Optimal Cybersecurity Investments for SIS Model

Van Sy Mai
ANTD, NIST
Gaithersburg, MD USA
vansy.mai@nist.gov

Richard J. La
ACMD, NIST & Univ. of MD
College Park, MD USA
hyongla@umd.edu

Abdella Battou
ANTD, NIST
Gaithersburg, MD USA
abdella.battou@nist.gov

***Abstract*—We study the problem of minimizing the (time) average security costs in large systems comprising many interdependent subsystems, where the state evolution is captured by a susceptible-infected-susceptible (SIS) model. The security costs reflect security investments, economic losses and recovery costs from infections and failures following successful attacks. We show that the resulting optimization problem is non-convex and propose two algorithms – one for solving a convex relaxation, and the other for finding a local minimizer, based on a reduced gradient method. Also, we provide a sufficient condition under which the convex relaxation is exact and its solution coincides with that of the original problem. Numerical results are provided to validate our analytical results and to demonstrate the effectiveness of the proposed algorithms.**

***Index Terms*—Cybersecurity investments; Optimization; SIS model**

I. Introduction

Complex modern engineering systems, such as information and communication networks and power systems, consist of many interdependent systems. In order to deliver their services, the comprising systems must work together and oftentimes support each other. Unfortunately, the interdependence makes it possible for a local failure and infection of a system by malware to spread to other systems. Similarly, contagious diseases spread via contacts in social networks. From this viewpoint, it is clear that any investment in security of the complex system or the control of disease spread should take into account the interdependence in the systems and social contacts in order to achieve the best benefits.

There is already a large volume of literature that examines how to optimize the (security) investments in complex systems or the mitigation of disease spread. For example, in [8]–[10], [12], researchers adopted a game theoretic formulation to study the problem of security investments with distributed agents. In another line of research, which is more closely related to our study, researchers investigated optimal strategies using vaccines/immunization (prevention) [4], [22], antidotes or curing rates (recovery) [2], [15], [19] or a combination of both preventive and recovery measures [17], [23].

A key difference between these studies and ours is that, unlike the previous studies that focus on the exponential decay rate to disease-free state as a key performance metric, we are interested in minimizing the (time) average costs of a system operator. The system costs account for both security investments and recovery/repair costs ensuing infections or failures, which we call *infection costs*, under the assumption that attacks arrive according to some stochastic process. Security investments shape the breach probability of comprising systems, i.e., the probability that the systems fall victim to attacks and become infected or fail. As attacks do not stop in our setting, in general it is not possible to achieve infection-free state at steady state, and the exponential decay rate is no longer a suitable performance metric.

This difference leads to a non-convex optimization problem that cannot be solved easily, whereas several of previous studies led to convex optimization problems, e.g., semidefinite program in [19], [22] and geometric program in [15], [17], [23]. To obtain a good solution to our problem, we first propose a convex relaxation, the optimal value of which provides a lower bound to that of our problem. We also derive a sufficient condition for the convex relaxation to be exact (Corollary 1 below), in which case the solution of the convex relaxation solves our problem. In addition, we propose a reduced gradient-based algorithm (RGA) that produces a local minimizer. Together, these two approaches offer a bound on the optimality gap. Our numerical studies show that when the infection costs are large, the sufficient condition for the convex relaxation to be exact holds. Regrettably, solving the convex relaxation does not scale well with the system size. However, the gap between the optimal value of convex relaxation and the value achieved by a local minimizer produced by the RGA tends to be small, suggesting that we can use the RGA to find a good solution to our problem.

The rest of the paper is organized as follows: Section II explains the notation and terminology we adopt. Section III describes the setup and the problem formulation, including the optimization problem. We describe the proposed algorithms for finding a lower bound and an upper bound on the optimal value of the optimization problem in Sections IV and V, respectively. Section VI presents some numerical results. All technical proofs are omitted and can be found in [16].

II. Preliminaries

A. Notation and Terminology

We use $\mathbb{R}$ and $\mathbb{R}_+$ to denote the set of real numbers and nonnegative real numbers, respectively. For a finite set $\mathcal{A}$, $|\mathcal{A}|$ denotes its cardinality. For a matrix $A = [a_{i,j}]$, let $a_{i,j}$ denote its (i,j) element, A^T its transpose, $\rho(A)$ its spectral radius, and $\underline{\sigma}(A)$ and $\bar{\sigma}(A)$ its smallest and largest real parts of its eigenvalues. For two matrices A and B, we write $A \geq B$ if $A - B$ is a nonnegative matrix (i.e., a matrix with

all nonnegative elements). We use boldface letters to denote vectors, e.g., $\mathbf{x} = [x_1, ..., x_n]^\mathsf{T}$ and $\mathbf{1} = [1, ..., 1]^\mathsf{T}$. For any two vectors $\mathbf{x}$ and $\mathbf{y}$ of the same dimension, $\mathbf{x} \circ \mathbf{y}$ and $\frac{\mathbf{x}}{\mathbf{y}}$ are their element-wise product and division, respectively. If $\mathbf{x} \in \mathbb{R}^n$, $\mathtt{diag}(\mathbf{x}) \in \mathbb{R}^{n\times n}$ denotes the diagonal matrix with diagonal elements $x_1, \ldots, x_n$. A directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of nodes $\mathcal{V}$ and a set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ of directed edges. A directed path is a sequence of edges in the form $\big((i_1, i_2), (i_2, i_3), ..., (i_{k-1}, i_k)\big)$. A graph is strongly connected if there is a directed path from each node to any other node.

B. M-Matrix Theory

A matrix $A \in \mathbb{R}^{n\times n}$ is an M-matrix if it can be expressed as $A = sI - B$, where $B = [b_{i,j}] \in \mathbb{R}_+^{n\times n}$ and $s \geq \rho(B)$. The set of (nonsingular) $n \times n$ M-matrices is denoted by ($\mathbb{M}_+^{n\times n}$) $\mathbb{M}^{n\times n}$. Note that this definition implies that the off-diagonal elements of A are nonpositive and the diagonal elements are nonnegative; any matrix satisfying these conditions is called a Z-matrix. We shall make use of the following results (see, e.g, [21]) on the properties of a nonsingular M-matrix.

