A Robust Optimization Approach to Designing Near-Optimal Strategies for Constant-Sum Monitoring Games

Aida Rahmattalabi, Phebe Vayanos, and Milind Tambe

University of Southern California
{rahmatta,vayanos,tambe}@usc.edu

Abstract. We consider the problem of monitoring a set of targets, using scarce monitoring resources (e.g., sensors) that are subject to adversarial attacks. In particular, we propose a constant-sum Stackelberg game in which a defender (leader) chooses among possible monitoring locations, each covering a subset of targets, while taking into account the monitor failures induced by a resource-constrained attacker (follower). In contrast to the previous Stackelberg security models in which the defender uses mixed strategies, here, the defender must commit to pure strategies. This problem is highly intractable as both players' strategy sets are exponentially large. Thus, we propose a solution methodology that automatically partitions the set of adversary's strategies and maps each subset to a *coverage policy*. These policies are such that they do not overestimate the defender's payoff. We show that the partitioning problem can be reformulated *exactly* as a mixed-integer linear program (MILP) of moderate size which can be solved with off-the-shelf solvers. We demonstrate the effectiveness of our proposed approach in various settings. In particular, we illustrate that even with few policies, we are able to closely approximate the optimal solution and outperform the heuristic solutions.

1 Introduction

Protection[1] of important targets is a critical security problem with a wide range of applications including environmental surveillance, and infrastructure security. One of the strategies is to monitor the targets by allocating resources, such as inspection posts, sensor devices, etc. However, this allocation task can become extremely challenging if one considers the possibility of malicious attacks [9]. Such adversarial actions will increase the vulnerability of targets; therefore, more strategic monitoring policies should be implemented in order to ensure a high level of robustness against potential adversarial attacks.

Game theory, and in particular, Stackelberg games have been used to model complex security problems, in which a defender first commits to a strategy and

[1] Throughout the paper, we will use the terms "cover", "monitor", "protect" interchangeably.

an attacker who can surveil the defender's strategy acts next to maximize the harm. Some of the important applications of these models can be found in [15, 16]. In Stackelberg security games, it is often assumed that the defender commits to a randomized strategy. While in many applications, the randomness is advantageous to the defender by making the action less predictable, there are also many security domains for which a randomized solution is not feasible, e.g., static sensor placement for monitoring. Also, most of the previous work has focused on models in which the targets are subject to attacks, whereas it is also possible for an adversary to attack the defender's resources.

In this paper, we introduce the "strategic monitoring problem", in which a defender aims to maximize the total value of the targets it protects by placing a limited number of monitors. An adversary who aims to make the targets vulnerable, attacks some of the monitors such that the value of the unprotected targets is maximized. We view this problem as a two-player Stackelberg game.

In our model, we assume that all of the targets are at risk and important to be protected; therefore, the defender obtains a positive payoff equal to the total value of protected targets, whereas the attacker's reward is evaluated based on the accumulative value of the targets that are unprotected. As a result, the sum of both players' payoff is equal to the total value of the targets. Our goal is to find a minimax pure strategy for the defender, that is, a strategy that maximizes the minimum payoff that the defender can obtain.

In terms of modeling, our model extends the existing literature of Stackelberg security games by considering a more general attack model, which allows the adversary to attack the resources. In addition, we solve for pure strategies for the defender. Our commitment to pure strategies is due to the assumption that the monitors are fixed and as a result randomized solutions are not applicable. Furthermore, our model is general as it can accommodate for heterogeneous targets (with arbitrary values) and monitors (with different monitoring powers). We will elaborate on this in the formal problem description.

In terms of technical contributions, the strategic monitoring problem that we study is highly intractable as both players' strategy set is exponentially large. In order to tackle this problem, we propose a novel max-min-max binary optimization model, which allows us to leverage techniques from robust optimization literature. In particular, we extend the K-adaptability idea from two-stage robust optimization literature, based on which first the desired set of monitors together with K candidate coverage policies are selected. This is equivalent to partitioning adversary actions into K subsets, such that each subset is mapped to a particular coverage policy. The coverage policies are such that the value of the covered targets is not overestimated, but as high as possible. We extend the work of Hanasusanto et al. [11] by generalizing their approach to the case of discrete adversary actions by exploiting the specific structure of our problem. We show that, in contrast to their formulation, we can reformulate the K-adaptability problem as an MILP that is exact. The significance of the MILP formulation is that it is polynomial in all problem inputs; thus, it circumvents the exponentiality of the attacker's action. Furthermore, our approach bridges the gap between

the suboptimal heuristic solutions and the fully optimal, yet intractable exact approach, where the trade-off between complexity and optimality can be tuned using a single design parameter K.

In the remainder of this paper, we first give an overview of the related work. Next, in Section 3 we formally define the strategic monitoring problem as a constant-sum Stackelberg game and we show that it can be equivalently modeled as a two-stage robust optimization problem. Following that, we introduce the K-adaptability counterpart problem and we prove it can be reformulated exactly as a single optimization problem of moderate size. Finally, in Section 4 we present results that demonstrate how the presented approach performs across different criteria. The paper concludes with a summary of contributions.

2 Related Work

The strategic monitoring problem falls under the category of *large scale constant-sum games* with exponential strategy space. Mainly, there are two approaches to tackle large scale games: One approach is based on iterative strategy generations used by double-oracle algorithms [12, 8] for which there is no guaranteed polynomial run-time. The other approach focuses on using compact representations of the games, where a common approach is based on clustering strategies to solve simpler games. In this regard, Bard et al. [1] propose a greedy-based clustering approach. Also, in [2] authors use k-means clustering to construct the abstract games. What we propose in this work can be viewed as an automatic generation of a partition of adversary's strategy set, which is not reliant on any metric such as the ones used in the clustering algorithms. In fact, the partitioning is performed implicitly by choosing limited number of coverage policies.

This problem is also related to *robust sub-modular optimization*. In this regard, Krause et al. [13] formalized a general max-min problem, and they proposed an approximation algorithm to maximize the worst-case performance of a sub-modular function against a set of possible failure scenarios but their algorithm is only efficient for moderately-sized set of scenarios. Later, Orlin et al. [14] studied a problem, in which one chooses a set of up to I items, and nature counteracts by eliminating at most J of the selected items. The objective to maximize a monotone sub-modular set function. The authors propose a greedy-based algorithm with a constant (0.387) factor approximation result, valid for $J = o(\sqrt{I})$. This work was followed by [7], in which they show the same approximation factor for $J = o(I)$. In [17], the authors propose another greedy-based algorithm and provide a bound using the curvature of the sub-modular function. Although these greedy algorithms are computationally efficient, the approximation guarantees are quite loose, whereas in some applications, such as monitoring, it is more desirable to spend more time in the decision making phase, since once the monitoring locations are chosen, they will be in use for a long duration.

Finally, our solution approach draws from *robust optimization* (RO) literature. RO models concern decision making problems affected by uncertainty, in which the uncertainty is modeled as a set, also referred to as uncertainty set.

This class of problems are modeled as max-min optimization problems and can also be considered as a zero-sum game against "nature" which acts as an adversary by choosing the worst setting of the uncertain parameters. For further reading one can refer to [3, 5]. *Two-stage robust optimization* is an extension of the single-stage RO problems in which the decision maker chooses a secondary action upon observing nature's choice.

This class of problems are intractable in general [4], specially if the second-stage actions are binary. However, there exists efficient approximation schemes which have been proven to perform well in practice. In particular, finite adaptability has been proposed [6], in which the nature's action set, is partitioned and a second-stage decision is determined for each partition. These partitions can be either fixed by the modeler [18] or decided in the optimization [6, 11] process. In the present work, we propose a novel two-stage optimization model for the strategic monitoring problem and we build on the work of [11] which proposes a methodology for obtaining K partitions, also known as K-adaptability. In [11], the authors show that for polyhedron (convex) uncertainty sets, a two-stage robust optimization can be approximately reformulated as an MILP. We generalize their result to the case of discrete sets, and we provide an MILP reformulation that is exact.

