Regular Strategies and Strategy Improvement: Efficient Tools for Solving Large Patrolling Problems

Antonín Kučera*
Faculty of Informatics, Masaryk University
Botanická 68a, 60200 Brno
Czech Republic
kucera@fi.muni.cz

Tomáš Lamser
Faculty of Informatics, Masaryk University
Botanická 68a, 60200 Brno
Czech Republic

ABSTRACT

In patrolling problems, the task is to compute an optimal strategy for a patroller who moves among vulnerable targets and aims at detecting possible intrusions. Previous approaches to this problem were mostly based on non-linear programming, and the solution space was restricted to positional strategies or to strategies dependent on a bounded history of patroller's moves. In this paper, we extend the solution space to *regular* strategies, and show that regular strategies are *strictly more powerful* than strategies dependent on a bounded history. Further, we design a *strategy improvement* technique for regular strategies which completely avoids the use of non-linear programming. Intuitively, we start with *some* regular strategy, and then repeatedly *improve* this strategy by incorporating a solution of a certain *linear* program. Our experiments demonstrate that the proposed approach can quickly produce strategies of very good quality even for quite large patrolling problems.

Keywords

Robotic patrolling; Security of Agent Systems; Stackelberg Equilibrium; Strategy Synthesis

1. INTRODUCTION

Game theoretic approaches to patrolling problems have received a considerable amount of attention in recent years (see, e.g., [15, 4] for an everview). The task is to design an optimal strategy (policy) for a patroller who moves among a given set of vulnerable targets and aims at detecting possible intrusions. Since the intruder can observe the moves of the patroller but the patroller has no knowledge about the plans and actions of the intruder, the solution concepts used in this setting are mostly based on Stackelberg equilibrium [17]. That is, the patroller's strategy for visiting the targets should maximize his expected utility against all best responses of the intruder.

Most of the existing algorithms for solving patrolling problems (see Section 2) are based on mathematical programming. This approach has to deal with two fundamental difficulties: First, some of the constraints are (inevitably) *non-linear*. Second, the encoding is usually restricted to *positional* patroller's strategies, where the decision depends only on the current patroller's position. That is, the solution encoded by the program is actually optimal only among the subset of positional strategies. It has been argued in, e.g., [8, 5], that the patroller can achieve a better expected utility when his decisions depend on the last K visited positions, where $K \geq 1$ is a suitable constant (such strategies are called *last-K strategies* in this paper). Although one can modify the programs constructed for positional strategies so that they work for *last-K* strategies, the size of the modified programs grows *exponentially* in K. Even for small instances (say, 10 targets and $K = 3$), one may thus obtain non-linear programs with more than 1000 variables. Hence, this approach does not scale with increasing K.

Our contribution. We consider a special class of *regular* patroller's strategies that utilize deterministic finite-state automata to collect some information about the history of patroller's moves. The patroller's decision after a history $v_1, \ldots, v_n$ of visited positions then depends not only on the current position v_n, but also on the state of the automaton entered after reading the word $v_1 \cdots v_n$. We show that regular strategies are *strictly more powerful* (i.e., can achieve a strictly better expected utility for the patroller) than all strategies of $\bigcup_{K=1}^{\infty}$ *last-K*. Further, we design a *strategy improvement* technique for regular strategies which completely avoids the use of non-linear programming. Intuitively, we start with *some* regular strategy, and then try to *improve* this strategy by performing a given number of *rounds*, where each round consists of three steps: (1) we construct a linear program whose coefficients depend on the regular strategy computed in the previous round. The size of this program is *linear* in the size of the patrolling problem and the size of the automaton used by the regular strategy; (2) we solve the linear program; (3) we construct a new regular strategy by taking a suitable "combination" of the strategy from the previous round and the solution of the linear program. Compared to the existing methods, our approach is applicable to patrolling problems of considerably larger size, which is witnessed by the experiments presented in Section 5.

2. RELATED WORK

Existing approaches to protecting the targets against intrusions are based either on optimizing static allocation of available resources to the targets in order to discover the

*Supported by the Czech Science Foundation, grant No. 15-17564S.

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

intruder, or by computing an optimal movement strategy for a mobile patroller. The first model, also known as *security games*, has been studied in [11, 12]. The earlier works [14] concentrated on finding an allocation that minimizes the chance for penetrating an unprotected target. Later, the main task was extended with a requirement that the allocation needs to satisfy some extra constraints [16].

In the second model, also known as *patrolling games*, the focus was primarily on finding locally optimal strategies for robotic patrolling units either on restricted graphs (e.g., on circles in [2, 3]), or arbitrary graphs with weighted preference on the targets [5, 6]. Alternatively, the work focused on some novel aspects of the problem, such as variants with moving targets [8, 10], multiple patrolling units [7], or movement of the intruder on the graph [6] and reaction to alarms [13]. Most of the existing literature assumes that the patroller is following a memoryless (Markov) strategy that depends solely on the current position of the patroller in the graph and they seek for a solution using mathematical programming. Few exceptions include duplicating each node of the graph to distinguish internal states of the patroller (e.g., in [2] authors consider a direction of the patrolling robot as a specific state; in [9], this concept is further generalized), or seeking for higher-order strategies in [5]. In [1], an algorithm for computing an ε-optimal strategy for the patroller is designed, but this algorithm is of exponential complexity.

The model considered in this paper is the same as in [5]. That is, we allow for an arbitrary topology of the game graph, we assume that the time needed to complete an intrusion can be different at each target, and the importance of the individual targets can be different for the patroller and the intruder.

3. REGULAR STRATEGIES

A *patrolling problem* is a tuple $\mathcal{G} = (V, E, T, d)$, where V is a finite set of *vertices* (patroller's positions), $E \subseteq V \times V$ are admissible moves, $T \subseteq V$ is a set of *targets*, and d assigns to every $t \in T$ the number $d(t) > 0$ of time units needed to complete the intrusion at t. We require that the directed graph (V, E) is strongly connected. It can be safely assumed that the patroller spends one unit of time in each vertex, because longer stays can be modeled by inserting auxiliary vertices and edges. A *history* is a finite path in $\mathcal{G}$, and a *walk* is an infinite path in $\mathcal{G}$. Note that a history h can also be seen as a (non-empty) finite word over the alphabet V. The set of all histories is denoted by $\mathcal{H}$.

A *finite-state observer for* $\mathcal{G} = (V, E, T, d)$ is a deterministic finite-state automaton $\mathcal{A} = (S, \delta)$, where S is a finite set of *states* and $\delta : S \times V \to S$ is a total transition function. We extend δ to histories in the expected way. Further, we require that the state space of $\mathcal{A}$ is *strongly connected*, i.e., for all $s, s' \in S$ where $s \neq s'$ there exists a history h such that $\delta(s, h) = s'$. An *$\mathcal{A}$-regular patroller's strategy* is a triple $\sigma = (\hat{v}, \hat{s}, \kappa)$, where $\hat{v} \in V$ is the *initial vertex*, $\hat{s} \in S$ is the *initial state*, and κ is a function which to every pair $(v, s) \in V \times S$ assigns a probability distribution μ over V such that $\mu(v') > 0$ only if $(v, v') \in E$. The set of all $\mathcal{A}$-regular patroller's strategies is denoted by $\Sigma_{\mathcal{A}}$. Given a history h ending in a vertex v, the next vertex is selected randomly according to the distribution $\kappa(v, s)$, where $s = \delta(\hat{s}, h)$. The state s represents the information collected by $\mathcal{A}$ after reading h. Thus, σ determines a unique probability measure $\mathcal{P}^\sigma$ over all patroller's walks in $\mathcal{G}$ initiated in $\hat{v}$.