Lemma 1. *Let $A \in \mathbb{R}^{n\times n}$ be a Z-matrix. A is a nonsingular M-matrix if and only if one of the following conditions holds:*

(a) *$A + D$ is nonsingular for every diagonal $D \in \mathbb{R}_+^{n\times n}$.*
(b) *A is inverse-positive, i.e., $\exists A^{-1} \in \mathbb{R}_+^{n\times n}$.*
(c) *A is monotone, i.e., $A\mathbf{x} \geq 0 \Rightarrow \mathbf{x} \geq 0, \forall \mathbf{x} \in \mathbb{R}^n$.*
(d) *Every regular splitting of A is convergent, i.e., if $A = M - N$ with $M^{-1}, N \in \mathbb{R}_+^{n\times n}$, then $\rho(M^{-1}N) < 1$.*
(e) *A is positive stable, i.e., $\underline{\sigma}(A) > 0$.*
(f) *$\exists \mathbf{x} > \mathbf{0}$ with $A\mathbf{x} \geq \mathbf{0}$ such that if $[A\mathbf{x}]_{i_0} = 0$, then $\exists i_1, ..., i_r$ with $a_{i_{k-1},i_k} \neq 0$ for $1 \leq k \leq r$ and $[A\mathbf{x}]_{i_r} > 0$.*
(g) *$\exists \mathbf{x} > \mathbf{0}$ with $A\mathbf{x} > \mathbf{0}$.*

The next result is a direct consequence of [13, Thm. 2].

Lemma 2. *Let $A \in \mathbb{M}^{n\times n}$ be irreducible. Then*

(i) *$\mathtt{diag}(\mathbf{z}) + A \in \mathbb{M}_+^{n\times n}$ for every $\mathbf{z} \in \mathbb{R}_+^n \setminus \{\mathbf{0}\}$.*
(ii) *$\big[\big(\mathtt{diag}(\mathbf{z}) + A\big)^{-1}\big]_{i,j}$ is a convex and decreasing function in $\mathbf{z} \in \mathbb{R}_+^n$ for all $1 \leq i, j \leq n$.*

III. Model and Formulation

Consider a large system consisting of N (component or sub-)systems, and denote the set of systems by $\mathcal{A} := \{1, 2, \ldots, N\}$. The security of the systems is interdependent in that the failure or infection of a system can cause that of other systems. As stated earlier, we study the problem of determining security investments for hardening or prevention of each system in order to defend the systems against attacks in large, complex systems, in which the comprising systems depend on each other for their function.[1] The goal of the system operator is to minimize the aggregate cost for all systems (per unit time), which accounts for both security investments and economic losses from failures/infections of systems.

[1]Throughout the paper, we shall use the words 'failure' and 'infection' interchangeably, in order to indicate that a system fell victim to an attack.

A. Setup

We assume that each system experiences direct attacks from malicious actors. Direct attacks on system $i \in \mathcal{A}$ occur in accordance with a Poisson process with rate $\lambda_i \in \mathbb{R}_+$. When a system experiences an attack, it suffers a failure and subsequent economic losses with some probability, called *breach probability*.

This breach probability depends on the security investment on the system: let $s_i \in \mathbb{R}_+$ be the security investment on system i. The breach probability of system i is determined by some function $q_i : \mathbb{R}_+ \to (0, 1]$. In other words, when the operator invests s_i on system i, its breach probability is equal to $q_i(s_i)$. We assume that q_i is strictly decreasing and continuously differentiable for all $i \in \mathcal{A}$.

When system i falls victim to an attack and becomes infected, the operator incurs costs c_i^r per unit time for recovery (e.g., inspection and repair of servers). Recovery times are modeled using independent and identically distributed (i.i.d.) exponential random variables with parameter $\delta_i > 0$. Besides recovery costs, the failure of system i may cause additional economic losses if, for example, some servers in system i have to be taken offline for inspection and repair and are inaccessible during the period to other systems that depend on the servers. To model this, we assume that the infection of system i introduces economic losses of c_i^e per unit time.

Besides the direct attacks, systems also experience indirect attacks from other failed or infected systems. For example, this can model the spread of virus/malware or failures in complex systems. The rate at which the infection of system i causes that of another system j is denoted by $\beta_{j,i} \in \mathbb{R}_+$. When $\beta_{j,i} > 0$, we say that system i supports system j or, equivalently, system j depends on system i. Let $B = [b_{i,j};\ i, j \in \mathcal{A}]$ be an $N \times N$ matrix that describes the infection rates among systems, where the element $b_{i,j}$ is equal to $\beta_{i,j}$. We adopt the convention $\beta_{i,i} = 0$ for all $i \in \mathcal{A}$.

Define a directed graph on $\mathcal{G} = (\mathcal{A}, \mathcal{E})$, where a directed edge from system i to system j, denoted by $\langle i, j\rangle$, belongs to the edge set $\mathcal{E}$ if and only if $\beta_{j,i} > 0$. We denote the associated adjacency matrix by A, i.e., $a_{i,j} = 1$ if $\beta_{i,j} > 0$ and $a_{i,j} = 0$ otherwise. Throughout the paper, we will assume that the nonnegative adjacency matrix A (or, equivalently, B) is irreducible. Note that this is equivalent to assuming that the graph $\mathcal{G}$ is strongly connected.

B. Model

We adopt the susceptible-infected-susceptible (SIS) model to capture the evolution of system state. Let $p_i(t)$ be the probability that system i is at the 'infected' state (I) at time $t \in \mathbb{R}_+$. We approximate the dynamics of $\mathbf{p}(t) := (p_i(t); i \in \mathcal{A})$, $t \in \mathbb{R}_+$, using the following differential equations, which are similar to those employed in [7], [15], [17], [19], [22], [23]: for fixed security investments, $\mathbf{s} = (s_i;\ i \in \mathcal{A}) \in \mathbb{R}_+^N$,

$$\dot{p}_i(t) = (1 - p_i(t))q_i(s_i)\Big(\lambda_i + \sum_{j\in\mathcal{A}} \beta_{i,j}p_j(t)\Big) - \delta_i p_i(t). \quad (1)$$

In practice, the breach probability q_i can be a complicated function of the security investment. However, it has been shown that, under some conditions, the breach probability function is decreasing and log-convex [1]. Here, in order to make progress, we assume that the breach probability functions can be approximated (in the regime of interest) using a log-convex function of the form $q_i(s) = (1+\kappa_i s)^{-1}$ for all $i \in \mathcal{A}$. The parameter $\kappa_i > 0$ models how quickly the breach probability decreases with security investment for system i.

Define $\alpha_i := \kappa_i \delta_i$, $i \in \mathcal{A}$, and $\boldsymbol{\alpha} := (\alpha_i;\ i \in \mathcal{A})$. The following theorem tells us that, for a fixed security investment vector $\mathbf{s} := (s_i;\ i \in \mathcal{A}) \in \mathbb{R}^N_+$, there exists a unique equilibrium of the differential system given by (1).

Theorem 1. *Suppose $\boldsymbol{\lambda} \gneq \mathbf{0}$, $\boldsymbol{\delta} > \mathbf{0}$ and $\mathbf{s} \geq \mathbf{0}$ are fixed. If the network is strongly connected, i.e., B is irreducible, there exists a unique equilibrium $\mathbf{p}^* \in (0,1)^N$ of (1). Moreover, starting with any $\mathbf{p}_0$ satisfying $\mathbf{p}^* \leq \mathbf{p}_0 \leq \mathbf{1}$, the iteration*

$$\mathbf{p}_{k+1} = \frac{\boldsymbol{\lambda} + B\mathbf{p}_k}{\boldsymbol{\lambda} + B\mathbf{p}_k + \boldsymbol{\alpha} \circ \mathbf{s} + \boldsymbol{\delta}},\ k \in \mathbb{N}, \tag{2}$$

converges linearly to $\mathbf{p}^*$ with some rate $\rho_0 < 1 - \min_{i\in\mathcal{A}} p_i^*$.