3 Strategic Monitoring Problem

We are given a set of monitoring locations $\mathcal{N} := \{1, \ldots, N\}$, and a set of targets $\mathcal{T} := \{1, \ldots, T\}$. Each target n has a (normalized) value $U_n \in [0, 1]$ which indicates the importance of that target. Further, each monitor $n' \in \mathcal{N}$ can cover a subset of the targets. We represent the target coverage via a bipartite graph $G = (\mathcal{N}, \mathcal{T}, \mathcal{E})$, where $\mathcal{E}$ is the set of edges between $\mathcal{N}$ and $\mathcal{T}$. An edge from $n' \in \mathcal{N}$ to $n \in \mathcal{T}$, denoted by (n', n), exists if n can be monitored by n' (e.g., n is within the observable range of n'). For each target n, we define $\delta(n) := \{n' \in \mathcal{N} : (n', n) \in \mathcal{E}\}$ which is the set of nodes that can monitor n. Figure 1 depicts an example graph, in which the circles are the monitoring locations, and the squares are the targets that need to be protected. We consider a constant-sum Stackelberg game as:

$$\max_{\substack{\mathcal{X} \subseteq \mathcal{N} \\ |\mathcal{X}| \leq I}} \min_{\substack{\mathcal{Z} \subseteq \mathcal{X} \\ |\mathcal{Z}| \leq J}} F(\mathcal{X} \backslash \mathcal{Z}), \tag{1}$$

in which a defender aims to select a set of nodes $\mathcal{X} \subseteq \mathcal{N}$ of cardinality at most I as monitors such that it maximizes the coverage of the targets after an adversary eliminates subset $\mathcal{Z} \subseteq \mathcal{X}$ of the chosen monitors. The payoff function $F(\cdot)$ evaluates the targets that are covered, given the defender and attacker's strategies, and it is defined precisely as follows:

$$F(\mathcal{Y}) := \sum_{n \in \mathcal{T}} U_n \mathcal{I}\left(\exists n' \in \mathcal{Y} \ : \ n' \in \delta(n)\right), \tag{2}$$

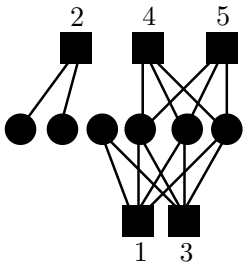

Fig. 1. An example input of the strategic monitoring problem. In this figure the circles represent targets, and squares are the monitoring locations.

where $\mathcal{I}(\cdot)$ is the indicator function defined as:

$$\mathcal{I}(P) = \begin{cases} 0 & \text{if } P \equiv \textbf{FALSE} \\ 1 & \text{if } P \equiv \textbf{TRUE} \end{cases}. \tag{3}$$

According to this definition, a target is covered iff at least one its neighbors is chosen as a monitor and has not been attacked by the adversary.

In the following proposition, we show the importance of modeling the adversary. We prove that the optimal solution of the problem that ignores the existence of an adversary can be quite sub-optimal in the presence of adversary.

Observation 1 *The optimal solution of a problem that ignores the possibility of adversarial attacks can be sub-optimal in Problem* (1) *with optimality gap in the order* $\mathcal{O}(T)$*.*

Proof. We prove this by means of an example. Consider an instance of Problem (1) on the network depicted in Figure 1, with input given as $N = 5$, $T = 6$, $I = 2$, $J = 1$. We also assume that all of the targets have a value equal to 1. In the absence of an adversary, (or if we ignore the adversary), an optimal solution is to choose nodes 1 and 2 which will cover all of the targets. If an adversary exists, however, this decision can be highly sub-optimal as in this case if node 1 is attacked, only 2 targets will be covered. By optimizing against an adversary, the optimal decision is to select nodes 1, and 3. This solution obtains a coverage of 4.

In this particular example, we observed an optimality gap of 2 (=4-2). Now, consider the same network structure with T targets, in which nodes 1, and 3 are connected to $T - 2$ targets and node 2 covers the remaining 2 targets. The optimality gap in this case is $T - 4$ which increases linearly with the number of targets. Therefore, we can conclude that in the worst-case this gap is $\mathcal{O}(T)$. □

In the description of Problem (1), the adversary's choice is *dependent* on the decision maker's choice $\mathcal{X}$. We propose an alternative formulation in which the dependence on $\mathcal{X}$ is removed, and the adversary can choose from the ground set $\mathcal{N}$ (instead of $\mathcal{X}$). We show that the two problems are equivalent.

Proposition 1. *Stackelberg game model* (1) *is equivalent to:*

$$\max_{\substack{\mathcal{X}\subseteq\mathcal{N}\\ |\mathcal{X}|\leq I}} \min_{\substack{\mathcal{Z}\subseteq\mathcal{N}\\ |\mathcal{Z}|\leq J}} F(\mathcal{X}\backslash\mathcal{Z}), \tag{4}$$

in which the adversary can choose among the set $\mathcal{N}$.

Proof. This proof is based on the intuition that a rational adversary will always choose among the selected monitors by the defender, even if it is given the option to attack other nodes. The formal proof is given below.

Fix an arbitrary $\mathcal{X} \subseteq \mathcal{N}$ and let z and w denote the optimal objective values of the inner minimization problems in (1) and (4), respectively. We will show that $w = z$, which given the choice of $\mathcal{X}$ is arbitrary, results in the equivalence of the two problems. Since $X \subseteq \mathcal{N}$, it follows that $w \leq z$. We show that the converse is also true. Let $\mathcal{Z}^\star$ be optimal decision for the inner minimization problem in (4). We show that one can construct a solution $\overline{\mathcal{Z}} \subseteq \mathcal{X}$ feasible in the inner minimization problem of (1) such that $F(\mathcal{X}\backslash\overline{\mathcal{Z}}) = F(\mathcal{X}\backslash\mathcal{Z}^\star)$, implying that $z \leq w$. If $\mathcal{Z}^\star \subseteq \mathcal{X}$, we can define $\overline{\mathcal{Z}} = \mathcal{Z}^\star$ and the claim follows. Else, let $z \in \mathcal{Z}^\star\backslash\mathcal{X}$ and define $\overline{\mathcal{Z}} := \mathcal{Z}^\star\backslash\{z\}$. Then $\mathcal{X}\backslash\overline{\mathcal{Z}} = \mathcal{X}\backslash\{\mathcal{Z}^\star\backslash\{z\}\} = \mathcal{X}\backslash\mathcal{Z}^\star$ and thus $F(\mathcal{X}\backslash\overline{\mathcal{Z}}) = F(\mathcal{X}\backslash\mathcal{Z}^\star)$. As the choice of $\mathcal{X}$ was arbitrary, the proof is complete. □

3.1 Reformulation as a Two-Stage Robust Binary Program

In this section, we show that the strategic monitoring problem can be reformulated as a two-stage binary program. Since the two Problems (1) and (4) are equivalent, we will focus on the latter. Indeed, as it will become apparent later on, this simplification will enable us to reformulate Problem (1) exactly as an MILP. The two-stage binary program is as follows:

$$\max_{\boldsymbol{x}\in\mathcal{U}} \min_{\boldsymbol{\xi}\in\Xi} \max_{\boldsymbol{y}\in\{0,1\}^T} \left\{ \sum_{n\in\mathcal{T}} U_n y_n \ : \ \sum_{n'\in\delta(n)} \xi_{n'} x_{n'} \geq y_n,\ \forall n \in \mathcal{T} \right\}. \tag{5}$$

In this formulation, $\boldsymbol{x}$ is a binary vector and $x_n = 1$ iff node n is chosen to place a monitor. Binary vector $\boldsymbol{\xi}$ encodes whether a node is not attacked, where $\xi_n = 0$ iff node n is attacked by the adversary. Also, binary vector $\boldsymbol{y}$ indicates which targets are monitored. Note that the value of $\boldsymbol{y}$ can be determined after the adversary's action is revealed, which forces the introduction of the second-stage counting stage. Set $\mathcal{U} = \{\boldsymbol{x} \ : \ \sum_{n\in\mathcal{N}} x_n \leq I\}$ is the set of all feasible monitor selections. Also, Ξ is the set of feasible actions of the adversary and it is defined as:

$$\Xi := \left\{ \boldsymbol{\xi} \in \{0,1\}^{|\mathcal{N}|} \ : \ \sum_{n\in\mathcal{N}} (1 - \xi_n) \leq J \right\}. \tag{6}$$

This set expresses that at most J nodes can be attacked by the adversary, which is equivalent to the definition used in Problem (4). The first maximization problem determines the value of $\boldsymbol{x}$, i.e., the set of monitoring locations. In the inner

minimization problem, the adversary chooses which monitors to attack. Finally, the innermost maximization problem determines the covered targets i.e., the problem $\max_{\boldsymbol{y}\in\{0,1\}^T}\left\{\sum_{n\in\mathcal{T}} U_n y_n : \sum_{n'\in\delta(n)} \xi_{n'} x_{n'} \geq y_n, n\in\mathcal{T}\right\}$ models the payoff function $F(\cdot)$ introduced in Problem (1). The constraints of this problem stipulate that a target node is monitored if there is at least a monitor among its neighbors, which is not attacked.

