messages

C. Adversary policy changes

We also studied how the detection ratio evolves over time during the execution (Figure 7). The detection ratio remains stable over the execution except when the adversaries change their strategy. In Figure 7, 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 V-C and the agents quickly adapt their strategy. The detection ratio thus returns to its previous level.

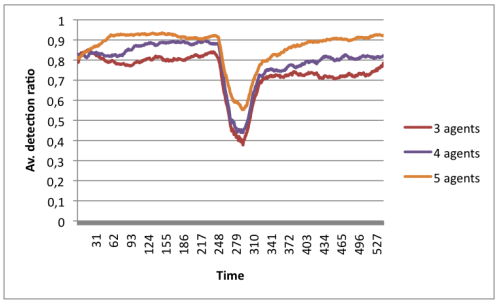

Fig. 7: Detection ratio over time

As shown in Figure 8 for 4 agents and 16 targets, we varied the number of times the adversaries change their strategy and we studied the impact on the detection ratio. 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. Performances could then be improved by introducing more patrolling agents.

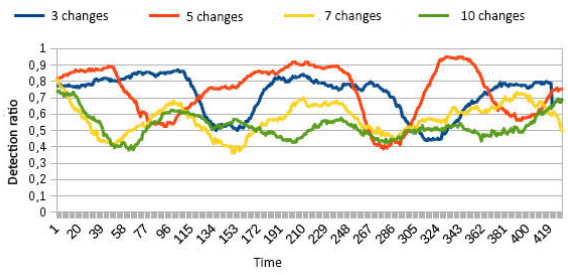

Fig. 8: Influence of the number of strategy changes

VII. Conclusion

We presented a new framework for effective cooperation between patrolling agents in uncertain and partially observable domains. We relaxed the hypothesis of a single strong and rational opponent, usually done in existing approaches. We also proposed a more realistic model that accounts for uncertainty on action outcomes. Our approach thus extends existing multiagent patrolling works along several dimensions: multiple cooperative patrollers with limited observability, multiple adversaries performing multiples illegal actions, partially observable adversary strategies which may not be rational and evolving over time, uncertainty on action duration. To our knowledge, our framework is the first attempt to address all these issues together. We formalized the multiagent decision problem as a DEC-POMDP defined for a fixed context formalizing the adversary behavior. We showed how this context can be built from the observations made by the patrollers and we proposed a method to cope with the non-stationarity of adversary behaviors i.e. contexts. Patrollers are thus able to detect changes of adversary strategies and adapt their behavior consequently. We then proposed an evolutionary algorithm which offers an alternative to existing solving algorithms. Our algorithm computes patrolling strategies in a distributed way and is able to re-use strategies of previous contexts to compute a new patrolling strategy adapted to the current context. Future work will explore more complex representations of the adversaries including the temporal dimension in the context. Intuitively, time influences the actions of the adversaries and the likelihood of changes in the adversaries strategies (an illegal action is less likely to occur on a target if such an action has just been detected on that target).

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*, vol. 33, no. 3, pp. 14–28, 2012.

[2] 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.

[3] 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)*, vol. 1, 2008, pp. 55–62.

[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] 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*, vol. 11, pp. 189–226, 2016.

[6] 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.

[7] 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.

[8] 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.

[9] 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)*, vol. 1, 2009, pp. 57–64.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] P. Doshi and P. J. Gmytrasiewicz, "A framework for sequential planning in multi-agent settings," *CoRR*, vol. abs/1109.2135, 2011.

[17] M. Jain, B. An, and M. Tambe, "An overview of recent application trends at the aamas conference: Security, sustainability and safety," *AI Magazine*, vol. 33, no. 3, pp. 14–28, 2012.

[18] E. Hadoux, A. Beynier, and P. Weng, "Solving Hidden-Semi-Markov-Mode Markov Decision Problems," in *Scalable Uncertainty Management*, ser. Lecture Notes in Computer Science, vol. 8720, Sep. 2014, pp. 176–189.

[19] 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.

[20] 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.

[21] M. Mazurowski and J. Zurada, "Solving decentralized multi-agent control problems with genetic algorithms," in *IEEE Congress on Evolutionary Computation*, 2007.

[22] B. Eker and H. L. Akın, "Solving decentralized pomdp problems using genetic algorithms," *Autonomous Agents and Multi-Agent Systems*, vol. 27, no. 1, pp. 161–196, Jul. 2013.

[23] S. Droste, T. Jansen, and I. Wegener, "On the analysis of the (1+1) evolutionary algorithm," *Theoretical Computer Science vol. 276*, 2002.

[24] S. Kullback and R. Leibler, "On information and sufficiency," *Annals of Mathematical Statistics*, 1951.

[25] Madp toolbox. [Online]. Available: http://www.fransoliehoek.net/index.php?fuseaction=software.madp