Note that the unique equilibrium of the differential system described by (1) specifies the probability that each system will be infected at steady state. For this reason, we take the average cost of the system, denoted by $C_{\rm avg}(\mathbf{s})$, to be

$$C_{\rm avg}(\mathbf{s}) := w(\mathbf{s}) + \textstyle\sum_{i\in\mathcal{A}} c_i p_i^*(\mathbf{s}) = w(\mathbf{s}) + \mathbf{c}^{\mathsf{T}}\mathbf{p}^*(\mathbf{s}),$$

where $c_i := c_i^r + c_i^e$, $\mathbf{c} = (c_i;\ i \in \mathcal{A})$, and $w(\mathbf{s})$ quantifies the security investment costs, e.g., $w(\mathbf{s}) = \sum_{i\in\mathcal{A}} s_i$. In the remainder of the paper, we assume that w is convex and strictly increasing.

Equipped with Theorem 1, we can formulate the problem of determining optimal security investments that minimize the average cost $C_{\rm avg}(\mathbf{s})$ as follows:

$$\begin{aligned} (\mathrm{P}) \qquad \min_{\mathbf{s}\in\mathbb{R}^N_+, \mathbf{p}\in\mathbb{R}^N_+} \quad & f(\mathbf{s},\mathbf{p}) := w(\mathbf{s}) + \mathbf{c}^{\mathsf{T}}\mathbf{p} && (3)\\ \text{s.t.} \quad & \mathbf{g}(\mathbf{s},\mathbf{p}) = \mathbf{0} && (4)\end{aligned}$$

where $\mathbf{g}(\mathbf{s},\mathbf{p}) = (g_i(\mathbf{s},\mathbf{p});\ i \in \mathcal{A})$ and

$$g_i(\mathbf{s},\mathbf{p}) = (1-p_i)\big(\lambda_i + \sum_{j\in\mathcal{A}} \beta_{i,j} p_j\big) - (\alpha_i s_i + \delta_i) p_i,\ i \in \mathcal{A}.$$

Recall that, for given $\mathbf{s} \in \mathbb{R}^N_+$, only the unique equilibrium $\mathbf{p}^* \in (0,1)^N$ in Theorem 1 satisfies the constraint in (4).

This problem (P) is nonconvex due to the nonconvexity of the equality constraint functions in (4). In particular, g_i contains both quadratic or bilinear terms $p_i p_j$ and $p_i s_i$. In the following sections, we develop two algorithms: the first is based on a convex relaxation and provides a lower bound on the optimal value of the problem (P). The other is based on the reduced gradient method and offers an upper bound on the optimal value. The difference between two bounds gives us a quantitative measure of optimality gap.

IV. Convex Relaxation and Lower Bounds

In this section, we describe a convex relaxation of the problem (P) and provide a sufficient condition for it to be exact, i.e., its optimal point is also an optimal point of (P).

A. Convex Relaxation

Given $\boldsymbol{\lambda} \gneq \mathbf{0}$ and irreducible B, Theorem 1 tells us that the unique equilibrium of (1) which satisfies (4) is strictly positive. As a result, we can rewrite the constraints in (4) as

$$(\mathbf{p}^{-1} - \mathbf{1}) \circ (\boldsymbol{\lambda} + B\mathbf{p}) = \boldsymbol{\alpha} \circ \mathbf{s} + \boldsymbol{\delta}, \tag{5}$$

where $\mathbf{p}^{-1} = (p_i^{-1};\ i \in \mathcal{A})$. By introducing a new variable

$$\mathbf{z} := \mathbf{p}^{-1} \circ (\boldsymbol{\lambda} + B\mathbf{p}), \tag{6}$$

the constraint in (5) can be rewritten as

$$\mathbf{z} = \boldsymbol{\alpha} \circ \mathbf{s} + \boldsymbol{\delta} + \boldsymbol{\lambda} + B\mathbf{p}. \tag{7}$$

Note that (7) is affine in $\mathbf{z}$, $\mathbf{s}$ and $\mathbf{p}$, and the non-convexity in the equality constraints (mentioned at the end of the previous section) is now captured by $\mathbf{z}$, which can be expressed as

$$(\mathtt{diag}(\mathbf{z}) - B)\mathbf{p} = \boldsymbol{\lambda} \gneq \mathbf{0}. \tag{8}$$

Since B is irreducible, Lemma 1-(f) tells us that the constraints in (8) and $\mathbf{p} > 0$ imply that $(\mathtt{diag}(\mathbf{z}) - B)$ is a nonsingular M-matrix and $\mathbf{p} = (\mathtt{diag}(\mathbf{z}) - B)^{-1}\boldsymbol{\lambda}$. As a result, the original problem can be reformulated as

$$\begin{aligned} (\mathrm{P2}) \qquad \min_{\mathbf{s},\mathbf{p},\mathbf{z}} \quad & f(\mathbf{s},\mathbf{p}) \\ \text{s.t.} \quad & \mathbf{p} = (\mathtt{diag}(\mathbf{z}) - B)^{-1}\boldsymbol{\lambda} \\ & \mathbf{z} = \boldsymbol{\alpha} \circ \mathbf{s} + \boldsymbol{\delta} + \boldsymbol{\lambda} + B\mathbf{p} \\ & \mathbf{s} \in \mathbb{R}^N_+, \quad \mathbf{p} \in \mathbb{R}^N_+, \quad \mathbf{z} \in \Omega, \end{aligned}$$

where

$$\Omega := \big\{\mathbf{z} \in \mathbb{R}^N_+ \mid \mathtt{diag}(\mathbf{z}) - B \in \mathbb{M}^{N\times N}_+\big\}. \tag{9}$$

We can show that the set Ω in (9) is convex; see [16]. Moreover, it follows from Lemma 2 that for any $1 \leq i,j \leq N$, the element $\big[\big(\mathtt{diag}(\mathbf{z}) - B\big)^{-1}\big]_{i,j}$ is convex and (element-wise) decreasing in $\mathbf{z} \in \Omega$. For these reasons, we obtain the following convex relaxation of (P2):

$$\begin{aligned} (\mathrm{P_R}) \qquad \min_{\mathbf{s},\mathbf{p},\mathbf{z}} \quad & f(\mathbf{s},\mathbf{p}) && (10)\\ \text{s.t.} \quad & \mathbf{p} \geq (\mathtt{diag}(\mathbf{z}) - B)^{-1}\boldsymbol{\lambda} && (11)\\ & \mathbf{z} = \boldsymbol{\alpha} \circ \mathbf{s} + \boldsymbol{\delta} + \boldsymbol{\lambda} + B\mathbf{p} && (12)\\ & \mathbf{s} \in \mathbb{R}^N_+, \quad \mathbf{p} \leq \mathbf{1}, \quad \mathbf{z} \in \Omega. \end{aligned}$$

This convex relaxation can be solved by numerical convex solvers to provide a lower bound on the optimal value of (P).