Remark 1. In Problem (5), set $\mathcal{U}$ can be defined by any arbitrary linear constraints, and our solution approach remains valid. However, we are only considering cardinality constraints in the definition of $\mathcal{U}$.

Proposition 2. *Problem* (4) *is equivalent to the two-stage robust monitoring Problem* (5).

Proof. Problem (4) is equivalent to:

$$\max_{\substack{\mathcal{X}\subseteq\mathcal{N}\\ |\mathcal{X}|\leq I}} \min_{\substack{\mathcal{Z}\subseteq\mathcal{N}\\ |\mathcal{Z}|\leq J}} F(\mathcal{X}\backslash\mathcal{Z}) = \max_{\boldsymbol{x}\in\mathcal{U}} \min_{\boldsymbol{\xi}\in\Xi} F(\{n\in\mathcal{N} : x_n = 1\}\backslash\{\boldsymbol{\xi}\in\Xi : \xi_n = 0\}),$$

thus, it suffices to show that for any $\boldsymbol{x}$, and $\boldsymbol{\xi}$:

$$\begin{aligned} &F(\{n\in\mathcal{N} : x_n = 1\}\backslash\{\boldsymbol{\xi}\in\Xi : \xi_n = 0\}) = \\ &\max_{\boldsymbol{y}\in\{0,1\}^T}\left\{\sum_{n\in\mathcal{T}} U_n y_n \ : \ \sum_{n'\in\delta(n)} \xi_{n'} x_{n'} \geq y_n,\ \forall n\in\mathcal{T}\right\}. \end{aligned} \tag{7}$$

Let $\boldsymbol{y}^\star$ be the optimal solution of the maximization problem:

$$\forall n\in\mathcal{T} : y_n^\star = 1 \Rightarrow \sum_{n'\in\delta(n)} \xi_{n'} x_{n'} \geq 1.$$

Also, we note that the opposite direction holds true, meaning that:

$$\sum_{n'\in\delta(n)} \xi_{n'} x_{n'} \geq 1 \Rightarrow y_n^\star = 1,$$

otherwise, we can construct a new solution $\tilde{\boldsymbol{y}}$ with higher objective which contradicts the optimality of $\boldsymbol{y}^\star$. As a result,

$$y_n^\star = 1 \Leftrightarrow \exists n'\in\delta(n) : \xi_{n'} = 1,\ x_{n'} = 1,$$

or equivalently:

$$y_n^\star = \mathcal{I}(\exists n'\in\delta(n) : \xi_{n'} = 1,\ x_{n'} = 1).$$

By summing over all $n\in\mathcal{T}$:

$$\begin{aligned} \sum_{n\in\mathcal{T}} U_n y_n^\star &= \sum_{n\in\mathcal{T}} U_n \mathcal{I}(\exists n'\in\delta(n) : \xi_{n'} = 1,\ x_{n'} = 1) \\ &= F(\{n\in\mathcal{N} : x_n = 1\}\backslash\{\boldsymbol{\xi}\in\Xi : \xi_n = 0\}), \end{aligned} \tag{8}$$

where the last equality follows by the definition of the coverage function $F(\cdot)$.

□

3.2 K-Adaptability

K-Adaptability has been proposed to approximate the solution to the two-stage robust optimization problems with integer recourse decisions. In K-adaptability, K non-adjustable second-stage policies $\boldsymbol{y}^k, k \in \{1, \cdots, K\}$ are chosen in the first stage, that is before the adversary takes an action. Upon observing the adversary's action, the best policy among the feasible ones will be output as the solution. This is equivalent to automatically partitioning the adversary actions into K subsets, such that each subset is mapped to particular covering policy. The covering policies are such that the number of covered nodes is not overestimated, but is as high as possible. In the strategic monitoring game, the second-stage variables are in fact indicator functions that indicate, for each node, whether it is covered or not. In K-adaptability, we approximate this indicator function, where we limit ourselves to a small number (K) of counting policies. The payoff will be then evaluated based on the indicator function, thus, K-adaptability serves as an approximation scheme of the payoff function.

K-Adaptability in Strategic Monitoring Problem

The K-adaptability counterpart of Problem (5) can be expressed as:

$$\max_{\substack{\boldsymbol{x}\in\mathcal{U}\\ \boldsymbol{y}^k\in\{0,1\}^T}} \min_{\boldsymbol{\xi}\in\Xi} \max_{k\in\mathcal{K}} \Big\{ \sum_{n\in\mathcal{N}} U_n y_n^k \;:\; y_n^k \leq \sum_{n'\in\delta(n)} \xi_{n'} x_{n'}, \forall n \in \mathcal{T} \Big\}. \tag{9}$$

In Formulation (9), $\boldsymbol{x}$ encodes which nodes are chosen as monitors. Variables $\boldsymbol{y}^k$ are the K covering policies, where each policy $\boldsymbol{y}^k$ indicates which target nodes are covered. In other words, $y_n^k = 1$ means that according to the k^{th} policy, node n is monitored. These policies are chosen in the first stage, before observing the adversary's action. In addition, $\boldsymbol{\xi}$ denotes the adversary's action which lies in the set of adversary's pure strategies Ξ. This set is defined in Equation (6). Also, set $\mathcal{K} := \{1, \cdots, K\}$.

In the first maximization problem, the defender chooses both the monitoring nodes, and K covering policies. If $y_n^k = 1$, it means that according to policy k, node n is monitored. In the minimization problem, the adversary counteracts by choosing which nodes to attack. After observing which monitoring nodes are not attacked, the best feasible policy is chosen in the inner-most maximization problem. Policy k is feasible if it satisfies the constraints in the innermost maximization problem. The chosen policy is an approximation to the true payoff that the defender receives.

These policies approximate the true coverage, meaning that instead of enumerating all defender-attacker pairs of actions and evaluating the corresponding payoffs, one approximates the payoff, using K covering policies. This function is determined simultaneously with the defender's optimal strategy, in the formulation presented. We will illustrate the K-adaptability via an example.

Example 1. Consider an instance of the problem on a graph depicted in Figure 2, where all of the targets have equal values. We consider a setting with $I = 3$, and

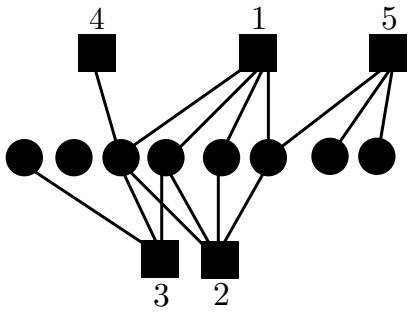

Fig. 2. Companion figure to Example 1. An example to illustrate the K-adaptability

$J = 1$. *For* $K = 1$, *an optimal solution to the* 1*-adaptability problem is* $\boldsymbol{x} = [1, 1, 0, 0, 1]$, *with the policy* $\boldsymbol{y}^1 = [0, 0, 1, 1, 1, 1, 0, 0]$. *According to this solution, the defender chooses nodes 1, 2, and 5 as monitors. In this case, the adversary's best response is to attack node 5 which results in the coverage of only 4 target nodes (those covered by 1, and 2) and this is captured by policy* $\boldsymbol{y}^1$. *Note that the policy* y^1 *is feasible under any other attacker's response. This means that even if the adversary chooses nodes 1 or 2 to attack, the same 4 targets would be covered. In fact, under these scenarios more targets are covered, but the policy under-counts those covered targets by setting their coverage value to 0 in order to ensure feasibility for the case that node 5 is attacked. As a result, we obtain a conservative approximation of the problem.*

