Minimizing Expected Intrusion Detection Time in Adversarial Patrolling

Extended Abstract

David Klaška
Masaryk University
Brno, Czechia
david.klaska@mail.muni.cz

Antonín Kučera
Masaryk University
Brno, Czechia
tony@fi.muni.cz

Vít Musil
Masaryk University
Brno, Czechia
musil@fi.muni.cz

Vojtěch Řehák
Masaryk University
Brno, Czechia
rehak@fi.muni.cz

ABSTRACT

In adversarial patrolling games, a mobile Defender strives to discover intrusions at vulnerable targets initiated by an Attacker. The Attacker's utility is traditionally defined as the probability of completing an attack, possibly weighted by target costs. However, in many real-world scenarios, the actual damage caused by the Attacker depends on the *time* elapsed since the attack's initiation to its detection. We introduce a formal model for such scenarios, and we show that the Defender always has an *optimal* strategy achieving maximal protection. We also prove that *finite-memory* Defender's strategies are sufficient for achieving protection arbitrarily close to the optimum. Then, we design an efficient *strategy synthesis* algorithm based on differentiable programming and gradient descent. We evaluate the efficiency of our method experimentally.

KEYWORDS

Strategy synthesis; Security Games; Adversarial Patrolling

ACM Reference Format:

David Klaška, Antonín Kučera, Vít Musil, and Vojtěch Řehák. 2022. Minimizing Expected Intrusion Detection Time in Adversarial Patrolling: Extended Abstract. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 3 pages.

1 INTRODUCTION

Patrolling games are a special type of security games [16] where a mobile Defender moves among protected targets with the aim of detecting possible incidents. Compared with static monitoring facilities, patrolling is more flexible and less costly on implementation and maintenance. Due to these advantages [19], patrolling is indispensable in detecting crimes [6, 8], managing disasters [13], wildlife protection [17, 18], etc. Apart from human Defenders (police squads, rangers [17], etc.) where the patrolling horizon is bounded, recent technological advances motivate the study of robotic patrolling with the *unbounded horizon.*

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online.

Most of the existing patrolling models can be classified as either *regular* or *adversarial* [3, 7, 14]. Regular patrolling is a form of surveillance where the Defender aims at discovering accidents as quickly as possible by minimizing the time lag between two consecutive visits for each target. In adversarial patrolling [1, 2, 4, 5], the Defender strives to protect the targets against an Attacker exploiting the best attack opportunities maximizing the damage. The solution concept is typically based on Stackelberg equilibrium [15, 20]. In infinite-horizon adversarial patrolling models, every target τ is assigned a finite *resilience* $d(\tau)$, and an attack at τ is discovered if the Defender visits τ in the next $d(\tau)$ time units. Although this model is adequate in many scenarios, it is not applicable when the actual damage depends on the *time elapsed since initiating the attack.* For example, if the attack involves setting a fire, punching a hole in a fuel tank, or setting a trap, then the associated damage *increases with time.* In this case, the Defender should *minimize the expected attack discovery time* rather than maximize the probability of visiting a target before a deadline.

Our main contribution can be summarized as follows:

- We propose a formal model for infinite-horizon adversarial patrolling where the damage caused by attacking a target depends on the time needed to discover the attack.
- We prove that regular strategies can achieve the same limit protection value as general strategies.
- We design an efficient algorithm synthesizing a regular Defender's strategy for a given patrolling graph, and we evaluate its functionality experimentally.

A detailed presentation of our work can be found in [11].

2 THE MODEL

Terrain model. A *patrolling graph* is a tuple $G = (V, T, E, tm, \alpha)$ where V is a finite set of *vertices* (Defender's positions), $T \subseteq V$ is a non-empty set of *targets*, $E \subseteq V \times V$ is a set of *edges* (admissible Defender's moves), $tm\colon E \to \mathbb{N}_+$ specifies the traversal time of an edge, and $\alpha\colon T \to \mathbb{R}_+$ defines the costs of targets. We require that G is strongly connected, and we write $u \to v$ instead of $(u, v) \in E$.

The sets of all non-empty finite and infinite paths in G are denoted by $\mathcal{H}$ (*histories*) and $\mathcal{W}$ (*walks*), respectively.