An *intruder's strategy* is a function $\pi : \mathcal{H} \to \{wait, enter_t \mid t \in T\}$ which says whether the intruder should wait or try to penetrate a target t after observing a given history of patroller's moves. The intruder is allowed to attack at most once along a patroller's walk, which means that if $\pi(h) = enter_t$ for some $t \in T$, then $\pi(h') = wait$ for all proper prefixes h' of h. The set of all intruder's strategies is denoted by Π. Given a strategy $\pi \in \Pi$ and a patroller's walk $w = v_0, v_1, v_2, \ldots$, we say that the intruder *waits along* w if $\pi(h) = wait$ for every finite prefix h of w. If there is $j \geq 0$ such that $\pi(v_0, \ldots, v_j) = enter_t$, then we say that the intruder *enters* t *along* w, and he is either *captured* in t or *penetrates* t, depending on whether t appears among the vertices $v_j, \ldots, v_{j+d(t)-1}$ or not, respectively. Given an $\mathcal{A}$-regular patroller's strategy $\sigma = (\hat{v}, \hat{s}, \kappa)$ and an intruder's strategy π, the probability of all walks w initiated in $\hat{v}$ such the intruder waits along w, is captured in t, and penetrates t, is denoted by $\mathcal{P}^\sigma(noattack^\pi)$, $\mathcal{P}^\sigma(capture_t^\pi)$, and $\mathcal{P}^\sigma(penetrate_t^\pi)$, respectively.

In general, the patroller and the intruder may regard the importance of each outcome differently, and hence there are separate families of *utilities* $U_c^P(t)$, $U_p^P(t)$ to the patroller and $U_c^I(t)$, $U_p^I(t)$ to the intruder if the outcome is $capture_t$ and $penetrate_t$, respectively. Without restrictions, we assume that both players receive 0 for the outcome *noattack*, and require that $U_p^P(t) \leq 0 \leq U_c^P(t)$ and $U_c^I(t) \leq 0 \leq U_p^I(t)$ for every target t. Given a pair of strategies σ, π, the *expected patroller's utilility* is defined by

$$\mathrm{EU}_P^{\sigma,\pi} = \sum_{t \in T} \mathcal{P}^\sigma(capture_t^\pi) \cdot U_c^P(t) + \mathcal{P}^\sigma(penetrate_t^\pi) \cdot U_p^P(t)$$

The *expected intruder's utility* $\mathrm{EU}_I^{\sigma,\pi}$ is defined in the same way, but $U_c^I(t)$, $U_p^I(t)$ are used instead of $U_c^P(t)$, $U_p^P(t)$.

For each $\sigma \in \Sigma_{\mathcal{A}}$, we define the set $\mathrm{BR}(\sigma)$ of all $\pi^* \in \Pi$ such that $\mathrm{EU}_I^{\sigma,\pi^*} = \sup_{\pi \in \Pi} \mathrm{EU}_I^{\sigma,\pi}$, i.e., π^* is a *best response to* σ. A pair of strategies $(\sigma^*, \pi^*) \in \Sigma_{\mathcal{A}} \times \Pi$ is a *Stackelberg equilibrium* if the following conditions are satisfied:

(a) π^* is a best response to σ^*;

(b) π^* maximizes the expected patroller's utility among all best responses, i.e., $\mathrm{EU}_P^{\sigma^*,\pi^*} = \sup_{\pi \in \mathrm{BR}(\sigma)} \mathrm{EU}_P^{\sigma^*,\pi}$;

(c) the patroller is not motivated to change his strategy, i.e., $\mathrm{EU}_P^{\sigma^*,\pi^*} = \sup_{\sigma \in \Sigma_{\mathcal{A}}} \sup_{\pi \in \mathrm{BR}(\sigma)} \mathrm{EU}_P^{\sigma,\pi}$.

THEOREM 1. *For every patrolling problem $\mathcal{G}$, every finite-state observer $\mathcal{A}$, and all $U_c^P, U_p^P, U_c^I, U_p^I$, there exists a Stackelberg equilibrium.*

PROOF SKETCH. First we show that for every $\sigma \in \Sigma_{\mathcal{A}}$, the set $\mathrm{BR}(\sigma)$ is non-empty. The argument crucially depends on the regularity of σ, which guarantees that the patroller plays in one of the finitely many ways after re-visiting a given vertex. Then, we show that there exists a strategy $\pi^* \in \mathrm{BR}(\sigma)$ which maximizes the expected patroller's utility among all best responses. Finally, we prove there exists at least one $\sigma^* \in \Sigma_{\mathcal{A}}$ satisfying the condition (c) above. This is achieved by considering an infinite sequence $\sigma_0, \sigma_1, \ldots$ of strategies such that $\sup_{\pi \in \mathrm{BR}(\sigma_i)} \mathrm{EU}_P^{\sigma_i,\pi}$ converges monotonically to $\sup_{\sigma \in \Sigma_{\mathcal{A}}} \sup_{\pi \in \mathrm{BR}(\sigma)} \mathrm{EU}_P^{\sigma,\pi}$ as $i \to \infty$, and shoving that σ^* can be chosen as the limit of a converging subsequence of $\sigma_0, \sigma_1, \ldots$ □

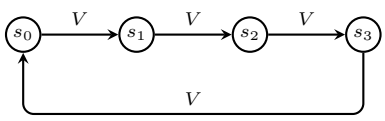

Figure 1: A finite-state observer $\mathcal{A}$.

Due to Theorem 1, we can also define the *Stackelberg value* achievable by $\mathcal{A}$-regular strategies in $\mathcal{G}$ by

$$Val(\mathcal{G},\mathcal{A}) \;=\; \mathrm{EU}_P^{\sigma^*,\pi^*}$$

where (σ^*,π^*) is a Stackelberg equilibrium. Further, for every $\sigma^* \in \Sigma_{\mathcal{A}}$, we define the *value of σ^* in $\mathcal{G}$* by

$$Val(\sigma^*,\mathcal{G}) \;=\; \sup_{\pi\in \mathrm{BR}(\sigma)} \mathrm{EU}_P^{\sigma^*,\pi}.$$

We say that $\sigma^* \in \Sigma_{\mathcal{A}}$ is *$\mathcal{A}$-optimal* if $Val(\sigma^*,\mathcal{G}) = Val(\mathcal{G},\mathcal{A})$.

In the special case of *zero-sum* patrolling problems, where the utilities also satisfy $U_c^P(t)+U_c^I(t)=U_p^P(t)+U_p^I(t)=0$ for all $t\in T$, the intruder maximizes his expected utility iff he minimizes the expected patroller's utility. Further, every $\pi \in \mathrm{BR}(\sigma)$ achieves the same (and hence the maximal) expected patroller's utility. Thus, for zero-sum patrolling problems we obtain that

$$Val(\sigma^*,\mathcal{G}) \;=\; \inf_{\pi\in\Pi} \mathrm{EU}_P^{\sigma^*,\pi}$$
$$Val(\mathcal{G},\mathcal{A}) \;=\; \sup_{\sigma\in\Sigma_{\mathcal{A}}} \inf_{\pi\in\Pi} \mathrm{EU}_P^{\sigma,\pi}$$

Example 1. *To get some intuition about the efficiency of regular strategies, consider a (zero-sum) patrolling problem $\mathcal{G}$ where*

- $V=\{v_1,\ldots,v_{30}\}$
- $E = V\times V,\quad T = V,$
- $d(v_i)=3$ *for all* $1\le i\le 10$, *and* $d(v_j)=5$ *for all* $11\le j\le 30$,
- $U_c^P(t)=U_c^I(t)=0$, $U_p^P(t)=-1$, $U_p^I(t)=1$ *for all* t.