Theorem 2. *Let $\mathbf{x}^*_{\rm R} := (\mathbf{s}^*_{\rm R}, \mathbf{p}^*_{\rm R}, \mathbf{z}^*_{\rm R})$ denote an optimal point of $(\mathrm{P_R})$ and f^* the optimal value of (P). Then, we have*

$$f(\mathbf{s}^*_{\rm R}, \mathbf{p}^*_{\rm R}) \leq f^* \leq f(\tilde{\mathbf{s}}(\mathbf{x}^*_{\rm R}), \tilde{\mathbf{p}}(\mathbf{x}^*_{\rm R})),$$

where $(\tilde{\mathbf{s}}(\mathbf{x}^*_{\rm R}), \tilde{\mathbf{p}}(\mathbf{x}^*_{\rm R}))$ is a feasible solution for problem (P) given by $\tilde{\mathbf{p}}(\mathbf{x}^*_{\rm R}) = (\mathtt{diag}(\mathbf{z}^*_{\rm R}) - B)^{-1}\boldsymbol{\lambda}$ and $\tilde{\mathbf{s}}(\mathbf{x}^*_{\rm R}) = \mathbf{s}^*_{\rm R} + \mathtt{diag}(\boldsymbol{\alpha}^{-1})B(\mathbf{p}^*_{\rm R} - \tilde{\mathbf{p}}(\mathbf{x}^*_{\rm R}))$.

Clearly, $\mathbf{x}_{\mathrm{R}}^*$ solves (P) if the inequality constraints in (11) are all active at $\mathbf{x}_{\mathrm{R}}^*$, which means $f(\mathbf{s}_{\mathrm{R}}^*, \mathbf{p}_{\mathrm{R}}^*) = f(\tilde{\mathbf{s}}(\mathbf{x}_{\mathrm{R}}^*), \tilde{\mathbf{p}}(\mathbf{x}_{\mathrm{R}}^*))$. Based on this, we can provide a following sufficient condition for the convex relaxation $(\mathrm{P_R})$ to be exact.

Corollary 1. *Let* $(\tilde{\mathbf{s}}, \tilde{\mathbf{p}}) = (\tilde{\mathbf{s}}(\mathbf{x}_{\mathrm{R}}^*), \tilde{\mathbf{p}}(\mathbf{x}_{\mathrm{R}}^*))$ *be given as in Theorem 2. The above convex relaxation* $(\mathrm{P_R})$ *is exact if*

$$\nabla w(\tilde{\mathbf{s}})^{\mathsf{T}} \mathtt{diag}(\boldsymbol{\alpha}^{-1}) B \leq \mathbf{c}^{\mathsf{T}}. \tag{13}$$

Roughly speaking, the condition in (13) means that when the infection costs $\mathbf{c}$ are high, the convex relaxation $(\mathrm{P_R})$ is exact and we can find globally optimal security investments, i.e., a solution to (P), by solving the convex relaxation instead.

Remark 1. Although $(\mathrm{P_R})$ is a convex problem, there are a few obstacles to solving it numerically. First, the Jacobian of constraint functions in (11), which involves the derivative of inverse matrix $(\mathtt{diag}(\mathbf{z}) - B)^{-1}$, tends to be dense even when B is sparse. As a result, off-the-shelf convex solvers may not be suitable for large systems. Second, although the constraint set Ω defined in (9) is convex, it does not have a closed-form expression and is not numerically easy to handle, especially for large networks. This is because Ω is not closed and the relaxed problem becomes invalid outside Ω. Thus, numerical algorithms should stay inside Ω and, for this reason, the nonsingularity of the M-matrix, $\mathtt{diag}(\mathbf{z}) - B$, should be ensured at every step. In general, it takes $O(N^3)$ to check if the matrix satisfies this condition [20]. The following approach can, however, help reduce the computational burden.

(s1) Starting at some $\mathbf{z}_0 \in \Omega$, solve $(\mathrm{P_R})$ only with the constraint $\mathbf{z} \in \mathbb{R}_+^N$. Then, check if the obtained solution $\mathbf{x}_{\mathrm{R}}^*$ satisfies $\mathbf{z}_{\mathrm{R}}^* \in \Omega$, If so, $\mathbf{x}_{\mathrm{R}}^*$ solves $(\mathrm{P_R})$. Otherwise, go to step (s2).

(s2) Choose a simpler $\tilde{\Omega} \subset \Omega$ and solve $(\mathrm{P_R})$ subject to a new constraint $\mathbf{z} \in \tilde{\Omega}$. At the obtained solution $\tilde{\mathbf{x}}_{\mathrm{R}} = (\tilde{\mathbf{s}}_{\mathrm{R}}, \tilde{\mathbf{p}}_{\mathrm{R}}, \tilde{\mathbf{z}}_{\mathrm{R}})$, if $\tilde{\mathbf{z}}_{\mathrm{R}}$ lies in the interior of $\tilde{\Omega}$, then $\tilde{\mathbf{x}}$ is optimal for $(\mathrm{P_R})$; otherwise we construct a new $\tilde{\Omega}$ so that $\tilde{\mathbf{z}}_{\mathrm{R}}$ belongs to the interior of new $\tilde{\Omega}$ and repeat. In the next subsection, we propose an efficient way of choosing the subset $\tilde{\Omega}$ that is more suitable for numerical algorithms.

B. Construction of Convex Subsets of Ω

A key observation to constructing a sequence of suitable subsets of Ω is that, in view of Lemmas 1 and 2, Ω can be expressed as $\Omega = \bigcup_{\underline{\mathbf{z}} \in \partial\Omega} \{\mathbf{z} \in \mathbb{R}_+^N \mid \mathbf{z} \gtrsim \underline{\mathbf{z}}\}$, where $\partial\Omega$ denotes the boundary of Ω. Thus, for every $\underline{\mathbf{z}} \in \partial\Omega$, $\tilde{\Omega}(\underline{\mathbf{z}}) := \{\mathbf{z} \in \mathbb{R}_+^N \mid \mathbf{z} \gtrsim \underline{\mathbf{z}}\} \subset \Omega$. Our goal is to find some $\underline{\mathbf{z}} \in \partial\Omega$ such that at an optimal point $\tilde{\mathbf{x}}_{\mathrm{R}}$ that solves the relaxed problem with Ω replaced by $\tilde{\Omega}(\underline{\mathbf{z}})$, $\tilde{\mathbf{z}}_{\mathrm{R}}$ is an interior point of $\tilde{\Omega}(\underline{\mathbf{z}})$, i.e., $\tilde{\mathbf{x}}_{\mathrm{R}}$ solves $(\mathrm{P_R})$. We observe empirically that a static selection of the subset $\tilde{\Omega}$ does not always lead to a good solution and an iterative algorithm described below yields better performance.