Now, let us compare this solution to the solution to the 2*-adaptability problem. With* $K = 2$, *the payoff function is described approximately via two policies. In this case, the optimal defender strategy is* $\boldsymbol{x} = [1, 0, 1, 0, 1]$ *and the two policies are equal to:* $\boldsymbol{y}^1 = [1, 0, 1, 1, 1, 1, 0, 0]$, *and* $\boldsymbol{y}^2 = [0, 0, 1, 1, 0, 1, 1, 1]$. *If the attacker chooses to attack either node 1 or 3, policy* y^2 *will be feasible , which indicates that 5 nodes will be always covered (in either of the scenarios). If the attacker chooses node 5, policy* y^1 *is feasible which covers another set of 5 nodes. Comparing to the* $K = 1$ *case, the coverage is increased by 1.*

This example also gives insights on how our approach allows an adjustable approximation to the true optimal solution with a single parameter K. *In fact, in this example, the solution of* $K = 1$ *is the same as the greedy algorithms proposed in [14, 7, 10]. By increasing* K, *the optimal objective value of the* K*-adaptability problem approaches the optimal solution of the original problem. Also, in this example, the solution of the 2-adaptability problem is optimal as it yields the optimal coverage of 5.*

Proposition 3. *Value of* K *in order to recover an optimal solution to Problem* (5) *is upper-bounded by* $\binom{I}{J}$. *Moreover, there are instances of Problem* (5) *for which this bound is tight, in the sense that exactly* $K = \binom{I}{J}$ *policies are needed in order to obtain the optimal coverage.*

Proof. We note that Problem (5) is always solvable since given any fixed value of K, $\boldsymbol{x} = \mathbf{0}$ and $\boldsymbol{y}^k = \mathbf{0}$, $\forall k \in \{1, \cdots, K\}$ is always a feasible solution. Also, observe that the cardinality of the set of feasible second stage actions is $|\{0,1\}^T| = 2^T$. Thus, there exists an optimal solution with 2^T policies. Given an optimal solution $(\boldsymbol{x}^\star, \boldsymbol{y}^{1^\star}, \ldots, \boldsymbol{y}^{K^\star})$, we show that we can construct an optimal solution with the same objective value and with only $K' = \binom{I}{J}$ policies.

Since $(\boldsymbol{x}^\star, \boldsymbol{y}^{1^\star}, \ldots, \boldsymbol{y}^{K^\star})$ is optimal, there exists a partition of Ξ into K disjoint subsets $\{E^{(k)}\}_{k=1}^K$ such that $\boldsymbol{y}^k$ is feasible and optimal, for all $\boldsymbol{\xi} \in \Xi^{(k)}$. Specifically, we can define:

$$\Xi^{(k)} = \left\{ \boldsymbol{\xi} \in \Xi : k = \min \left\{ k' : k' \in \underset{k \in \{1,\cdots,K\}}{\arg\max} \{ \sum_{n \in \mathcal{T}} U_n {y^k}_n^\star : \sum_{n' \in \delta(n)} \xi_{n'}^k x_{n'}^\star \geq {y^{k^\star}}_n, \forall n \right\} \right\}.$$

Also, it follows directly from the definition of $\{\Xi^{(k)}\}_{k=1}^K$ that:

$$\boldsymbol{\xi} \in \Xi^{(k)} \Rightarrow \boldsymbol{\xi}' \in \Xi^{(k)} : \boldsymbol{\xi}' \circ \boldsymbol{x} = \boldsymbol{\xi} \circ \boldsymbol{x},$$

where $(\circ)$ indicates the Hadamard product. This implies that for any K such that $\Xi^{(k)}$ is non-empty, $\boldsymbol{\xi}' \circ \boldsymbol{x}$ has a unique value for the $\boldsymbol{\xi} \in \Xi^{(k)}$. Finally, we note that, for any $\boldsymbol{x}$, there are only $\binom{I}{J}$ uniques values for $\boldsymbol{\xi} \in \Xi^{(k)}$, it follows that the maximal number of subsets that are non-empty is at most $\binom{I}{J}$. Since at most $\binom{I}{J}$ subsets are non-empty, we can eliminate all policies associated with empty subsets, and maintain an optimal solution.

Now we prove, via an example, that there exist instances where exactly $K = \binom{I}{J}$ is needed in order to obtain the optimal solution. Consider the example network in Figure 3. Let us assume the values $I = N$, and $J = 1$. For simplicity, we assume all targets have equal values. Here, the defender will choose all the monitoring nodes and the optimal coverage value is $N - 1$.

In the K-adaptability problem, an example policy (feasible for the case that node 1 is attacked) will be equal to $[0, 1, \cdots, 1]$, which gives the $N - 1$ coverage. However, this policy is not feasible under other attack scenarios. In general if node n is attacked, a feasible policy would be a vector whose entries are equal to 1, except for the n^{th} entry, which is equal to 0. Since the total number of scenarios is N, we can only obtain the optimal coverage of $N - 1$ with $K = \binom{N}{1}$ policies.

□

Remark 2. This result is stronger than what authors in [11] propose. Their upper bound on the number of policies needed in order to obtain an optimal solution to the Problem (5) is $K = 2^T$ (remember T is the number of targets). Here, we showed that this bound can be improved to $K = \binom{I}{J}$.

The following proposition provides the lower bound on the value of K in order to ensure that the K-adaptability problem yields a non-zero solution.

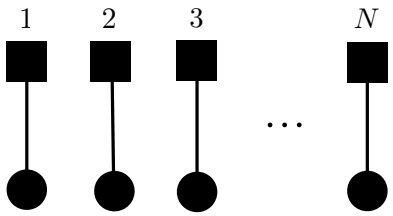

Fig. 3. Companion figure of Propositions 3 and 4.

Proposition 4. *Let $\Delta^{(t)}$ be the t^{th} highest degree in the network (e.g. $\Delta^{(1)}$ is the maximum degree and $\Delta^{(N)}$ is the minimum degree). For any $K \geq \min\{\ T\ :\ \min(I, \sum_{t=1}^{T} \Delta^{(t)}) \geq J+1\ \}$, at least one node will be covered. Moreover, this bound is tight in the sense that there are instances for which exactly $K = \min\{\ T\ :\ \min(I, \sum_{t=1}^{T} \Delta^{(t)}) \geq J+1\}$ is needed to have a non-zero coverage.*

Proof. We first show a way to construct a solution $\boldsymbol{x}$ to the Problem (9), which requires the minimum number of policies (value of K) and obtains a non-zero coverage.

The intuition is that if we require that at least one node is covered, that node had better have the maximum number of neighbors, because this will increase the likelihood that the node will be covered. As a result, we rank and rename all the nodes in descending order of their degree. Let us use $\Delta^{(t)}$ to denote the t^{th} highest degree node in the network, meaning that $\Delta^{(1)} \geq \Delta^{(2)} \geq \cdots \geq \Delta^{(N)}$.

We start from the first node in this order, and we select all of its neighbors. We continue until either we exceed the budget I, or we have chosen all the neighbors. Next, we check whether $\min(I, \sum_{t=1}^{T} \Delta^{(t)}) \leq J$, in which T is the current node's index, i.e., T^{th} highest-degree node. This condition determines if the number of chosen nodes is less than the number of nodes that can be attacked. If this condition holds, we move to the next highest-degree node and repeat the steps.

At termination, there are $T^\star$ nodes which have neighbor nodes that are chosen. The condition $J < \min(I, \sum_{t=1}^{T^\star} \Delta^{(t)})$ also suggest that *at least one* of these $T^\star$ nodes will be covered since the total number of chosen nodes exceeds the number of nodes that are unavailable. We do not know a priori which of these $T^\star$ node will be covered. Therefore, we define $T^\star$ policies, where policy $\boldsymbol{y}^k := \boldsymbol{e}^k$ ($\boldsymbol{e}^k$ is a all-zeros vector with 1 in k^{th} entry), $\forall k \in \{1, \cdots, T^\star\}$. These policies are feasible in Problem (9), as for each possible adversary's action, one of the above policies will be a feasible coverage. Also, the worst-case coverage is 1.

So far, we have constructed a solution which ensures a worst-case coverage of 1. In other words, $T^\star$ is an upper bound on K. Next, we prove that there are network structures for which this bound is tight, meaning that exactly $K = T^\star$ is needed in order to obtain a non-zero objective. Consider a network structure

such as the one depicted in Figure 3. For $I = 2$ and $J = 1$, a solution is to choose nodes 1, and 2 and $T^\star = 2$. We can observe that with fewer policies, i.e., $K = 1$, the covering policy is all zeros, as this is the only feasible policy for all adversarial actions. This means that *exactly* 2 policies are needed in order to obtain a non-zero coverage which is equal to 1, and indeed it is optimal.