Consider a finite-state observer $\mathcal{A}$ of Fig. 1. Further, let V_0,V_1 be a partition of $\{v_1,\ldots,v_{10}\}$ into two subsets of 5 elements, and U_0,U_1,U_2,U_3 be a partition of $\{v_{11},\ldots,v_{30}\}$ into four subsets of 5 elements. Also observe that for all $(\sigma,\pi)\in\Sigma_{\mathcal{A}}\times\Pi$, we have that $1+\mathrm{EU}_P^{\sigma,\pi}$ is the probability of all walks where the intruder is captured or waits forever. Hence, the patroller/intruder actually aims at maximizing/minimizing this probability.

Let $\sigma=(v_1,s_3,\kappa)$ be an $\mathcal{A}$-regular strategy such that $\kappa(v,s_i)$ is a uniform distribution over the set $V_j\cup U_k$, where $j=(i+1) \bmod 2$ and $k=(i+1) \bmod 4$. Then, one can easily confirm that $Val(\sigma,\mathcal{G})=-9/10$. That is, if the patroller uses the strategy σ, the probability of all walks where the intruder is either captured or waits forever is at least $1/10$ (for every intruder's strategy). Intutively, this seems quite close to the optimum, because in the first 4 moves, each vertex of $\{v_1,v_{11},\ldots,v_{30}\}$ should be visited at least once, and each vertex of $\{v_2,\ldots,v_{10}\}$ should be visited "about twice on average". Hence, the probability of all walks where the intruder is captured or waits forever is bounded by a number close to $4/(21+2\cdot 9)=4/39$. The "periodicity" of σ captured by $\mathcal{A}$ is essential for achieving the efficiency of σ, and it cannot be mimicked by any strategy which depends just on some finite suffix of the history.

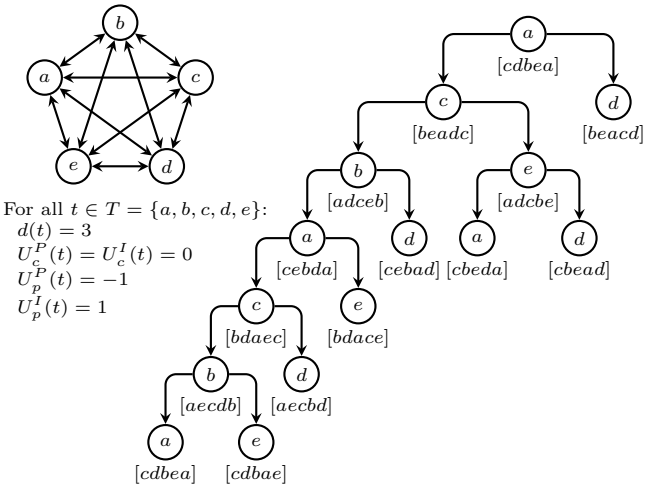

Figure 2: A zero-sum patrolling problem.

Now we give a rigorous proof that regular strategies can be strictly better than all strategies of $\bigcup_{K=1}^{\infty}$ *last-K* (see Section 1), even for zero-sum patrolling problems with fully connected environment where all vertices are targets, $d(t)$ is the same for all targets, and the patroller/defender aims just at maximizing/minimizing the probability of all walks where the intruder is either captured or waits forever. First, observe that *last-K* strategies can be seen as $\mathcal{A}_K$-regular strategies, where the finite-state observer $\mathcal{A}_K$ "remembers" the sequence of the last K visited vertices in its state-space (hence, $\mathcal{A}_K$ needs $(|V|+1)^K$ control states). We have the following:

Theorem 2. *Let $\mathcal{G}$ be the zero-sum patrolling problem of Fig. 2 (left). Then there exists a finite-state observer $\mathcal{A}$ such that $Val(\mathcal{G},\mathcal{A}) > Val(\mathcal{G},\mathcal{A}_K)$ for all $K\ge 1$.*

Proof Sketch. The set of states of $\mathcal{A}$ are all permutations of $\{a,b,c,d,e\}$ which encode the information about the current vertex, and about two previously visited vertices and their immediate successors. For example, the permutation $[cebda]$ says that the current vertex is a, the vertex visited before a was b and the other successor of b was d, and the vertex visited before b was c and the other successor of c was e. The transition function of $\mathcal{A}$ is defined as follows. For every state (permutation) $[x_1,x_2,x_3,x_4,x_5]$ we put $\delta([x_1,x_2,x_3,x_4,x_5],x_1) = [x_3,x_4,x_5,x_2,x_1])$ and $\delta([x_1,x_2,x_3,x_4,x_5],x_2)=[x_3,x_4,x_5,x_1,x_2])$. For the other y such that $x_1\neq y\neq x_2$ we define $\delta([x_1,x_2,x_3,x_4,x_5],y)$ arbitrarily. Further, for every vertex x and every state of the form $[y_1,y_2,y_3,y_4,x]$ we put $\kappa(x,[y_1,y_2,y_3,y_4,x]) = \mu(y_1,y_2)$, where $\mu(y_1,y_2)$ selects between y_1 and y_2 with probability $1/2$. For the other arguments, κ is defined arbitrarily. Let us consider an $\mathcal{A}$-regular strategy $\sigma^* = (a,[aecdb],\kappa)$. A fragment of this strategy is shown in Fig. 2 (right), together with the states of $\mathcal{A}$ entered after reading the history associated to the nodes. One can easily verify that $\inf_{\pi\in\Pi}\mathrm{EU}_P^{\sigma^*,\pi}=-1/2$. That is, if the patroller uses the strategy σ, the probability of all walks where the intruder is either captured or waits forever is at least $1/2$ (for every intruder's strategy). Further, for *every* finite-state observer $\mathcal{B}$ and *every* $\mathcal{B}$-regular strategy σ we have that this probability is at most $1/2$, because σ is obliged to

visit four vertices in at most two steps, which means that some of them is inevitably visited with probability at most 1/2. Hence, $Val(\mathcal{G}, \mathcal{A}) = -1/2$.

Now we prove that for an arbitrarily large $K \geq 1$ we have that $Val(\mathcal{G}, \mathcal{A}_K) < -1/2$. For the sake of contradiction, assume that $Val(\mathcal{G}, \mathcal{A}_K) \geq -1/2$, which implies that $Val(\mathcal{G}, \mathcal{A}_K) = -1/2$ by using the same argument as above. The core of the argument reveals that every optimal $\mathcal{A}_K$-regular strategy must, from certain point on, behave identically as the $\mathcal{A}$-regular strategy σ constructed above. This is because whenever a vertex x is visited, each of the four remaining vertices must be visited *exactly* with probability 1/2 in the next two steps. This very soon enforces the regularity captured by $\mathcal{A}$. However, $\mathcal{A}_K$-regular strategies cannot behave like the strategy σ. Consider the leftmost branch in the tree of Fig. 2 (right). This branch is an infinite sequence of vertices obtained by repeating the sequence *acb* infinitely often. The behaviour of σ along this branch depends on the length of the history modulo 6, which cannot be deduced from an arbitrarily long suffix of the history. □

4. STRATEGY IMPROVEMENT

A Stackelberg equilibrium for a given patrolling problem and a given finite-state observer is hard to compute in general. In this section, we consider a slightly relaxed variant of this task: Compute a pair of strategies (σ^*, π^*) such that

(a) π^* is a best response to σ^*;

(b) π^* maximizes the expected patroller's utility among all best responses to σ^*;

(c) $\mathrm{EU}_P^{\sigma^*,\pi^*}$ is as large as possible.