Defender and Attacker. A *Defender's strategy* is a function γ assigning to every history $h \in \mathcal{H}$ of Defender's moves a probability distribution on V such that $\gamma(h)(v) > 0$ only if $hv \in \mathcal{H}$, i.e., $u \to v$ where u is the last vertex of h. We also use $walk(h)$ to denote the set of all walks initiated by a given $h \in \mathcal{H}$. For every *initial vertex* v where the Defender starts patrolling, the strategy γ determines a probability space over the walks in the standard way.

The Attacker observes the history of Defender's moves and decides whether and where to initiate an attack. An *observation* is a sequence $o = v_1, \ldots, v_n, v_n \to v_{n+1}$, where $v_1, \ldots, v_n$ is a path in G. Intuitively, $v_1, \ldots, v_n$ is the sequence of vertices visited by the Defender, v_n is the currently visited vertex, and $v_n \to v_{n+1}$ is the edge taken next. The set of all observations is denoted by Ω. An *Attacker's strategy* is a function $\pi \colon \Omega \to \{wait, attack_\tau : \tau \in T\}$. We require that for every walk $w = v_1, v_2, \ldots$ there is a *unique* n such that $\pi(v_1, \ldots, v_n, v_n \to v_{n+1}) = attack_\tau$ for some target τ.

Protection value. Suppose the Defender commits to a strategy γ and the Attacker selects a strategy π. The *expected damage* caused by π against γ is the expected time to discover an attack scheduled by π weighted by target costs. More precisely, for every walk $w = v_1, v_2, \ldots$, let n be the unique index where $\pi(v_1, \ldots, v_n, v_n \to v_{n+1}) = attack_\tau$ for some $\tau \in T$, and let $m > n$ be the least index such that $v_m = \tau$ (if no such index exists, then $m = \infty$). Furthermore, we put $\mathcal{D}^\pi(w) = \alpha(\tau) \cdot \sum_{i=n}^{m} tm(v_i, v_{i+1})$.

The expected damage caused by π against γ initiated in v is defined as the expected value of $\mathcal{D}^\pi$ in the probability space over the walks determined by γ and v, denoted by $\mathbb{E}^{\gamma,v}[\mathcal{D}^\pi]$. Since the Defender may choose the initial vertex v, we define the *protection value achieved by* γ and the *limit protection value* as follows:

$$\mathrm{Val}(\gamma) \;=\; \min_{v} \sup_{\pi}\ \mathbb{E}^{\gamma,v}[\mathcal{D}^\pi] \qquad\qquad \mathrm{Val} \;=\; \inf_{\gamma}\ \mathrm{Val}(\gamma)$$

We say that a Defender's strategy γ is *optimal* if $\mathrm{Val}(\gamma) = \mathrm{Val}$.

3 FINITE-MEMORY DEFENDER'S STRATEGIES

A general Defender's strategy depends on the whole history of moves and cannot be finitely represented. A computationally feasible subclass are *finite-memory* (or *regular*) strategies [9, 10, 12] where the relevant information about the history is represented by finitely memory elements assigned to each vertex.

Formally, let $\mathrm{mem} \colon V \to \mathbb{N}$ be a function assigning to every vertex the number of *memory elements*. The set of *augmented vertices* is defined by $\widehat{V} = \{(v, m) \colon v \in V,\ 1 \le m \le \mathrm{mem}(v)\}$. We use $\widehat{v}$ to denote an augmented vertex of the form (v, m) where $m \le \mathrm{mem}(v)$. A *regular* Defender's strategy for G is a function $\sigma \colon \widehat{V} \to Dist(\widehat{V})$ where $\sigma(v, m)(v', m') > 0$ only if $v \to v'$. We say that σ is *unambiguous* if for all $v, v' \in V$ and $m \le \mathrm{mem}(v)$ there is at most one m' such that $\sigma(v, m)(v', m') > 0$.

Intuitively, the Defender starts patrolling in a designated *initial vertex* v with *initial memory element* m, and then traverses the vertices of G and updates the memory according to σ.