Finally, note that, for the sake of this proof, the values of the targets are not important as we are only interested in a non-zero solution and this will be achieved by making sure that there are enough covering policies. □

3.3 Reformulation as an MILP

In this section, we derive an exact formulation for the K-adaptability counterpart of the strategic monitoring problem. Our approach is inspired by work of Hanasusanto et al. [11] who show that the K-adaptability problem of a two-stage robust optimization problem with binary second-stage actions can be approximately reformulated, as an MILP, for Ξ defined as a non-empty polyhedron. In this section, we show a stronger result by proving that we can provide an MILP formulation that is exact, and it extends to the discrete set Ξ.

The constraints in the inner maximization problem make Problem (9) less well-behaved. An alternative formulation is:

$$\max_{\substack{\boldsymbol{x}\in\mathcal{U}\\ \boldsymbol{y}^k\in\{0,1\}^T}} \ \min_{\boldsymbol{l}\in\mathcal{L}} \ \min_{\boldsymbol{\xi}\in\Xi(\boldsymbol{x},\boldsymbol{y}^{\mathcal{K}},\boldsymbol{l})} \ \max_{k\in\mathcal{K},\, l_k=0} \sum_{n\in\mathcal{T}} U_n y_n^k, \tag{10}$$

in which, $\mathcal{L} = \{0, \ldots, T\}^K$. Also, set $\Xi(\boldsymbol{x}, \boldsymbol{y}^{\mathcal{K}}, \boldsymbol{l})$ is a subset of the set Ξ, dependent on $\boldsymbol{x}$, $\boldsymbol{y}^{\mathcal{K}} := \{\boldsymbol{y}^1, \cdots, \boldsymbol{y}^K\}$, and $\boldsymbol{l}$ and is defined as:

$$\Xi(\boldsymbol{x}, \boldsymbol{y}^{\mathcal{K}}, \boldsymbol{l}) = \left\{ \boldsymbol{\xi} \in \Xi : \begin{array}{ll} y^k_{l_k} > \displaystyle\sum_{n'\in\delta(l_k)} \xi_{n'} x_{n'}, & \text{if } \ \forall k \in \mathcal{K} : l_k > 0 \\ y^k_n \le \displaystyle\sum_{n'\in\delta(n)} \xi_{n'} x_{n'}, \ \forall n \in \mathcal{T}, & \text{if } \ \forall k \in \mathcal{K} : l_k = 0 \end{array} \right\}, \tag{11}$$

In the above definition, vector $\boldsymbol{l}$ encodes which of the K second-stage policies are feasible. If $l_k = 0$, it means that policy k is feasible; therefore, all the constraints must be satisfied, i.e., all the coverage constraints for all of the targets. Note that the inner-most maximization problem chooses the best feasible policy ($l_k = 0$). On the other hand, if $l_k > 0$, it indicates that there is at least one constraint that is violated by policy k, and the value of l_k indicates which constraint. In this definition, and according to the first constraint for $l_k > 0$, policy k violates the constraint corresponding to node l_k, whereas if $l_k = 0$, it means that policy k must satisfy all the constraints, thus the constraints are imposed for all $n \in \mathcal{T}$. As a result, by introducing $\boldsymbol{l}$, the constraints of the inner maximization problem are absorbed by decision $\boldsymbol{l}$, and parameterized sets $\Xi(\boldsymbol{x}, \boldsymbol{y}^{\mathcal{K}}, \boldsymbol{l})$.

Remark 3. In the above definition of vector $\boldsymbol{l}$, it is sufficient to find at least one constraint violation in order for policy k to be infeasible, and l_k records the index

of that constraint (equivalently, the node for which the coverage constraint is violated).

While the discrete nature of set Ξ prohibits any attempts to use duality theory in the reformulation as an MILP, in the following proposition, we show that using certain structure of sets defined by Equation (11), we can disregard the intergality constraints and obtain a convex set.

Proposition 5. *Problem* (10) *remains unchanged if we replace the set* Ξ *with the following:*

$$\Xi := \left\{ \boldsymbol{\xi} \in [0,1]^{|\mathcal{N}|} \ : \ \sum_{n \in \mathcal{N}} (1 - \xi_n) \leq J \right\}. \tag{12}$$

in which the integrality constraint on $\boldsymbol{\xi}$ *is relaxed.*

Proof. Throughout this proof we use Ξ^{convex} and Ξ to refer to the convex, and discrete sets, respectively. In order to show that the optimal objective value of Problem (10) does not change under the set Ξ^{convex}, first, note that the payoff function is only dependent on the $\sum_{n \in \mathcal{T}} y_n^k, \forall k \in \mathcal{K}$ and not the values of $\boldsymbol{\xi}$. As a result, we only need to prove, for any arbitrary $(\boldsymbol{x}, \boldsymbol{y}^{\mathcal{K}}, \boldsymbol{l})$:

$$\text{if } \Xi^{\text{convex}}(\boldsymbol{x}, \boldsymbol{y}^{\mathcal{K}}, \boldsymbol{l}) \neq \emptyset \Rightarrow \Xi(\boldsymbol{x}, \boldsymbol{y}^{\mathcal{K}}, \boldsymbol{l}) \neq \emptyset.$$

This follows since for cases when both sets are non-empty, for any fix $(\boldsymbol{x}, \boldsymbol{y}^{\mathcal{K}}, \boldsymbol{l})$, the objective values of both problems are equal. Let us choose an arbitrary $(\boldsymbol{x}, \boldsymbol{y}^{\mathcal{K}}, \boldsymbol{l})$. For the sake of conciseness, we drop the dependence on $\boldsymbol{x}$, and $\boldsymbol{y}^{\mathcal{K}}$.

Now, suppose $\tilde{\boldsymbol{\xi}} \in \Xi^{\text{convex}}(\boldsymbol{l})$:

$$y_{l_k}^k > \sum_{n' \in \delta(l_k)} \tilde{\xi}_{n'} x_{n'}, \qquad \text{if } \ \forall k \in \mathcal{K} : l_k > 0, \tag{13}$$

$$y_n^k \leq \sum_{n' \in \delta(n)} \tilde{\xi}_{n'} x_{n'}, \ \forall n \in \mathcal{T}, \qquad \text{if } \ \forall k \in \mathcal{K} : l_k = 0, \tag{14}$$

According to Equation (13), and since $\boldsymbol{x} \geq \boldsymbol{0}$, $\tilde{\boldsymbol{\xi}} \geq \boldsymbol{0}$ and $\boldsymbol{y} \leq \boldsymbol{1}$:

$$l_k > 0 \Rightarrow y_{l_k} = 1, \ \xi_{n'} x_{n'} = 0, \ \forall n' \in \delta(l_k) \Rightarrow$$

$$\xi_{n'} = 0, \ \forall n' \in \delta(l_k) : x_{n'} = 1.$$

Now, we define $\hat{\xi}_n := \lceil \tilde{\xi}_n \rceil, \ \forall n \in \mathcal{N}$, and we show that $\hat{\xi}_n \in \Xi(\boldsymbol{l})$. In order for $\hat{\xi}_n$ to be in $\Xi(\boldsymbol{l})$, it must satisfy the constraints that define the set $\Xi(\boldsymbol{l})$.

$$\forall k : l_k > 0 \ \sum_{n' \in \delta(l_k)} \hat{\xi}_{n'} x_{n'} = \sum_{n' \in \delta(l_k) : x_{n'} = 1} \hat{\xi}_{n'} x_{n'} = \sum_{n' \in \delta(l_k) : x_{n'} = 1} \lceil \tilde{\xi}_n \rceil x_{n'} = 0 \leq y_{l_k}^k. \tag{15}$$

Also,

$$\forall k : l_k = 0 \ \sum_{n' \in \delta(n)} \hat{\xi}_{n'} x_{n'} \geq \sum_{n' \in \delta(n)} \tilde{\xi}_{n'} x_{n'} \geq y_n^k, \ \forall n \in \mathcal{T}. \tag{16}$$

The proof is complete, as we showed $\hat{\boldsymbol{\xi}} \in \Xi(\boldsymbol{l})$.