Hence, the intruder is still not motivated to change his strategy, and the patroller may still be satisfied with his expected utility.

Our approach to computing (σ^*, π^*) is based on *strategy improvement*. Intuitively, we start with *some* regular strategy for a given patrolling problem, and then improve this strategy in a finite number of rounds by repeatedly constructing and solving a "small" linear program. In the end, we obtain a pair (σ^*, π^*) satisfying the above requirement, together with the associated $\mathrm{EU}_P^{\sigma^*,\pi^*}$.

For the rest of this section, we fix a patrolling problem $\mathcal{G} = (V, E, T, d)$, a finite-state observer $\mathcal{A} = (S, \delta)$, and utility functions $U_c^P, U_p^P, U_c^I, U_p^I$.

4.1 Zero-Sum Patrolling Problems

For simplicity, we first assume that our patrolling problem is *zero-sum*, i.e., $U_c^P(t) + U_c^I(t) = U_p^P(t) + U_p^I(t) = 0$ for all $t \in T$. General utility functions are discussed in Section 4.2.

Before describing the algorithm, we need to introduce further notions. Let $C \subseteq V \times S$. A *selector for* C is a function $sel : C \to 2^V$ such that, for every $(v, s) \in C$, the set $sel(v, s)$ is non-empty and contains only vertices that are immediate successors of v (i.e., if $v' \in sel(v, s)$, then $(v, v') \in E$). A *consistent path* in (C, sel) is a finite sequence $(v_0, s_0), \ldots, (v_n, s_n)$ such that $(v_i, s_i) \in C$ for all $i \leq n$, and $v_{i+1} \in sel(v_i, s_i)$, $s_{i+1} = \delta(s_i, v_{i+1})$ for all $i < n$. We say that (C, sel) is an *end component* if the following conditions are satisfied:

- C is closed under sel, i.e., if $(v, s) \in C$ and $v' \in sel(v, s)$, then $(v', s') \in C$, where $s' = \delta(s, v')$.
- For all $(v, s), (v', s') \in C$, there is a finite consistent path from (v, s) to (v', s') in (C, sel).

Let $\sigma = (\hat{v}, \hat{s}, \kappa)$ be an $\mathcal{A}$-regular strategy. Then κ determines a selector sel_κ for $V \times S$ where $sel_\kappa(v, s)$ consists of all $v' \in V$ such that $\kappa(v, s)(v') > 0$. Let C_σ be the set of all $(v, s) \in V \times S$ which can be reached from $(\hat{v}, \delta(\hat{s}, \hat{v}))$ via a consistent path in $(V \times S, sel_\kappa)$. Further, let sel_σ be a selector for C_σ obtained by restricting sel_κ to C_σ. We say that σ *stays* in an end component (C, sel) if $C_\sigma \subseteq C$ and $sel_\sigma(v, s) \subseteq sel(v, s)$ for all $(v, s) \in C$.

THEOREM 3. *For every $\mathcal{A}$-regular strategy $\sigma = (\hat{v}, \hat{s}, \kappa)$ there exist $\bar{v} \in V$ and $\bar{s} \in S$ such that the $\mathcal{A}$-regular strategy $\bar{\sigma} = (\bar{v}, \bar{s}, \kappa)$ satisfies $Val(\bar{\sigma}, \mathcal{G}) \geq Val(\sigma, \mathcal{G})$ and $(C_{\bar{\sigma}}, sel_{\bar{\sigma}})$ is an end component.*

PROOF SKETCH. Let $\to_\sigma$ be a binary relation on $V \times S$ such that $(v, s) \to_\sigma (v', s')$ iff $\kappa(v, s)(v') > 0$ and $s' = \delta(s, v')$. Let D be a bottom strongly connected component (BSCC) of $(V \times S, \to_\sigma)$. For every $(v, s) \in D$ and $t \in T$, let $U_\sigma[(v, s), t]$ be the expected patroller's utility when the strategy σ is changed into (v, s', κ) where $\delta(s', v) = s$, and the intruder's strategy π satisfies $\pi(v) = enter_t$. Further, let $Val(D) = \min\{U_\sigma[(v, s), t] \mid (v, s) \in D, t \in T\}$ and let $\bar{D}$ be a BSCC of $(V \times S, \to_\sigma)$ such that $Val(\bar{D}) \geq Val(D)$ for every BSCC D of $(V \times S, \to_\sigma)$. Let $(v, s) \in \bar{D}$. We put $\bar{v} = v$ and set $\bar{s}$ so that $\delta(\bar{s}, v) = s$. One can easily verify that $Val(\bar{\sigma}, \mathcal{G}) \geq Val(\sigma, \mathcal{G})$. Further, $(C_{\bar{\sigma}}, sel_{\bar{\sigma}})$ is an end component because $\bar{D}$ is a BSCC of $(V \times S, \to_\sigma)$. □

For the rest of this subsection, we fix an *initial* $\mathcal{A}$-regular strategy $\sigma = (\hat{v}, \hat{s}, \kappa)$. Due to Theorem 3, we may safely assume that (C_σ, sel_σ) is an end component.

When improving σ, it may happen that the original selector sel_σ is extended. However, we constrain such extensions by choosing an end component (C, sel) such that (C_σ, sel_σ) is (componentwise) contained in (C, sel); the $\mathcal{A}$-regular strategy computed by our strategy improvement algorithm is then guaranteed to stay in (C, sel). This is particularly useful in situations when we want to restrict the scope of our algorithm to strategies that do not assign positive probabilities to some edges of E when $\mathcal{A}$ is in certain states. If we do not want to implement any such restrictions, we set (C, sel) to the *largest* end component subsuming (C_σ, sel_σ), which is guaranteed to exist and is easy to compute. For the rest of this section, we fix some end component (C, sel) that subsumes (C_σ, sel_σ).