An important question is whether regular strategies can achieve the same limit protection value as general strategies. The answer is positive, and it is proven in two steps. First, we show that there exists an *optimal* Defender's strategy γ satisfying $\mathrm{Val}(\gamma) = \mathrm{Val}$. Then, for arbitrarily small $\varepsilon > 0$, we demonstrate the existence of a regular strategy σ such that $\mathrm{Val}(\sigma) \le \mathrm{Val}(\gamma) + \varepsilon$. Proofs can be found in [11].

Theorem 3.1. *For every patrolling graph, there exists a Defender's strategy* γ *such that* $\mathrm{Val}(\gamma) = \mathrm{Val}$.

Theorem 3.2. *Let G be a patrolling graph, and let Reg be the class of all regular strategies for G. Then* $\inf_{\sigma \in Reg} \mathrm{Val}(\sigma) = \mathrm{Val}$.

4 STRATEGY SYNTHESIS ALGORITHM

Let G be a patrolling graph and σ a regular strategy for G. First, we show how to compute $\mathrm{Val}(\sigma)$.

Let $\widehat{E}$ be the set of all $(\widehat{u}, \widehat{v}) \in \widehat{V} \times \widehat{V}$ such that $\sigma(\widehat{u})(\widehat{v}) > 0$, i.e., $\widehat{E}$ is the set of *augmented edges* used by σ. For every target τ, let $\pi[\tau]$ be the Attacker strategy where for all $(u, v) \in E$ we have that $\pi[\tau](u, u \to v) = attack_\tau$, i.e., $\pi[\tau]$ attacks τ immediately after the Defender starts its walk.

For every $\widehat{e} = (\widehat{u}, \widehat{v}) \in \widehat{E}$ and $\tau \in T$, let $\mathcal{L}_{\tau,\widehat{e}}$ be the expected damage caused by an attack at τ scheduled right after the Defender starts traversing $\widehat{e}$, i.e.,

$$\mathcal{L}_{\tau,\widehat{e}} \quad = \quad \mathbb{E}^{\sigma,\widehat{u}}\big[\mathcal{D}^{\pi[\tau]} \mid walk(\widehat{e})\big]\,.$$

Hence, $\mathcal{L}_{\tau,\widehat{e}}$ is the conditional expected value of $\mathcal{D}^{\pi[\tau]}$ under the condition that the Defender's walk starts by traversing $\widehat{e}$.

Consider the directed graph $\widehat{G} = (\widehat{V}, \widehat{E})$, and let $\mathcal{B}$ denote the set of all *bottom* strongly connected components of $\widehat{G}$. Let

$$\mathcal{L}(\sigma) \quad = \quad \min_{B \in \mathcal{B}}\ \max_{\tau \in T}\ \max_{\widehat{e} \in E(B)}\ \mathcal{L}_{\tau,\widehat{e}}$$

where $E(B) = \widehat{E} \cap (B \times B)$ is the set of augmented edges in the component B used by σ. We have the following:

Theorem 4.1. *Let σ be a regular strategy for a patrolling graph G. Then* $\mathrm{Val}(\sigma) \le \mathcal{L}(\sigma)$. *If σ is unambiguous, then* $\mathrm{Val}(\sigma) = \mathcal{L}(\sigma)$.

A proof of Theorem 4.1 can be found in [11]. Our strategy synthesis algorithm is based on interpreting $\mathcal{L}$ as a piecewise differentiable function and applying methods of differentiable programming. We start from a random strategy σ, repeatedly compute $\mathcal{L}(\sigma)$ and update the strategy against the direction of its gradient. This is repeated many times and the algorithm returns the best strategy found. A detailed description of the optimization scheme is given in [11], together with two sets of experiments on graphs with increasing sizes focusing on runtime analysis and the achieved protection values.

ACKNOWLEDGEMENTS

Research was sponsored by the Army Research Office and was accomplished under Grant Number W911NF-21-1-0189.

Vít Musil is also supported from Operational Programme Research, Development and Education – Project Postdoc2MUNI (No. CZ.02.2.69/0.0/0.0/18_053/0016952).

Disclaimer. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

REFERENCES

[1] N. Agmon, S. Kraus, and G. Kaminka. 2008. Multi-Robot Perimeter Patrol in Adversarial Settings. In *Proceedings of ICRA 2008*. IEEE Computer Society Press, 2339–2345.