Let $\mathbf{h} > \mathbf{0}$ be a normal vector of the plane tangent to the closure of Ω, denoted by $\bar{\Omega}$, at some $\underline{\mathbf{z}} \in \partial\Omega$ such that

$$\begin{aligned} \underline{\mathbf{z}} = \quad & \arg\min_{\mathbf{z} \in \mathbb{R}_+^N} \quad \{\mathbf{h}^{\mathsf{T}}\mathbf{z} \mid \mathbf{z} \in \bar{\Omega}\} \\ = \quad & \arg\min_{\mathbf{z} \in \mathbb{R}_+^N} \quad \{\mathbf{h}^{\mathsf{T}}\mathbf{z} \mid \underline{\sigma}\big(\mathtt{diag}(\mathbf{z}) - B\big) = 0\}, \end{aligned} \tag{14}$$

where the second equality follows from the fact that we are minimizing a linear function over a closed convex set. The minimization in (14) amounts to finding the smallest diagonal perturbation $\mathbf{z}$ (in 1-norm weighted by $\mathbf{h}$) so that B becomes (negative) stable. We show in [16] that this is in fact a *matrix balancing* problem, for which efficient algorithms exist (see [3], [18] for nearly-linear time centralized algorithms and [14], [15] for distributed algorithms with geometric convergence).

Our proposed algorithm is provided in Algorithm 1. Initially, we choose some $\bar{h} > 1$ and $\mathbf{h} = \boldsymbol{\alpha}^{-1} \circ \nabla w(\mathbf{s}_0)$, where $\mathbf{s}_0$ is the initial choice for security investments. This heuristic is based on the relaxed problem by weighting only the investment cost $w(\mathbf{s})$ without considering $\mathbf{p}$. Subsequent iterations are based on dominant eigenvalue with time-varying weights determined by $\mathbf{h}^+$ that reflects active constraints of $\tilde{\mathbf{z}}_{\mathrm{R}}$ (of the current solution). Moreover, since $\tilde{\mathbf{z}}_{\mathrm{R}} \in \Omega$, we have $\underline{\sigma}\big(\mathtt{diag}(\tilde{\mathbf{z}}_{\mathrm{R}}) - B\big) > 0$. Thus, we can construct a new subset $\tilde{\Omega}(\underline{\mathbf{z}})$ by translating the set $\{\mathbf{z} \geq \tilde{\mathbf{z}}_{\mathrm{R}}\}$ towards the boundary $\partial\Omega$ along the direction of $\mathbf{h}^+$, so that $\tilde{\mathbf{z}}_{\mathrm{R}}$ lies in the interior of new $\tilde{\Omega}(\underline{\mathbf{z}})$. In our numerical studies, we use $\bar{h} = 10$.

Algorithm 1: Algorithm for Convex Relaxation $(\mathrm{P_R})$

init: $t = 0$, $\bar{h} > 1$, $\underline{\mathbf{z}}^{(0)}$ from (14)
while *stopping cond. not met* **do**
  $(\tilde{\mathbf{s}}_{\mathrm{R}}^{(t+1)}, \tilde{\mathbf{p}}_{\mathrm{R}}^{(t+1)}, \tilde{\mathbf{z}}_{\mathrm{R}}^{(t+1)}) \leftarrow$ solve $(\mathrm{P_R})$ with $\mathbf{z} \in \tilde{\Omega}(\underline{\mathbf{z}}^{(t)})$
  $\mathcal{I}_{ac} \leftarrow \{i \in \mathcal{A} \mid [\tilde{\mathbf{z}}_{\mathrm{R}}^{(t+1)}]_i = [\underline{\mathbf{z}}^{(t)}]_i\}$
  if $\mathcal{I}_{ac} = \varnothing$ **then**
    break
  $\mathbf{h}^+ \leftarrow (h_i^+ = 1, i \notin \mathcal{I}_{ac}; h_i^+ = \bar{h}, i \in \mathcal{I}_{ac})$
  $d \leftarrow \underline{\sigma}\big(\mathtt{diag}(\mathbf{h}^+)^{-1}(\mathtt{diag}(\tilde{\mathbf{z}}_{\mathrm{R}}^{(t+1)}) - B)\big)$
  $\underline{\mathbf{z}}^{(t+1)} \leftarrow \tilde{\mathbf{z}}_{\mathrm{R}}^{(t+1)} - d\mathbf{h}^+$
  $t \leftarrow t + 1$

V. Upper Bound via a Reduced Gradient Method

Although the convex relaxation $(\mathrm{P_R})$ may be exact under certain conditions, this is not true in general, in which case it only provides a lower bound on the optimal value of (P). In addition, it may not scale well due to the constraint in (11); see also Remark 1 above and numerical results in Section VI. For these reasons, we also propose an efficient algorithm for finding a local minimizer of the non-convex problem (P). This provides an upper bound on the optimal value, which, together with the optimal value of the convex relaxation when available, can be used to offer a bound on the optimality gap.

Among different non-convex optimization approaches, we choose the reduced gradient method [6], [11] because it is well suited to the problem (P) and, more importantly, is scalable. Moreover, local convergence of the proposed algorithm can be established with the help of known convergence results of the generalized reduced gradient method (e.g., [6], [11]).

A. Main Algorithm

First, together with Theorem 1, the implicit function theorem tells us that the condition $\mathbf{g}(\mathbf{s}, \mathbf{p}) = \mathbf{0}$ in (4) defines a continuous mapping $\mathbf{p}^* : \mathbf{s} \in \mathbb{R}_+^N \mapsto \mathbf{p}^*(\mathbf{s}) \in (0,1)^N$ such

that $\mathbf{g}(\mathbf{s}, \mathbf{p}^*(\mathbf{s})) = 0$. Thus, (P) can be transformed into a reduced problem only with optimization variables $\mathbf{s}$:

$$\min_{\mathbf{s}\in\mathbb{R}^N_+} \quad F(\mathbf{s}) := w(\mathbf{s}) + \mathbf{c}^\mathsf{T}\mathbf{p}^*(\mathbf{s}). \tag{15}$$

Suppose that $(\mathbf{s}^\star, \mathbf{p}^\star)$ is a feasible point of (P). Then, the gradient of F at $\mathbf{s}^\star$ is equal to $\nabla F(\mathbf{s}^\star) = \nabla w(\mathbf{s}^\star) + J(\mathbf{s}^\star)^\mathsf{T}\mathbf{c}$, where $J(\mathbf{s}^\star) = \left[\partial p_i^*(\mathbf{s}^\star)/\partial s_j\right]$. This matrix can be computed by totally differentiating $\mathbf{g}(\mathbf{s}, \mathbf{p}^*(\mathbf{s})) = \mathbf{0}$ at $\mathbf{s}^\star$: the calculation of which yields

$$M(\mathbf{s}^\star)J(\mathbf{s}^\star) = -\mathtt{diag}(\boldsymbol{\alpha}\circ\mathbf{p}^\star) \tag{16}$$

with $M(\mathbf{s}^\star) = \mathtt{diag}(\boldsymbol{\alpha}\circ\mathbf{s}^\star + \boldsymbol{\delta} + \boldsymbol{\lambda} + B\mathbf{p}^\star) - \mathtt{diag}(\mathbf{1}-\mathbf{p}^\star)B$. The following lemma shows that $M(\mathbf{s}^\star)$ is nonsingular.