For all $(v, s) \in C$ and $t \in T$, we define the set $CP[(v, s), t]$ of *t-capturing paths* consisting of all consistent paths $(v_0, s_0), \ldots, (v_n, s_n)$ in (C, sel) such that $(v_0, s_0) = (v, s)$, $v_n = t$, $n < d(t)$, and $v_i \neq t$ for all $i < n$. A *probability assignment* for (C, sel) is a function λ that assigns to each $(v, s) \in C$ a probability distribution over $sel(v, s)$. Now consider the non-linear program of Fig. 3 constructed for (C, sel). The variables $x[(v, s), v']$ encode a probability assignment for (C, sel) (constraints (1) and (2)). The auxiliary variables $c[(v, s), t]$ encode the probability of capturing the intruder entering t when the patroller visits v and $\mathcal{A}$ is in the state s after reading the current history (constraint (3)). The variable z encodes the expected utility of the patroller against the "best attack" of the intruder (assuming that the patroller plays according to the probability assignment encoded by the variables $x[(v, s), v']$).

max z
subject to:

$\forall (v,s) \in C,\ \forall t \in T:$

$$0 \leq x[(v,s),v'] \qquad \forall v' \in sel(v,s) \qquad (1)$$

$$1 = \sum_{v' \in sel(v,s)} x[(v,s),v'] \qquad (2)$$

$$c[(v,s),t] = \sum_{\substack{(v_0,s_0),\ldots,(v_n,s_n) \\ \in CP[(v,s),t]}} \prod_{i=0}^{n-1} x[(v_i,s_i),v_{i+1}] \qquad (3)$$

$$z \leq c[(v,s),t] \cdot U_c^P(t) + (1 - c[(v,s),t]) \cdot U_p^P(t) \qquad (4)$$

Figure 3: A non-linear program $\mathrm{NLP}[(C,sel)]$**.**

Solving the program of Fig. 3 is computationally difficult. Nevertheless, the program can be used to *improve* an existing probability assignment λ for (C,sel) in the following way: We construct a *linear* program $\mathrm{LP}[(C,sel),\lambda])$ by replacing the righ-hand side of the non-linear constrains (3) with

$$\sum_{\substack{(v_0,s_0),\ldots,(v_n,s_n) \\ \in CP[(v,s),t]}} \left(x[(v_{n-1},s_{n-1}),v_n] \cdot \prod_{i=0}^{n-2} \lambda(v_i,s_i)(v_{i+1}) \right).$$

That is, the variables $x[(v_i,s_i),v_{i+1}]$ for $0 \leq i \leq n-2$ are replaced with the corresponding constants of λ. By solving $\mathrm{LP}[(C,sel),\lambda])$, we obtain concrete values for all variables $x[(v,s),v']$, where $(v,s) \in C$ and $v' \in sel(v,s)$, that minimize the objective z. These values form another probability assignment for (C,sel) denoted by $Solve(\mathrm{LP}[(C,sel),\lambda])$. Now, we compute an "improved" probability assignment by taking a suitable combination $Combine(\lambda,\alpha)$ of λ and $\alpha = Solve(\mathrm{LP}[(C,sel),\lambda])$. As we shall see in Section 5, even a simple convex combination works quite well in practice; however, *Combine* can be a more complicated function in general. Our strategy improvement algorithm (i.e., Algorithm 1) performs N such rounds, starting with the unique probability assignment λ_σ satisfying the following conditions:

- $\lambda_\sigma(v,s)(v') = \kappa(v,s)(v')$ for all $(v,s) \in C_\sigma$ and $v' \in sel_\sigma(v,s)$.
- For all $(v,s) \in C \smallsetminus C_\sigma$, $\lambda_\sigma(v,s)$ is the uniform distribution on $sel(v,s)$.

Let λ be the probability assignment for (C,sel) obtained after completing the **repeat-until** loop at lines 3–7 of Algorithm 1. For every pair $(v,s) \in C$, the assignment λ determines an $\mathcal{A}$-regular strategy $\sigma[(v,s),\lambda] = (v,\bar{s},\bar{\lambda})$, where $\bar{s} \in S$ is a state satisfying $\delta(\bar{s},v) = s$ and $\bar{\lambda}$ is (some) extension of λ to all elements of $V \times S$ (since (C,sel) is an end component, the choice of $\bar{\lambda}(v',s')$ for $(v',s') \notin C$ is actually irrelevant). One may be tempted to think that $Val(\sigma[(v,s),\lambda],\mathcal{G})$ is the same for all $(v,s) \in C$. However, this is generally not true, and we need to invest some extra computational effort to identify an "optimal" pair (v,s) and compute the associated $Val(\sigma[(v,s),\lambda],\mathcal{G})$. This is achieved by the procedure $Evaluate(\lambda)$ at line 8 of Algorithm 1, which is described in the next paragraphs.

Let $\rightarrow_\lambda \subseteq C \times C$ be a relation defined by $(v,s) \rightarrow_\lambda (v',s')$ iff $\lambda(v,s)(v') > 0$ and $s' = \delta(s,v')$. Further, let $D \subseteq C$ be a bottom strongly connected component (BSCC) of the directed graph $(C,\rightarrow_\lambda)$. For every $(v,s) \in D$ and $t \in T$, let $c_\lambda[(v,s),t]$ be the probability of capturing the intruder entering t when the patroller visits v and $\mathcal{A}$ is in the state s after reading the current history. That is,

$$c_\lambda[(v,s),t] = \sum_{\substack{(v_0,s_0),\ldots,(v_n,s_n) \\ \in CP[(v,s),t]}} \prod_{i=0}^{n-1} \lambda(v_i,s_i)(v_{i+1}).$$

We also use $U_\lambda[(v,s),t]$ to denote the associated expected patroller's utility, i.e.,

$$U_\lambda[(v,s),t] = c_\lambda[(v,s),t] \cdot U_c^P(t) + (1 - c_\lambda[(v,s),t]) \cdot U_p^P(t).$$

Finally, we put

$$enter_D = \min\{U_\lambda[(v,s),t] \mid (v,s) \in D, t \in T\}$$

and $Val(D) = \min\{0, enter_D\}$. We claim that for all $(v,s) \in D$ we have that $Val(\sigma[(v,s),\lambda],\mathcal{G}) = Val(D)$. To see this, first realize that if $enter_D > 0$, then the optimal strategy for the intruder against $\sigma[(v,s),\lambda]$ is to *wait* forever, because otherwise his expected utility only decreases. Hence, $Val(\sigma[(v,s),\lambda],\mathcal{G}) = 0$ in this case. Otherwise, the intruder should *wait* until the patroller enters a vertex v so that $\mathcal{A}$ is in a state s where $U_\lambda[(v,s),t] = enter_D$ for some $\in T$, and then *enter* the target t. Such a situation must eventually occur with probability one, because D is a BSCC of $(C,\rightarrow_\lambda)$. Thus, the intruder achieves his (optimal) expected utility $-Val(D)$.

Let D be a BSCC of $(C,\rightarrow_\lambda)$ such that $Val(D) \geq Val(D')$ for every BSCC D' of $(C,\rightarrow_\lambda)$. Then, for every $(v,s) \in C$ we clearly have that $Val(\sigma[(v,s),\lambda],\mathcal{G}) \leq Val(D)$. Hence, the patroller should stick to $\sigma[(v,s),\lambda]$ where $(v,s) \in D$, and the value of this strategy is equal to $Val(D)$. The procedure $Evaluate(\lambda)$ returns $(\sigma[(v,s),\lambda],\pi)$, $Val(D)$, where π is the intruder's strategy achieving the expected utility $-Val(D)$ described in the previous paragraph.

Algorithm 1: Strategy improvement for $\mathcal{G}$, $\mathcal{A}$, U_c^P, U_p^P

input : σ, (C,sel), N
output : (σ^*,π^*), $Val(\sigma^*,\mathcal{G})$

1 $\lambda \leftarrow \lambda_\sigma$
2 $i \leftarrow 0$
3 **repeat**
4 $\quad \alpha \leftarrow Solve(\mathrm{LP}[(C,sel),\lambda])$
5 $\quad \lambda \leftarrow Combine(\lambda,\alpha)$
6 $\quad i \leftarrow i+1$
7 **until** $i = N$
8 (σ^*,π^*), $Val(\sigma^*,\mathcal{G}) \leftarrow Evaluate(\lambda)$

4.2 Non-Zero-Sum Patrolling Problems

In this subsection, we show how to extend the results of Section 4.1 to non-zero-sum patrolling problems. Recall that our aim is to compute a pair of strategies (σ^*,π^*) such that π^* is a best response to σ^* maximizing the expected patroller's utility among all best responses, and the expected utility of the patroller is as large as possible.