□

Before reformulating the problem as an MILP, we note that the set described by Equation (11) is not closed. We propose to substitute this set with the following set:

$$\Xi_{\mathrm{c}}(\boldsymbol{x},\boldsymbol{y}^{\mathcal{K}},\boldsymbol{l}) = \left\{ \boldsymbol{\xi}\in\Xi : \begin{array}{ll} y^k_{l_k} \geq \sum\limits_{n'\in\delta(l_k)} \xi_{n'}x_{n'} + 1, & \text{if } l_k>0,\ \forall k\in\mathcal{K} \\ y^k_n \leq \sum\limits_{n'\in\delta(n)} \xi_{n'}x_{n'},\ \forall n\in\mathcal{T}, & \text{if } l_k=0,\ \forall k\in\mathcal{K} \end{array} \right\}. \tag{17}$$

Proposition 6. *Sets $\Xi_c(\boldsymbol{x},\boldsymbol{y},\boldsymbol{l})$ and $\Xi(\boldsymbol{x},\boldsymbol{y},\boldsymbol{l})$ are equal.*

Proof. It suffices to show that if $l_k > 0$:

$$y^k_{l_k} > \sum_{n'\in\delta(l_k)} \xi_{n'}x_{n'} \Leftrightarrow y^k_{l_k} \geq \sum_{n'\in\delta(l_k)} \xi_{n'}x_{n'} + 1.$$

For a given $(\boldsymbol{x},\boldsymbol{y},\boldsymbol{l})$, $\boldsymbol{\xi}$ satisfies the constraint $\left(y^k_{l_k} > \sum_{n'\in\delta(l_k)} \xi_{n'}x_{n'}\right)$ only if $\left(\sum_{n'\in\delta(l_k)} \xi_{n'}x_{n'} = 0\right)$.

The same is true for $\boldsymbol{\xi}$ satisfying the constraint $\left(y^k_{l_k} \geq \sum_{n'\in\delta(l_k)} \xi_{n'}x_{n'} + 1\right)$. Therefore, the two sets are equal. □

Remark 4. This result is stronger than [11] as we are able to obtain an exact reformulation rather than an approximate formulation.

Next, we present the MILP reformulation of Problem (9).

Theorem 1. *Problem* (9) *can be exactly reformulated as the following MILP:*

$$\begin{array}{ll} \max & \tau \\ s.t. & \boldsymbol{x}\in\mathcal{U},\ \boldsymbol{y}^k\in\{0,1\}^N,\ k\in\mathcal{K},\ \tau\in\mathbb{R} \\ & \left.\begin{array}{l} \boldsymbol{\lambda}(\boldsymbol{l})\in\Delta_K(\boldsymbol{l}),\ \boldsymbol{\alpha}(\boldsymbol{l})\in\mathbb{R}^{N+2}_+,\ \boldsymbol{\beta}^k(\boldsymbol{l})\in\mathbb{R}^N_+,\ \forall k\in\mathcal{K},\ \boldsymbol{\nu}(\boldsymbol{l})\in\mathbb{R}^K_+ \\ \tau \leq \sum\limits_{n\in\mathcal{N}} -\alpha_n(\boldsymbol{l}) + (N-J)\alpha_{N+1}(\boldsymbol{l}) - \sum\limits_{\substack{k\in\mathcal{K}\\ l_k\neq 0}} (y^k_{l_k}-1)\nu_k(\boldsymbol{l}) + \dots \\ \quad \dots \sum\limits_{\substack{k\in\mathcal{K}\\ l_k=0}}\sum\limits_{n\in\mathcal{T}} y^k_n\beta^k_n(\boldsymbol{l}) + \sum\limits_{k\in\mathcal{K}}\lambda_k(\boldsymbol{l})\sum\limits_{n\in\mathcal{T}} U_n y^k_n, \\ -\alpha_n(\boldsymbol{l}) + \alpha_{N+1}(\boldsymbol{l}) - \sum\limits_{\substack{k\in\mathcal{K}\\ l_k\neq 0}}\sum\limits_{n'\in\delta(l_k)} x_{n'}\nu_k(\boldsymbol{l}) + \sum\limits_{\substack{k\in\mathcal{K}\\ l_k=0}}\sum\limits_{n'\in\delta(n)} x_{n'}\beta^k_n(\boldsymbol{l}) \leq 0, \forall n\in\mathcal{N}, \end{array}\right\} \forall \boldsymbol{l}\in\partial\mathcal{L} \\ & \left.\begin{array}{l} \boldsymbol{\alpha}(\boldsymbol{l})\in\mathbb{R}^{N+1}_+,\ \boldsymbol{\nu}(\boldsymbol{l})\in R^K_+ \\ \sum\limits_{n\in\mathcal{N}} -\alpha_n(\boldsymbol{l}) + (N-J)\alpha_{N+1}(\boldsymbol{l}) - \sum\limits_{\substack{k\in\mathcal{K}\\ l_k\neq 0}} (y^k_{l_k}-1)\nu_k(\boldsymbol{l}) \geq 1 \\ -\alpha_n(\boldsymbol{l}) + \alpha_{N+1}(\boldsymbol{l}) - \sum\limits_{\substack{k\in\mathcal{K}\\ l_k\neq 0}}\sum\limits_{n'\in\delta(l_k)} x_{n'}\nu_k(\boldsymbol{l}) = 0,\ \forall n\in\mathcal{N} \end{array}\right\} \forall \boldsymbol{l}\in\mathcal{L}_+ \end{array} \tag{18}$$

Proof. This result follows from Proposition 5 and 6 and derivation in [11] . In order to make the paper self-contained, we will provide the full derivation in Appendix B. □

Remark 5. For a fixed K, the size of the above MILP is polynomial in all problem inputs, thus, it circumvents the exponentiality of the attacker's action set.

4 Results

In this section, we present different numerical results that demonstrate the performance of K-adaptability, in terms of both computation effort and approximation quality. We use randomly generated graphs, where an edge between a monitor and a target exists with probability $P = 0.2$. Our results are averaged over 20 sample networks. In all experiments, there is a time limit of 60 minutes. Also, all the targets are assumed to have equal value. This assumption is to facilitate the interpretation of the results.

In our experiments, we compare our approach against an exact scenario-based MILP solution, which explicitly enumerates the adversary's actions and solves for the best defender strategy against the worst-case attacker action. The formulation for the scenario-based problem is presented in Appendix A. We also compare our approach to the greedy-based algorithm by Tzoumas et.al [17].

Optimal Coverage vs. K [N=20, T=5, I=8, J=5]: The first experiment compares the optimal solution of the K-adaptability problem, for various values of K, with the exact solution. Both problems use greedy solution as warm-start. In Figure 4, the vertical axis shows the normalized coverage (optimal coverage divided by the total number of targets). The first three bars in this plot are the

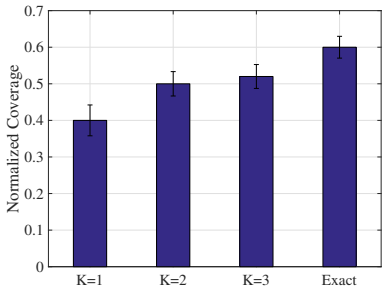

Fig. 4. Coverage vs. K

optimal coverage results from 1-, 2-, and 3-adaptability problems, and the last bar corresponds to the exact solution. Here, we can observe that by increasing K from 1 to 2 and 3, the optimality gap monotonically decreases, where for $K = 3$, this gap is less that 10%.

Coverage/Solver Time vs. Number of Attacks [N=30, T=8, I=8, K=2] We now investigate how our approach performs, both in terms of solver-time and solution quality, compared to the exact approach. Figure 4(a) shows the normalized coverage, plotted versus different numbers of adversarial attacks (J). The blue and yellow bars are the results of the 2-adaptability and exact problems, respectively. We observe that as J increases, the coverage decreases, until $J = 8$ for which the exact formulation could not find a feasible solution within the time-budget. This is because, going beyond $J = 7$, the number of attack scenarios, i.e., the number of constraints, becomes very large. For example, for $(J = 8)$, there were $\binom{30}{8} \approx 6 \times 10^6$ constraints and the solver did not obtain a feasible solution within the 1-hour time budget. However, we observe that the 2-adaptability solution does not suffer from this issue, as it is able to solve for such cases. Also, for $(J < 8)$, it closely approximates the exact solution.