Lemma 3. *The matrix $M(\mathbf{s}^\star)$ is a nonsingular M-matrix.*

As $M(\mathbf{s}^\star)$ is nonsingular, $J(\mathbf{s}^\star) = -M(\mathbf{s}^\star)^{-1}\mathtt{diag}(\boldsymbol{\alpha}\circ\mathbf{p}^\star)$ from (16) and the gradient of F is given by

$$\nabla F(\mathbf{s}^\star) = \nabla w(\mathbf{s}^\star) - \boldsymbol{\alpha}\circ\mathbf{p}^\star\circ\left(M(\mathbf{s}^\star)^{-\mathsf{T}}\mathbf{c}\right).$$

Hence, we can now apply the gradient descent algorithm on the reduced problem with step sizes $\{\gamma_t\}_{t\geq 0}$, for example, using the Armijo backtracking line search scheme.

Note that, after each update of $\mathbf{s}$ during a search, we need to compute the corresponding $\mathbf{p}$ so that $(\mathbf{s}, \mathbf{p})$ is feasible for (P). This can be done by using the fixed point iteration in (2). Our proposed algorithm is provided in Algorithm 2.

Algorithm 2: Reduced Gradient Method

1 **init**: $t = 0$, feasible $(\mathbf{s}^{(0)}, \mathbf{p}^{(0)})$
2 **while** *stopping cond. not met* **do**
3 $\quad M^{(t)} = \mathtt{diag}(\boldsymbol{\alpha}\circ\mathbf{s}^{(t)} + \boldsymbol{\delta} + \boldsymbol{\lambda} + B\mathbf{p}^{(t)}) - \mathtt{diag}(\mathbf{1}-\mathbf{p}^{(t)})B$
4 $\quad \mathbf{u} = (M^{(t)})^{-\mathsf{T}}\mathbf{c}$
5 $\quad \gamma_t \leftarrow$ LINE_SEARCH
6 $\quad \mathbf{s}^{(t+1)} = \left[\mathbf{s}^{(t)} - \gamma_t\left(\nabla w(\mathbf{s}^{(t)}) - \boldsymbol{\alpha}\circ\mathbf{p}^{(t)}\circ\mathbf{u}\right)\right]_+$
7 $\quad \mathbf{p}^{(t+1)} = \mathbf{p}^*(\mathbf{s}^{(t+1)})$ using (2)
8 $\quad t = t + 1$

There are several possible stopping conditions one can employ, e.g., (i) a maximum number of iterations is reached $t = T_{\max}$, or (ii) the change in solution is small: $\frac{\|\mathbf{s}^{(t+1)}-\mathbf{s}^{(t)}\|_2}{\|\mathbf{s}^{(t)}\|_2} \leq \epsilon$.

B. Computational Complexity and Issues

For large systems, a naive evaluation of ∇F, which requires the inverse matrix $M(\mathbf{s}^\star)^{-\mathsf{T}}$, becomes computationally expensive, if not infeasible. Thus, we develop an efficient subroutine for computing ∇F. This is possible because our algorithm only requires $\mathbf{u}$ (in line 4 of Algorithm 2), not $M(\mathbf{s}^\star)^{-\mathsf{T}}$.

For fixed $t \in \{0, 1, \ldots\}$, the vector $\mathbf{u}$ is a solution to a set of linear equations $M^\mathsf{T}\mathbf{u} = \mathbf{c}$, where the matrix M^T tends to be sparse for most real graphs $\mathcal{G}$. Thus, there are many efficient algorithms for solving them. In this paper, we employ the power method: let $M^\mathsf{T} = D - E$, where D and E denote the diagonal part and off-diagonal parts of M^T, respectively. Then, the linear equations are equivalent to $\mathbf{c} = D\mathbf{u} - E\mathbf{u}$. Since D is invertible, the following fixed point relation holds:

$$\mathbf{u} = D^{-1}E\mathbf{u} + D^{-1}\mathbf{c} =: G(\mathbf{u}). \tag{17}$$

As $M \in \mathbb{M}^{N\times N}_+$ (Lemma 3), Lemma1-(d) tells us that $M^\mathsf{T} = D - E$ is a convergent splitting and the mapping G in (17) is a contraction mapping with coefficient $\rho(D^{-1}E) < 1$. Hence, the iteration $\mathbf{u}_{k+1} = G(\mathbf{u}_k)$ converges to the solution $\mathbf{u}$ exponentially fast. Moreover, this iteration is highly scalable; first, $E^\mathsf{T} = \mathtt{diag}(\mathbf{1}-\mathbf{p})B$ is sparse, requiring only $O(|\mathcal{E}|)$ memory space. Second, the computation also takes $O(|\mathcal{E}|)$ operations. Thus, after k_u updates with total running time of $O(k_u|\mathcal{E}|)$, we obtain an estimate of $\mathbf{u}$ with convergence error proportional to $\left(\rho(D^{-1}E)\right)^{k_u}$.

VI. Numerical Results

In this section, we provide some numerical results that demonstrate the performance of the proposed algorithms. Our numerical studies are carried out in MATLAB on a laptop with 8GB RAM and a single Intel Core i5 processor with clock speed of 2.4GHz.[2] We consider 5 different strongly connected scale-free networks with $(N, |\mathcal{E}|) = (100, 474)$, (200, 1014), (500, 2738), (1000, 5750), and (1500, 8850). The power law parameter for node degrees is set to $\xi = 1.5$, and the minimum and maximum node degrees are equal to 2 and $\lceil 3\log N\rceil$, respectively. For all considered networks, we pick $\alpha_i = 1$ and $\delta_i = 0.1$. The direct attack rates λ_i, $i \in \mathcal{A}$, and infection rates $\beta_{i,j}$, $\langle i, j\rangle \in \mathcal{E}$, are modeled using i.i.d. Uniform(0,1) random variables. We choose

$$w(\mathbf{s}) = \mathbf{1}^\mathsf{T}\mathbf{s} \quad \text{and} \quad \mathbf{c} = \nu B^\mathsf{T}\mathbf{1} + 2\mathbf{c}_{\rm rand},\ \nu \in \{0, 0.5, 1\},$$

where the elements of $\mathbf{c}_{\rm rand}$ are given by i.i.d. Uniform(0,1) random variables. We consider $\mathbf{c}$ above, in order to reflect an observation that nodes which support more neighbors should, on the average, have larger economic costs modeled by c_i^e (Section III-A).