Here we restrict ourselves to a subset of $\mathcal{A}$-regular strategies σ such that (C_σ, sel_σ) is an end component, because

max $u^P[(v^*,s^*),t^*]$, where $((v^*,s^*),t^*)$ is a fixed element of R.
subject to:

$\forall (v,s) \in C,\ \forall ((\bar{v},\bar{s}),\bar{t}) \in R,\ \forall t \in T:$

$$0 < x[(v,s),v'] \qquad \forall v' \in sel(v,s) \tag{5}$$

$$1 = \sum_{v' \in sel(v,s)} x[(v,s),v'] \tag{6}$$

$$c[(v,s),t] = \sum_{\substack{(v_0,s_0),\ldots,(v_n,s_n) \\ \in CP[(v,s),t]}} \prod_{i=0}^{n-1} x[(v_i,s_i),v_{i+1}] \tag{7}$$

$$u^I[(v,s),t] = c[(v,s),t] \cdot U_c^I(t) + (1-c[(v,s),t]) \cdot U_p^I(t) \tag{8}$$

$$u^P[(\bar{v},\bar{s}),\bar{t}] = c[(\bar{v},\bar{s}),\bar{t}] \cdot U_c^P(\bar{t}) + (1-c[(\bar{v},\bar{s}),\bar{t}]) \cdot U_p^P(\bar{t}) \tag{9}$$

$$u^I[(v,s),t] < u^I[(v^*,s^*),t^*] \qquad \text{where } ((v,s),t) \notin R \tag{10}$$

$$u^I[(\bar{v},\bar{s}),\bar{t}] = u^I[(v^*,s^*),t^*] \tag{11}$$

$$u^P[(\bar{v},\bar{s}),\bar{t}] = u^P[(v^*,s^*),t^*] \tag{12}$$

Figure 4: A non-linear program $\mathrm{NLP}[(C,sel),R]$**.**

then we can easily classify intruder's best responses (see Theorem 4).

For the rest of this section, we fix an end component (C,sel). Let $\sigma = (\hat{v},\hat{s},\kappa)$ be an $\mathcal{A}$-regular strategy such that $(C_\sigma, sel_\sigma) = (C,sel)$. Further, let $R \subseteq C \times T$. We say that an intruder's strategy π is *R-compatible* if the following conditions are satisfied:

- If $\pi(h) = enter_t$ where h ends in a vertex v, then $((v,s),t) \in R$ where $s = \delta(\hat{s},h)$.
- $\mathcal{P}^\sigma(noattack^\pi) = 0$.

Let $\mathrm{MBR}(\sigma) \subseteq \mathrm{BR}(\sigma)$ be the set of best responses to σ that maximize the patroller's expected utility. The next theorem gives a characterization of $\mathrm{MBR}(\sigma)$.

THEOREM 4. *Let σ be an $\mathcal{A}$-regular strategy such that $(C_\sigma, sel_\sigma) = (C,sel)$. Then there are two possibilities:*

- $\mathrm{MBR}(\sigma) = \{\pi\}$ *where $\pi(h) = wait$ for every history h;*
- $\mathrm{MBR}(\sigma) = \{\pi \in \Pi \mid \pi$ *is R-compatible*$\}$ *for some $R \subseteq C \times T$.*

PROOF SKETCH. For every $(v,s) \in C$ and $t \in T$, let $u^I[(v,s),t]$ and $u^P[(v,s),t]$ be the expected intruder's utility and the expected patroller's utility when the strategy σ is changed into (v,s',κ) where $\delta(s',v) = s$, and the intruder's strategy π satisfies $\pi(v) = enter_t$. Further, let $Val^I = \max\{u^I[(v,s),t] \mid (v,s) \in C, t \in T\}$. If $Val^I \leq 0$, then the intruder cannot gain anything by entering any target, and his (only) optimal response to σ is the strategy π such that $\pi(h) = wait$ for every history h. Now assume $Val^I > 0$. Let $\mathrm{BR} = \{((v,s),t) \mid u^I[(v,s),t] = Val^I\}$, and let $Val^P = \max\{u^P[(v,s),t] \mid ((v,s),t) \in \mathrm{BR}\}$. Finally, we put $R = \{((v,s),t) \in \mathrm{BR} \mid u^P[(v,s),t] = Val^P\}$. Clearly, for every R-compatible intruder's strategy π we have that $\mathrm{EU}_I^{\sigma,\pi} = Val^I$ and $\mathrm{EU}_P^{\sigma,\pi} = Val^P$, which proves the claim. □

The characterization of $\mathrm{MBR}(\sigma)$ given in Theorem 4 is implemented in the non-linear program of Fig. 4. This program covers the second possibility when $\mathrm{MBR}(\sigma)$ consists of all R-compatible intruder's strategies (the other possibility is discussed below). The variables $x[(v,s),v']$ encode a probability assignment for (C,sel), i.e., some $\mathcal{A}$-regular strategy σ (constraints (5) and (6)). We require that all of these probabilities are positive (constraint (5)), which implements the assumption $(C_\sigma, sel_\sigma) = (C,sel)$. The auxiliary variables $c[(v,s),t]$ encode the probability of capturing the intruder entering t when the patroller visits v and $\mathcal{A}$ is in the state s after reading the current history (constraint (7)). The variables $u^I[(v,s),t]$ encode the expected intruder's utility for σ and a $\{((v,s),t)\}$-compatible intruder's strategy (constraint (8)). Similarly, the $u^P[(\bar{v},\bar{s}),\bar{t}]$ variables, where $((\bar{v},\bar{s}),\bar{t}) \in R$, encode the expected patroller's utility for σ and a $\{((\bar{v},\bar{s}),\bar{t})\}$-compatible intruder's strategy (constraint (9)). Constraints (10) and (11) guarantee that $\mathrm{MBR}(\sigma)$ consists of R-compatible strategies. The $((v^*,s^*),t^*)$ is a fixed element of R (chosen arbitrarily).

Note that the possibility when $\mathrm{MBR}(\sigma) = \{\pi\}$, where $\pi(h) = wait$ for every history h, can be encoded by a simpler program consisting of the constraints (5)–(8) and $u^I[(v,s),t] \leq 0$. The objective of this program is irrelevant, we require just feasibility.

Now, for a given probability assignment λ for (C,sel), we can replace the right-hand sides of the non-linear constrains (7) with

$$\sum_{\substack{(v_0,s_0),\ldots,(v_n,s_n) \\ \in CP[(v,s),t]}} \left(x[(v_{n-1},s_{n-1}),v_n] \cdot \prod_{i=0}^{n-2} \lambda(v_i,s_i)(v_{i+1}) \right)$$

and thus obtain a linear program $\mathrm{LP}[(C,sel),R,\lambda]$. Given some initial strategy σ (and the associated probability assigment λ_σ for (C,sel)), we can again "improve" λ_σ by Algorithm 1 where $\mathrm{LP}[(C,sel),R,\lambda]$ is used instead of $\mathrm{LP}[(C,sel),\lambda]$. Observe that on top of choosing the "right" (C,sel) and the initial strategy σ, we also need to choose a suitable R. Although there are exponentially many candidates for the "right" R, a reasonable option is to try out all singletons first and then possibly proceed with larger subsets of C depending on the obtained results.