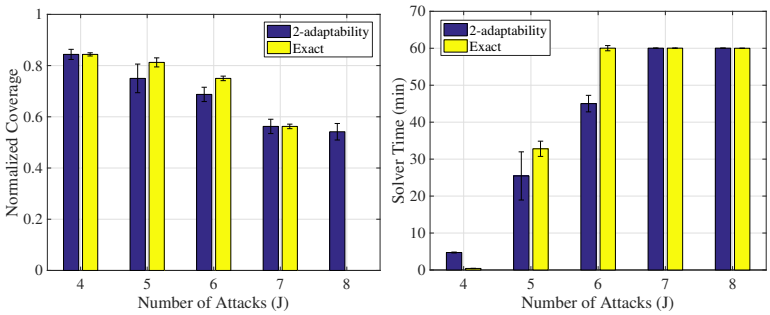

Fig. 5. Comparing solver time and coverage for exact and 2-adaptability problems

Figure 4(b) compares the solver time of the 2-adaptability and exact solution. We observe that the exact solution quickly becomes intractable as J increases. For larger J, both problems reach the time limit, however, as Figure 4(a) suggests, for $(J > 7)$ the exact approach fails to provide a solution within the time limit, whereas the 2-adaptability problem yields a high quality solution.

K-adaptability vs. Greedy [N=20, T=5, I=8, J=4, K=2]: In this experiment, we test our approach on harder graph instances, graphs with solutions that are hard for heuristic algorithms to find, and we average over 10 such graphs. For instance, see Figure 2. In our comparison, we use the greedy algorithm proposed in [17] as the baseline. There are several works on greedy solutions, however, most of them are limited in terms of allowable ranges for J [14, 7]. Thus, we compare our solution to the work of [17], since it applies to all regimes of J. Figure 6 shows the normalized coverage, for greedy, $1-$ and $2-$adaptability problems. This result indicates that by K as low as 1, on average we are able to recover the greedy solution, where the $2-$adaptability significantly outperforms

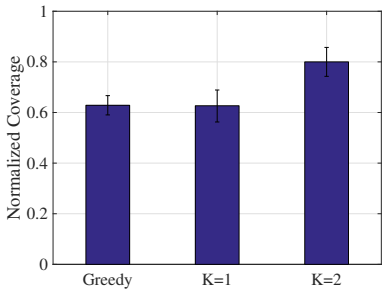

Fig. 6. Comparing Greedy Solution with 1- and 2-Adaptability Problems

the greedy solution. As a result, this experiment illustrates that on hard graph instances that greedy does not perform well, K-adaptability can outperform, using only small values of K.

5 Conclusion

This work studies a Stackelberg game model for the strategic monitoring problem. This problem is highly intractable. Thus, we provide a tractable approximation scheme based on K-adaptability formulation. Our solution methodology automatically partitions the set of adversary's strategies and maps each subset to a coverage policy. These policies are such that they do not overestimate the defender's payoff. We show that there exists an exact MILP reformulation of the K-adaptability problem whose size grows polynomially in the description of the problem input. We empirically show, the shortcomings of both the heuristic and exact approaches and that K-adaptability can remedy those issues. In particular, our experiments indicate that with even with small values of K, ranging from 1, to 3, K-adaptability recovers both the greedy and exact solutions.

Acknowledgement

This work was supported by the Army Research Office (W911NF-17-1-0370, W911NF-15-1-0515, W911NF-16-1-0069), National Science Foundation (CNS-1640624, IIS-1649972, and IIS-1526860), Office of Naval Research (N00014-15-1-2621), and the USC Office of the Provost and USC Viterbi School of Engineering.

A Exact Scenario-based MILP

In Problem (5), the optimal pure strategy for the defender can be obtained from the solution of the following deterministic MILP problem which enumerates all the attacker pure strategies. This reformulation is exact, however, it requires a

number of variables and constraints which is exponential in N. In this formulation $y_{\boldsymbol{\xi},n}$ is a binary variable and it is equal to 1 iff under attack scenario $\boldsymbol{\xi}$, target n is covered.

$$\begin{aligned} \max_{\substack{\boldsymbol{x}\in\mathcal{U}\\ \boldsymbol{y}\in\{0,1\}^{|\Xi|\times N}}} \quad & \tau \\ \text{s.t.} \quad & y_{\boldsymbol{\xi},n} \leq \sum_{n'\in\delta(n)} \xi_{n'} x_{n'},\ \forall n\in\mathcal{N},\ \forall \boldsymbol{\xi}\in\Xi \\ & \tau \leq \sum_{n\in\mathcal{N}} y_{\boldsymbol{\xi},n},\ \forall \boldsymbol{\xi}\in\Xi \end{aligned} \tag{19}$$

B Exact MILP Formulation of the K-Adaptability

The following reformulation is based on [11]. The objective function of the Problem (10) is identical to:

$$\min_{\boldsymbol{l}\in\mathcal{L}} \min_{\boldsymbol{\xi}\in\Xi_{\mathrm{c}}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{l})} \left[\max_{\boldsymbol{\lambda}\in\Delta_K(\boldsymbol{l})} \sum_{k\in\mathcal{K}} \lambda_k \sum_{n\in\mathcal{T}} U_n y_n^k \right], \tag{20}$$

where $\Delta_K(\boldsymbol{l}) = \{\boldsymbol{\lambda}\in\mathbb{R}_+ : \boldsymbol{e}^\top\boldsymbol{\lambda} = 1, \lambda_k = 0, \forall k\in\mathcal{K} : l_k \neq 0\}$. We define $\partial\mathcal{L} := \{\boldsymbol{l}\in\mathcal{L} : \boldsymbol{l} \not\geq \boldsymbol{0}\}$, and $\mathcal{L}_+ := \{\boldsymbol{l}\in\mathcal{L} > \boldsymbol{0}\}$. Note that $\Delta_K(\boldsymbol{l}) = \emptyset$ if and only if $\boldsymbol{l} > \boldsymbol{0}$. If $\Xi_{\mathrm{c}}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{l}) = \emptyset$ for all $\boldsymbol{l}\in\mathcal{L}_+$, then the problem is equivalent to:

$$\min_{\boldsymbol{l}\in\partial\mathcal{L}} \min_{\boldsymbol{\xi}\in\Xi_{\mathrm{c}}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{l})} \left[\max_{\boldsymbol{\lambda}\in\Delta_K(\boldsymbol{l})} \sum_{k\in\mathcal{K}} \lambda_k \sum_{n\in\mathcal{T}} U_n y_n^k \right]. \tag{21}$$

By applying the classical min-max theorem:

$$\min_{\boldsymbol{l}\in\partial\mathcal{L}} \max_{\substack{\boldsymbol{\lambda}\in\Delta_K(\boldsymbol{l})\\ \boldsymbol{l}\in\partial\mathcal{L}}} \min_{\boldsymbol{\xi}\in\Xi_{\mathrm{c}}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{l})} \sum_{k\in\mathcal{K}} \lambda_k \sum_{n\in\mathcal{T}} U_n y_n^k. \tag{22}$$

This problem is also equivalent to:

$$\max_{\substack{\boldsymbol{\lambda}(\boldsymbol{l})\in\Delta_K(\boldsymbol{l})\\ \boldsymbol{l}\in\partial\mathcal{L}}} \min_{\boldsymbol{l}\in\partial\mathcal{L}} \min_{\boldsymbol{\xi}\in\Xi_{\mathrm{c}}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{l})} \sum_{k\in\mathcal{K}} \lambda_k(\boldsymbol{l}) \sum_{n\in\mathcal{T}} U_n y_n^k. \tag{23}$$

We note that if $\Xi_{\mathrm{c}}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{l}) \neq \emptyset$, for some $\boldsymbol{l}\in\mathcal{L}_+$ the objective of Problem (10) evaluates to $-\infty$. Using the epigraph form, Problem (10) is equivalent to:

$$\begin{aligned} \max\ & \tau \\ \text{s.t.}\ & \boldsymbol{x}\in\mathcal{U}, \boldsymbol{y}^k\in\{0,1\}^N, k\in\mathcal{K} \\ & \tau\in\mathbb{R},\ \boldsymbol{\lambda}(\boldsymbol{l})\in\Delta_K(\boldsymbol{l}),\ \boldsymbol{l}\in\partial\mathcal{L} \\ & \tau \leq \sum_{k\in\mathcal{K}} \lambda_k(\boldsymbol{l}) \sum_{n\in\mathcal{T}} U_n y_n^k, && \forall \boldsymbol{l}\in\partial\mathcal{L},\ \boldsymbol{\xi}\in\Xi_{\mathrm{c}}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{l}) \\ & \Xi_{\mathrm{c}}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{l}) = \emptyset, && \forall \boldsymbol{l}\in\mathcal{L}_+. \end{aligned} \tag{24}$$

The semi-infinite constraint associated with $\boldsymbol{l} \in \partial\mathcal{L}$ is satisfied if and only if:

$$\begin{aligned}
\min\ & \sum_{k\in\mathcal{K}} \lambda_k(\boldsymbol{l}) \sum_{n\in\mathcal{T}} U_n y_n^k \\
\text{s.t. } & 0 \le \xi_{n'} \le 1,\ \forall n' \in \mathcal{N} \\
& \sum_{n'\in\mathcal{N}} \xi_{n'} \ge N - J \\
& y_{l_k}^k \ge \sum_{n'\in\delta(l_k)} \xi_{n'} x_{n'} + 1, \qquad \text{if } l_k > 0,\ \forall k \in \mathcal{K} \\
& y_n^k \le \sum_{n'\in\delta(n)} \xi_{n'} x_{n'},\ \forall n \in \mathcal{T},\ \text{if } l_k = 0,\ \forall k \in \mathcal{K}
\end{aligned} \tag{25}$$

is greater than τ.