In line 3 of Algorithm 1 for solving $(\mathrm{P_R})$ with Ω replaced by $\tilde{\Omega}(\underline{\mathbf{z}}^{(t)})$, we use the interior-point optimizer from package [5] with relative convergence tolerance set to 10^{-6} and Hessian matrices approximated by a quasi-Newton algorithm. The initial point $\tilde{\mathbf{x}}_{\rm R}^{(0)} = (\tilde{\mathbf{s}}_{\rm R}^{(0)}, \tilde{\mathbf{p}}_{\rm R}^{(0)}, \tilde{\mathbf{z}}_{\rm R}^{(0)})$ is chosen to be $\tilde{\mathbf{s}}_{\rm R}^{(0)} = \mathbf{0}$, $\tilde{\mathbf{p}}_{\rm R}^{(0)} = \mathbf{p}^*(\mathbf{0})$ using the iteration in (2), and $\tilde{\mathbf{z}}_{\rm R}^{(0)} = \boldsymbol{\lambda} + \boldsymbol{\delta} + B\tilde{\mathbf{p}}_{\rm R}^{(0)}$ according to (7). When computing $\tilde{\mathbf{p}}_{\rm R}^{(0)}$, the iteration in (2) is run until $\frac{||\mathbf{p}_{k+1}-\mathbf{p}_k||_2}{||\mathbf{p}_k||_2} \leq 10^{-8}$. In addition, we approximate the set of active constraints of $\tilde{\mathbf{z}}_{\rm R}$ (line 4 of Algorithm 1) using $\mathcal{I}_{ac} = \{i \in \mathcal{A} \mid [\tilde{\mathbf{z}}_{\rm R}]_i - [\underline{\mathbf{z}}]_i \leq 10^{-3}\}$, and select $\bar{h} = 10$.

For Algorithm 2, we select $(\mathbf{s}^{(0)}, \mathbf{p}^{(0)}) = (\mathbf{0}, \mathbf{p}^*(\mathbf{0}))$ as a feasible initial point[3] and stop the algorithm whenever $\frac{\|\mathbf{s}^{(t+1)}-\mathbf{s}^{(t)}\|_2}{\|\mathbf{s}^{(t)}\|_2} \leq 10^{-6}$ or $\frac{|F(\mathbf{s}^{(t+1)})-F(\mathbf{s}^{(t)})|}{F(\mathbf{s}^{(t)})} \leq 10^{-10}$. We compute $\mathbf{u}$ in line 4 using the fixed point iteration in (17) and $\mathbf{p}^*(\mathbf{s}^{(t+1)})$ in line 7 using the iteration in (2), with stopping conditions $\frac{\|\mathbf{u}_{k+1}-\mathbf{u}_k\|_2}{\|\mathbf{u}_k\|_2} \leq 10^{-8}$ and $\frac{\|\mathbf{p}_{k+1}-\mathbf{p}_k\|_2}{\|\mathbf{p}_k\|_2} \leq 10^{-8}$. We

[2]Mention of commercial products does not imply NIST's endorsement.

[3]If $(\tilde{\mathbf{s}}, \tilde{\mathbf{p}}) = (\tilde{\mathbf{s}}(\mathbf{x}_{\rm R}^*), \tilde{\mathbf{p}}(\mathbf{x}_{\rm R}^*))$ given in Theorem 2 is available, it can be used an initial point. Here, we choose $(\mathbf{0}, \mathbf{p}^*(\mathbf{0}))$ for numerical comparisons.

TABLE I
NUMERICAL RESULTS FOR SCALE-FREE NETWORKS.

$\nu = 0$	Algorithm 1			Algorithm 2				
N	$\lvert\tilde{f} - f_{\rm R}^*\rvert/f_{\rm R}^*$	iter_seq	t_s (s)	$\lvert\overline{f} - f_{\rm R}^*\rvert/f_{\rm R}^*$	iter	$\bar{k}_u$	$\bar{k}_p$	t_s (s)
100	1.34×10^{-1}	20/18	0.26	1.35×10^{-2}	56	9	7	0.02
200	1.28×10^{-1}	19/17/14	1.37	1.45×10^{-2}	34	9	7	0.02
500	1.29×10^{-1}	25/25/30	21.2	1.42×10^{-2}	48	7	6	0.05
1000	1.30×10^{-1}	29/24/23	150	1.30×10^{-2}	36	13	11	0.12
1500	1.57×10^{-1}	24/22/19	438	1.40×10^{-2}	56	9	7	0.34

$\nu = 0.5$	Algorithm 1			Algorithm 2				
N	$\lvert\tilde{f} - f_{\rm R}^*\rvert/f_{\rm R}^*$	iter_seq	t_s (s)	$\lvert\overline{f} - f_{\rm R}^*\rvert/f_{\rm R}^*$	iter	$\bar{k}_u$	$\bar{k}_p$	t_s (s)
100	1.66×10^{-2}	21/17	0.24	3.11×10^{-3}	75	9	7	0.02
200	1.28×10^{-2}	20/20/16	1.52	2.64×10^{-3}	71	9	7	0.03
500	1.25×10^{-2}	24/22	13.1	2.02×10^{-3}	75	11	8	0.09
1000	1.20×10^{-2}	27/21/15	126	1.96×10^{-3}	71	12	9	0.25
1500	1.38×10^{-2}	38/23/23	556	2.05×10^{-3}	77	13	11	0.47

$\nu = 1$	Algorithm 1			Algorithm 2				
N	$\lvert\tilde{f} - f_{\rm R}^*\rvert/f_{\rm R}^*$	iter_seq	t_s (s)	$\lvert\overline{f} - f_{\rm R}^*\rvert/f_{\rm R}^*$	iter	$\bar{k}_u$	$\bar{k}_p$	t_s (s)
100	3.54×10^{-9}	22	0.14	2.11×10^{-8}	169	18	12	0.05
200	2.71×10^{-9}	29	0.78	2.34×10^{-8}	177	15	10	0.10
500	1.83×10^{-9}	27/12	10.7	2.37×10^{-8}	209	16	11	0.33
1000	2.84×10^{-9}	31	66	1.58×10^{-8}	247	15	11	0.70
1500	6.01×10^{-9}	40	259	1.30×10^{-8}	220	16	12	1.30

employ the backtracking line search with $\gamma_0 = 0.5$, a shrinking factor of 0.85, and the Armijo condition parameter set to 10^{-4}.