5. EXPERIMENTS

In this section we experimentally evaluate the strategy improvement algorithm designed in Section 4. For simplicity, we restrict our attention to zero-sum patrolling problems where $U_c^P(t) = U_c^I(t) = 0$, $U_p^P(t) = -1000$, and $U_p^I(t) = 1000$ for all $t \in T$. For every finite-state observer $\mathcal{A}$ and all $(\sigma,\pi) \in \Sigma_\mathcal{A} \times \Pi$, let $\mathcal{P}^\sigma(defend^\pi)$ be the probability of all walks where the intruder is captured or waits forever. That is,

$$\mathcal{P}^\sigma(defend^\pi) = \mathcal{P}^\sigma(noattack^\pi) + \sum_{t \in T} \mathcal{P}^\sigma(capture_t^\pi)\,.$$

We have that

$$1000 + \mathrm{EU}_P^{\sigma,\pi} = 1000 \cdot \mathcal{P}^\sigma(defend^\pi)\,.$$

Hence, the patroller/intruder actually aims at maximizing/minimizing $\mathcal{P}^\sigma(defend^\pi)$. Since this quantity appears more intuitive than $\mathrm{EU}_P^{\sigma,\pi}$, we also define the *defending value* of a given $\mathcal{A}$-regular strategy σ by $DVal(\sigma,\mathcal{G}) = 1000 + Val(\sigma,\mathcal{G})$, and we use $DVal(\sigma,\mathcal{G})$ instead of $Val(\sigma,\mathcal{G})$ when presenting the performance of the computed strategies. Hence, if $DVal(\sigma,\mathcal{G}) = 250$, then the patroller defends all targets with probability at least 0.25 against all intruder's strategies, and there exists an intruder's strategy π such that

Table 1: The Outcomes of the 1st Comparison.

$\|V\|$	$\|V_4\|$	$\|V_7\|$	$DVal(\sigma_{\mathcal{A}})$, $N \leq 20$	$DVal(\sigma_{\mathcal{B}})$, $N \leq 20$
9	3	6	500.00000	448.78190
11	6	5	345.47092	337.56432
13	5	8	328.80138	316.48640
15	8	7	257.93970	252.89687
17	7	10	246.77666	241.28626
19	10	9	204.66290	202.02953
21	9	12	200.00000	194.72047
23	12	11	168.63060	168.15142
25	11	14	163.13892	162.93894
26	11	15	160.06754	158.73937
27	14	13	145.73724	143.98550
31	16	15	127.24003	125.88467
31	13	18	136.05868	133.96578
35	18	17	112.46766	111.82252
36	15	21	117.34851	115.87416
39	20	19	101.47058	100.58408
41	17	24	103.35841	102.08437
43	22	21	092.00802	091.39703
46	19	27	092.11294	091.22598
47	24	23	084.01209	083.74692

the probability of defending all targets against π is exactly 0.25.

Note that Algorithm 1 is parameterized by the choice of (C, sel), the initial strategy σ, the number of iterations N, and the function *Combine* invoked at line 5. In all of our experiments, we use $Combine(\lambda, \alpha) = 0.5 \cdot \lambda + 0.5 \cdot \alpha$. The number of iterations is typically set to 20. The choice of (C, sel) and σ substantially influences the quality of the strategy produced by Algorithm 1. In principle, (C, sel) and σ can be constructed either "manually" in some ad-hoc way, or synthesized automatically by inspecting the topological properties of the underlying directed graph of $\mathcal{G}$. We compare two such methods. The first one is simple; we construct a trivial finite-state observer $\mathcal{B}$ with just one state s such that $\delta(s, v) = s$ for every vertex v. The associated end component is the unique maximal end component of $V \times \{s\}$ (recall that the directed graph of $\mathcal{G}$ is strongly connected), and the initial strategy is a strategy which selects uniformly among all vertices.

The second method works as follows. Let $\mathcal{G} = (V, E, T, d)$ be a patrolling problem where $T = V$ and $d(t) \geq 2$ for all $t \in T$. Further, let $D(\mathcal{G}) = \{d(v) \mid v \in V\}$, and for every $d \in D(\mathcal{G})$, let V_d be the set of all $v \in V$ such that $d(v) = d$. As a running example, consider the case when $D(\mathcal{G}) = \{3, 4, 7, 13\}$. For simplicity, let us first assume that $\mathcal{G}$ has a full topology, i.e., $E = V \times V$. Then, we can partition each V_d into d pairwise disjoint subsets

$$V_d[0], \ldots, V_d[d{-}2], V_d[d{-}1]$$

so that each $V_d[i]$, where $0 \leq i \leq d-2$, has $|V_d|$ div $(d{-}1)$ elements, and $V_d[d{-}1]$ has $|V_d|$ mod $(d{-}1)$ elements. For example, if $d = 4$ and V_4 consists of 11 vertices, we obtain the sets $V_4[0], V_4[1], V_4[2], V_4[3]$ such that each $V_4[i]$, where $0 \leq i \leq 2$, contains 3 vertices, and $V_4[3]$ contains the two remaining vertices.

Table 2: Some cases when $\mathcal{A}$ outperforms $\mathcal{B}$.

$D(\mathcal{G})$	$DVal(\sigma_{\mathcal{A}})$, $N \leq 20$	$DVal(\sigma_{\mathcal{B}})$, $N \leq 20$
$\{6, 9\}$	500.00000	433.27517
$\{8, 9\}$	500.00000	426.33215
$\{12, 14\}$	500.00000	413.44493
$\{14, 15\}$	500.00000	409.91645
$\{22, 25\}$	500.00000	403.87134

Let k be the least common multiple of all $d-1$ such that $d \in D(\mathcal{G})$. In our running example, k is the least common multiple of $2, 3, 6, 12$, which is 12. Now we construct a finite-state observer $\mathcal{A}$ with states $\{s_0, \ldots, s_{k-1}\}$ such that, for every $v \in V$, $\delta(s_i, v) = s_j$ where $j = (i+1) \bmod k$.

Further, we construct an end component (C, sel) in the following way: For every $i \in \{0, \ldots, k-1\}$, let

$$M_i = \bigcup_{d \in D(\mathcal{G})} V_d[i \bmod (d{-}1)] \cup V_d[d{-}1]$$

In our running example, we obtain, e.g.,

$$M_8 = V_3[0] \cup V_3[2] \cup V_4[2] \cup V_4[3] \cup V_7[2] \cup V_7[6] \cup V_{13}[8] \cup V_{13}[12]$$

The set C consists of all pairs (v, s_i) such that $0 \leq i < k$ and $v \in M_i$. Further, for each $(v, s_i) \in C$, we define $sel(v, s_i) = M_i \smallsetminus \{v\}$.

Intuitively, (C, sel) is constructed so that, for every $d \in D(\mathcal{G})$, the vertices of $V_d[0], \ldots, V_d[d{-}2]$ are visited precisely once in $d-1$ steps after every history h, where the only possible exception is the vertex where h ends. Since it is not clear how the remaining vertices of $V_d[d{-}1]$ should be visited, (C, sel) allows to visit all of them "any time". The initial strategy σ is defined by $\sigma = (\hat{v}, s_{k-1}, \kappa)$, where $\hat{v}$ is a fixed vertex of some $V_d[0]$, and $\kappa(v, s)$ is the uniform distribution over $sel(v, s)$.