[2] N. Agmon, V. Sadov, G.A. Kaminka, and S. Kraus. 2008. The impact of adversarial knowledge on adversarial planning in perimeter patrol. In *Proceedings of AAMAS 2008*. 55–62.

[3] A. Almeida, G. Ramalho, H. Santana, P. Tedesco, T. Menezes, V. Corruble, and Y. Chevaleyr. 2004. Recent Advances on Multi-Agent Patrolling. *Advances in Artificial Intelligence – SBIA* 3171 (2004), 474–483.

[4] N. Basilico, N. Gatti, and F. Amigoni. 2009. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *Proceedings of AAMAS 2009*. 57–64.

[5] N. Basilico, N. Gatti, and F. Amigoni. 2012. Patrolling Security Games: Definitions and Algorithms for Solving Large Instances with Single Patroller and Single Intruder. *Artificial Inteligence* 184–185 (2012), 78–123.

[6] H. Chen, T. Cheng, and S. Wise. 2017. Developing an Online Cooperative Police Patrol Routing Strategy. *Computers, Environment and Urban Systems* 62 (2017), 19–29.

[7] L. Huang, M. Zhou, K. Hao, and E. Hou. 2019. A Survey of Multi-robot Regular and Adversarial Patrolling. *IEEE/CAA Journal of Automatica Sinica* 6, 4 (2019), 894–903.

[8] M. Jakob, O. Vanek, and M. Pechoucek. 2011. Using Agents to Improve International MaritimeTransport Security. *IEEE Intelligent Systems* 26, 1 (2011), 90–96.

[9] D. Klaška, A. Kučera, T. Lamser, and V. Řehák. 2018. Automatic Synthesis of Efficient Regular Strategies in Adversarial Patrolling Games. In *Proceedings of AAMAS 2018*. 659–666.

[10] D. Klaška, A. Kučera, V. Musil, and V. Řehák. 2021. Regstar: Efficient Strategy Synthesis for Adversarial Patrolling Games. In *Proceedings of UAI 2021*.

[11] D. Klaška, A. Kučera, V. Musil, and V. Řehák. 2022. Minimizing Expected Intrusion Detection Time in Adversarial Patrolling. arXiv:2202.01095 [cs.MA]

[12] A. Kučera and T. Lamser. 2016. Regular Strategies and Strategy Improvement: Efficient Tools for Solving Large Patrolling Problems. In *Proceedings of AAMAS 2016*. 1171–1179.

[13] I. Maza, F. Caballero, J. Capitán, J.R. Martínez de Dios, and A. Ollero. 2011. Experimental Results in Multi-UAV Coordination for Disaster Management and Civil Security Applications. *Journal of Intelligent and Robotic Systems* 61, 1–4 (2011), 563–585.

[14] D. Portugal and R. Rocha. 2011. A Survey on Multi-Robot Patrolling Algorithms. *Technological Innovation for Sustainability* 349 (2011), 139–146.

[15] A. Sinha, F. Fang, B. An, C. Kiekintveld, and M. Tambe. 2018. Stackelberg Security Games: Looking Beyond a Decade of Success. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2018)*. 5494–5501.

[16] M. Tambe. 2011. *Security and Game Theory. Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press.

[17] Y. Wang, Z.R. Shi, L. Yu, Y. Wu, R. Singh, L. Joppa, and F. Fang. 2019. Deep Reinforcement Learning for Green Security Games with Real-Time Information. In *Proceedings of AAAI 2019*. 1401–1408.

[18] L. Xu. 2021. Learning and Planning Under Uncertainty for Green Security. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2021)*.

[19] Z. Yan, N. Jouandeau, and A.A. Cherif. 2013. A Survey and Analysis of Multi-Robot Coordination. *International Journal of Advanced Robotic Systems* 10, 12 (2013), 1–18.

[20] Z. Yin, D. Korzhyk, C. Kiekintveld, V. Conitzer, and M. Tambe. 2010. Stackelberg vs. Nash in security games: Interchangeability, equivalence, and uniqueness. In *Proceedings of AAMAS 2010*. 1139–1146.