In order to obtain the dual formulation, we introduce an auxiliary variable $\xi_{T+1} = 1$, and we rewrite the objective as: $(\sum_{k\in\mathcal{K}} \lambda_k(\boldsymbol{l}) \sum_{n\in\mathcal{T}} U_n y_n^k)\ \xi_{T+1}$. Using strong linear programming duality:

$$\begin{aligned}
\max\ & \sum_{n\in\mathcal{N}} -\alpha_n(\boldsymbol{l}) + (N-J)\alpha_{N+1}(\boldsymbol{l}) - \sum_{\substack{k\in\mathcal{K}\\ l_k\neq 0}} (y_{l_k}^k - 1)\nu_k(\boldsymbol{l}) + \sum_{\substack{k\in\mathcal{K}\\ l_k=0}} \sum_{n\in\mathcal{T}} y_n^k \beta_n^k(\boldsymbol{l}) + \alpha_{N+2}(\boldsymbol{l}) \\
\text{s.t. } & \alpha_n(\boldsymbol{l}) \ge 0, n \in \{1,\dots,N+1\},\ \boldsymbol{\beta}^k(\boldsymbol{l}) \in \mathbb{R}_+^N,\ \forall k \in \mathcal{K},\ \boldsymbol{\nu}(\boldsymbol{l}) \in \mathbb{R}_+^K \\
& -\alpha_n(\boldsymbol{l}) + \alpha_{N+1}(\boldsymbol{l}) - \sum_{\substack{k\in\mathcal{K}\\ l_k\neq 0}} \sum_{n'\in\delta(l_k)} x_{n'}\nu_k(\boldsymbol{l}) + \sum_{\substack{k\in\mathcal{K}\\ l_k=0}} \sum_{n'\in\delta(n)} x_{n'}\beta_n^k(\boldsymbol{l}) \le 0, \forall n \in \mathcal{T}, \\
& \alpha_{N+2}(\boldsymbol{l}) = \sum_{k\in\mathcal{K}} \lambda_k(\boldsymbol{l}) \sum_{n\in\mathcal{T}} U_n y_n^k.
\end{aligned} \tag{26}$$

Also, the last constraint in formulation (24) is satisfied if the following linear program is infeasible:

$$\begin{aligned}
\min\ & 0 \\
\text{s.t. } & 0 \le \xi_n \le 1, \qquad \forall n \in \mathcal{N} \\
& \sum_{n\in\mathcal{N}} \xi_n \ge N - J \\
& y_{l_k}^k \ge \sum_{n'\in\delta(l_k)} \xi_{n'} x_{n'} + 1,\ \forall k \in \mathcal{K},\ l_k \neq 0.
\end{aligned} \tag{27}$$

Using strong duality, this occurs if the dual problem is unbounded. Since the feasible region of the dual problem constitutes a cone, the dual problem is unbounded if and only if there is a feasible solution with an objective value of 1 or more. The dual problem is as below:

$$\begin{aligned}
\max\ & \sum_{n\in\mathcal{N}} -\alpha_n(\boldsymbol{l}) + (N-J)\alpha_{N+1}(\boldsymbol{l}) - \sum_{\substack{k\in\mathcal{K}\\ l_k\neq 0}} (y_{l_k}^k - 1)\nu_k(\boldsymbol{l}) \\
\text{s.t. } & \boldsymbol{\alpha}(\boldsymbol{l}) \in \mathbb{R}_+^{N+1},\ \boldsymbol{\nu}(\boldsymbol{l}) \in \mathbb{R}_+^K \\
& -\alpha_n(\boldsymbol{l}) + \alpha_{N+1}(\boldsymbol{l}) - \sum_{\substack{k\in\mathcal{K}\\ l_k\neq 0}} \sum_{n'\in\delta(l_k)} x_{n'}\nu_k(\boldsymbol{l}) = 0,\ \forall n \in \mathcal{N}
\end{aligned} \tag{28}$$

References

1. Bard, N., Nicholas, D., Szepesvaári, C., Bowling, M.: Decision-theoretic clustering of strategies. In: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems. pp. 17–25. International Foundation for Autonomous Agents and Multiagent Systems (2015)
2. Basak, A.: Abstraction using analysis of subgames. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. pp. 4196–4197. AAAI Press (2016)
3. Ben-Tal, A., El Ghaoui, L., Nemirovski, A.: Robust optimization. Princeton University Press (2009)
4. Ben-Tal, A., Goryashko, A., Guslitzer, E., Nemirovski, A.: Adjustable robust solutions of uncertain linear programs. Mathematical Programming **99**(2), 351–376 (2004)
5. Bertsimas, D., Brown, D.B., Caramanis, C.: Theory and applications of robust optimization. SIAM review **53**(3), 464–501 (2011)
6. Bertsimas, D., Caramanis, C.: Finite adaptability in multistage linear optimization. IEEE Transactions on Automatic Control **55**(12), 2751–2766 (2010)
7. Bogunovic, I., Mitrović, S., Scarlett, J., Cevher, V.: Robust submodular maximization: A non-uniform partitioning approach. arXiv preprint arXiv:1706.04918 (2017)
8. Bošanskỳ, B., Jiang, A.X., Tambe, M., Kiekintveld, C.: Combining compact representation and incremental generation in large games with sequential strategies. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. pp. 812–818. AAAI Press (2015)
9. Brown, G., Carlyle, M., Salmerón, J., Wood, K.: Defending critical infrastructure. Interfaces **36**(6), 530–544 (2006)
10. Dahan, M., Sela, L., Amin, S.: Network monitoring under strategic disruptions. arXiv preprint arXiv:1705.00349 (2017)
11. Hanasusanto, G.A., Kuhn, D., Wiesemann, W.: K-adaptability in two-stage robust binary programming. Operations Research **63**(4), 877–891 (2015)
12. Jain, M., Korzhyk, D., Vaněk, O., Conitzer, V., Pěchouček, M., Tambe, M.: A double oracle algorithm for zero-sum security games on graphs. In: The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 1. pp. 327–334 (2011)
13. Krause, A., McMahan, H.B., Guestrin, C., Gupta, A.: Robust submodular observation selection. Journal of Machine Learning Research **9**(Dec), 2761–2801 (2008)
14. Orlin, J.B., Schulz, A.S., Udwani, R.: Robust monotone submodular function maximization. In: International Conference on Integer Programming and Combinatorial Optimization. pp. 312–324. Springer (2016)
15. Pita, J., Tambe, M., Kiekintveld, C., Cullen, S., Steigerwald, E.: Guards: game theoretic security allocation on a national scale. In: The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 1. pp. 37–44. International Foundation for Autonomous Agents and Multiagent Systems (2011)
16. Tsai, J., Kiekintveld, C., Ordonez, F., Tambe, M., Rathi, S.: Iris-a tool for strategic security allocation in transportation networks (2009)
17. Tzoumas, V., Gatsis, K., Jadbabaie, A., Pappas, G.J.: Resilient monotone submodular function maximization. arXiv preprint arXiv:1703.07280 (2017)
18. Vayanos, P., Kuhn, D., Rustem, B.: Decision rules for information discovery in multi-stage stochastic programming. In: Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on. pp. 7368–7373. IEEE (2011)