Note that the above construction requires the existence of edges only among the vertices of M_i and $M_{(i+1) \bmod k}$. Hence, it is applicable also to patrolling problems where E is a proper subset of $V \times V$, but E must be "dense" in the sense that it contains the required edges.

For simplicity, in our experiments we consider only patrolling problems with fully connected topology where all vertices are targets, i.e., $E = V \times V$ and $T = V$.

In our first experiment, we examine the case when $D(\mathcal{G}) = \{4, 7\}$ for various (almost random) instances. The outcomes are summarized in Table 1. The columns $DVal(\sigma_{\mathcal{A}})$, $N \leq 20$, and $DVal(\sigma_{\mathcal{B}})$, $N \leq 20$ show the defending values of the best $\mathcal{A}$-regular and $\mathcal{B}$-regular strategies obtained by Algorithm 1 in at most 20 rounds. Note that $\sigma_{\mathcal{A}}$ always outperforms $\sigma_{\mathcal{B}}$, but the difference is not so significant as one might expect. This is because the numbers in $D(\mathcal{G})$ are still relatively small. In Table 2, we show the outcomes (obtained after 20 rounds) for some specific cases where $|V_d| = d-1$ for every $d \in D(\mathcal{G})$. For example, for the patrolling problem considered in the first line of Table 2 we have that $|V_6| = 5$ and $|V_9| = 8$. Here, $DVal(\sigma_{\mathcal{A}})$ is always equal to 500, but $DVal(\sigma_{\mathcal{B}})$ decreases as the numbers in $D(\mathcal{G})$ increase.

An interesting feature of Algorithm 1 revealed by our experiments is that the outcomes can "oscillate" among better and worse strategies as N increases. This behaviour is documented in Table 3, where we consider the case $D(\mathcal{G}) = \{3, 4, 7, 13\}$, and show the defending values for the strategies

Table 3: The Outcomes of the 2nd Comparison.

$\|V\|$	$\|V_3\|$	$\|V_4\|$	$\|V_7\|$	$\|V_{13}\|$	$DVal(\sigma_{\mathcal{A}})$, $N = 10$	$DVal(\sigma_{\mathcal{A}})$, $N = 15$	$DVal(\sigma_{\mathcal{A}})$, $N = 20$
25	4	3	6	12	200.00000	200.00000	200.00000
29	5	4	7	13	164.31373	164.36930	164.37348
33	6	5	8	14	136.23451	137.11697	137.73950
37	7	6	9	15	120.24780	115.83532	118.51474
41	8	7	10	16	105.64002	105.81254	105.50371

$\|V\|$	$\|V_3\|$	$\|V_4\|$	$\|V_7\|$	$\|V_{13}\|$	$DVal(\sigma_{\mathcal{B}})$, $N = 10$	$DVal(\sigma_{\mathcal{B}})$, $N = 15$	$DVal(\sigma_{\mathcal{B}})$, $N = 20$
25	4	3	6	12	197.03123	197.11447	197.11542
29	5	4	7	13	162.85415	162.90246	162.90214
33	6	5	8	14	138.67156	138.70364	138.70487
37	7	6	9	15	120.69474	120.71614	120.71646
41	8	7	10	16	106.81787	106.83392	106.83405

obtained after 10, 15, and 20 rounds. The mentioned oscillation is present in several lines (for both observers). It can be influenced by changing the mixing ratio used by *Combine*. The two finite-state observers $\mathcal{A}$ and $\mathcal{B}$ do comparably well on the presented instances. This is mainly because $|V_d|$ is not much larger than d, and hence the size of $V_d[d-1]$ is comparable to the size of $V_d[j]$ for $j \leq d-2$.

All of our results have been obtained by a naive implementation of Algorithm 1 in Python that was run on an average Intel 64 PC with 4GB RAM. We used the PuLP package to solve the linear programs (all of our instances were solved in less then two minutes, often is seconds). Despite this lightweight implementation, we were able to analyze patrolling problems with about 100 vertices. We believe that a more serious implementation can easily handle substantially larger instances.

6. CONCLUSIONS

We designed a strategy improvement algorithm for regular strategies which avoids the use of non-linear programming. This algorithm is applicable to patrolling problems of realistic size, and it can quickly produce quite efficient strategies.

There is a lot of space for improvement, and we believe that the main ideas presented in this paper can be further developed and refined into an even more efficient framework. The strategy improvement technique is generic and possibly applicable also to other problems that have so far been tackled using non-linear programming and other computationally costly methods. We see all of these possibilities as great challenges for future work.

REFERENCES

[1] M. Abaffy, T. Brázdil, V. Řehák, B. Bošanský, A. Kučera, and J. Krčál. Solving adversarial patrolling games with bounded error. In *Proceedings of AAMAS 2014*, pages 1617–1618, 2014.

[2] N. Agmon, S. Kraus, and G. A. Kaminka. Multi-robot perimeter patrol in adversarial settings. In *Proceedings of ICRA 2008*, pages 2339–2345, 2008.

[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 AAMAS 2008*, pages 55–62, 2008.

[4] N. Basilico, N. Gatti, and F. Amigoni. Patrolling security games: Definition and algorithms for solving large instances with single patroller and single intruder. *Artificial Intelligence*, 184–185:78–123, 2002.

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

[6] 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 *WI-IAT*, pages 557–564, 2009.

[7] N. Basilico, N. Gatti, and F. Villa. Asynchronous Multi-Robot Patrolling against Intrusion in Arbitrary Topologies. In *AAAI*, 2010.

[8] B. Bosansky, V. Lisy, M. Jakob, and M. Pechoucek. Computing Time-Dependent Policies for Patrolling Games with Mobile Targets. In *Proceedings of AAMAS 2011*, 2011.

[9] B. Bosansky, O. Vanek, and M. Pechoucek. Strategy Representation Analysis for Patrolling Games. In *AAAI Spring Symposium*, 2012.

[10] F. Fang, A. X. Jiang, and M. Tambe. Optimal Patrol Strategy for Protecting Moving Targets with Multiple Mobile Resources. In *Proceedings of AAMAS 2013*, 2013.

[11] M. Jain, E. Karde, C. Kiekintveld, F. Ordóñez, and M. Tambe. Optimal defender allocation for massive security games: A branch and price approach. In *Workshop on Optimization in Multi-Agent Systems at AAMAS*, 2010.

[12] 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 AAMAS 2009*, pages 689–696, 2009.

[13] E. Munoz de Cote, R. Stranders, N. Basilico, N. Gatti, and N. Jennings. Introducing alarms in adversarial patrolling games: extended abstract. In *Proceedings of AAMAS 2013*, pages 1275–1276, 2013.

[14] J. Pita, M. Jain, J. Marecki, F. Ordónez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus. Deployed ARMOR protection: The application of a game theoretic model for security at the Los Angeles Int. Airport. In *Proceedings of AAMAS 2008*, pages 125–132, 2008.

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

[16] J. Tsai, S. Rathi, C. Kiekintveld, F. Ordóñez, and M. Tambe. IRIS—a tool for strategic security allocation in transportation networks categories and subject descriptors. In *Proceedings of AAMAS 2009*, pages 37–44, 2009.

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