Our numerical results are summarized in Table I. For Algorithm 1, we report the relative gap $|\tilde{f} - f_{\rm R}^*|/f_{\rm R}^*$, where $f_{\rm R}^*$ denotes the optimal value of $(\mathrm{P_R})$ and $\tilde{f} := f(\tilde{\mathbf{s}}(\mathbf{x}_{\rm R}^*), \tilde{\mathbf{p}}(\mathbf{x}_{\rm R}^*))$. From our numerical studies, we have the following observations: (i) As ν increases and infection costs become larger, as expected from Corollary 1, the gap diminishes and becomes negligible when $\nu = 1$. (ii) We also report in the column iter_seq the sequence of inner interior-point iterations. The number of outer iterations is relatively small (at most 3 in all cases as shown in the table). However, as expected, the runtime (in seconds) denoted by t_s does not scale well with the network size. For example, for a large network with $(N, |\mathcal{E}|) = (2000, 12076)$, the interior-point method failed to converge within an hour.

For Algorithm 2, we also report a similar relative gap $|\overline{f} - f_{\rm R}^*|/f_{\rm R}^*$, where $\overline{f} := f(\mathbf{s}^\star, \mathbf{p}^\star)$ and $(\mathbf{s}^\star, \mathbf{p}^\star)$ is the solution found by Algorithm 2. It is worth noting that the upper bound $\overline{f}$ is very close to the lower bound $f_{\rm R}^*$, even when the relaxation may not be exact (for $\nu = 0, 0.5$). In addition, it achieves optimal solutions when the relaxation is exact. This suggests that the algorithm can practically find global solutions to the original problem. We also report the maximum number of fixed point iterations needed for evaluating $\mathbf{u}$ in line 4 and $\mathbf{p}^*$ in line 7, denoted by $\bar{k}_u$ and $\bar{k}_p$, respectively. In our studies, $\bar{k}_u$ and $\bar{k}_p$ are all relatively small as expected from our earlier discussions (Theorem 1 and Section V-B). Finally, Algorithm 2 is highly scalable: in spite of a larger number of required iterations compared to Algorithm 1, the total runtime t_s is much smaller and is a fraction of that of Algorithm 1.

REFERENCES

[1] Yuliy Baryshnikov. IT security investment and Gordon-Loev's $1/e$ rule. In *Proc. of Annual Workshop on the Economics of Information Security*, 2012.

[2] Christian Borgs, Jennifer Chayes, Ayalvadi Ganesh, and Amin Saberi. How to distributed antidote to control epidemics. *Random Structures & Algorithms*, 37(2):204–222, September 2010.

[3] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[4] Reuven Cohen, Shlomo Havlin, and Daniel ben Avraham. Efficient immunization strategies for computer networks and populations. *Physical Review Letters*, 91(247901), December 2003.

[5] Jonathan Currie and David I Wilson. OPTI: lowering the barrier between open source optimizers and the industrial MATLAB user. *Foundations of Computer-aided Process Operations*, 24:32, 2012.

[6] Daniel Gabay and David G Luenberger. Efficiently converging minimization methods based on the reduced gradient. *SIAM Journal on Control and Optimization*, 14(1):42–61, 1976.

[7] Eric Gourdin, Jasmina Omic, and Piet Van Mieghem. Optimization of network protection against virus spread. In *Proc. of International Workshop on the Design of Reliable Communication Networks*, pages 86–93, 2011.

[8] Ashish R. Hota and Shreyas Sundaram. Interdependent security games on networks under behavioral probability weighting. *IEEE Trans. on Control of Network Systems*, 5(1):262–273, March 2018.

[9] Mohammad M. Khalili, Xueru Zhang, and Mingyan Liu. Incentivizing effort in interdependent security games using resource pooling. In *Proc. of Workshop on the Economics of Networks, Systems and Computation*, pages 1–6, 2019.

[10] Richard J. La. Interdependent security with strategic agents and global cascades. *IEEE/ACM Trans. on Networking*, 24(3):1378–1391, June 2016.

[11] Leon S Lasdon, Richard L Fox, and Margery W Ratner. Nonlinear optimization using the generalized reduced gradient method. *Revue française d'automatique, informatique, recherche opérationnelle. Recherche opérationnelle*, 8(V3):73–103, 1974.

[12] Marc Lelarge and Jean Bolot. A local mean field analysis of security investments in networks. In *Proc. of International Workshop on Economics of Networks Systems*, pages 25–30, 2008.

[13] Van Sy Mai and Eyad H Abed. Optimizing leader influence in networks through selection of direct followers. *IEEE Trans. on Automatic Control*, 64(3):1280–1287, 2018.

[14] Van Sy Mai and Abdella Battou. Asynchronous distributed matrix balancing and application to suppressing epidemic. In *Proc. of 2019 American Control Conference*, pages 2777–2782. IEEE, 2019.

[15] Van Sy Mai, Abdella Battou, and Kevin Mills. Distributed algorithm for suppressing epidemic spread in networks. *IEEE Control Systems Letters*, 2(3):555–560, 2018.

[16] Van Sy Mai, Richard La, and Abdella Battou. Optimal cybersecurity investments for SIS model. *arXiv preprint arXiv:2005.07257*, 2020. https://arxiv.org/abs/2005.07257.

[17] Cameron Nowzari, Victor M Preciado, and George J. Pappas. Optimal resource allocation for control of networked epidemic models. *IEEE Control Systems Magazine*, 4(2):159–169, June 2017.

[18] Rafail Ostrovsky, Yuval Rabani, and Arman Yousefi. Matrix balancing in L_p norms: bounding the convergence rate of Osborne's iteration. In *Proc. ACM-SIAM Symp. Discrete Algorithms*, pages 154–169, 2017.

[19] Stefania Ottaviano, Francesco De Pellegrini, Stefano Bonaccorsi, and Piet Van Mieghem. Optimal curing policy for epidemic spreading over a community network with heterogeneous population. *Journal of Complex Networks*, 6(6), October 2018.

[20] J Peña. A stable test to check if a matrix is a nonsingular M-matrix. *Mathematics of Computation*, 73(247):1385–1392, 2004.

[21] Robert J Plemmons. M-matrix characterizations. I–nonsingular M-matrices. *Linear Algebra and its Applications*, 18(2):175–188, 1977.

[22] Victor M. Preciado, Michael Zargham, Chinwendu Enyioha, Ali Jadbabaie, and George Pappas. Optimal vaccine allocation to control epidemic outbreaks in arbitrary networks. In *Proc. of IEEE Conference on Decision and Control*, pages 7486–7491. IEEE, 2013.

[23] Victor M. Preciado, Michael Zargham, Chinwendu Enyioha, Ali Jadbabaie, and George J. Pappas. Optimal curing policy for epidemic spreading over a community network with heterogeneous population. *IEEE Trans. on Control of Network Systems*, 1(1):99–108, March 2014.