

Resonance Reduction Against Adversarial Attacks in Dynamic Networks via Eigenspectrum Optimization

Alp Sahin¹, Nicolas Kozachuk², Rick S. Blum² and Subhrajit Bhattacharya¹

Abstract—Resonance is a well-known phenomenon that happens in systems with second order dynamics. In this paper we address the fundamental question of making a network robust to signal being periodically pumped into it at or near a resonant frequency by an adversarial agent with the aim of saturating the network with the signal. Towards this goal, we develop the notion of *network vulnerability*, which is measured by the expected resonance amplitude on the network under a stochastically modeled adversarial attack. Assuming a second order dynamics model based on the network graph Laplacian matrix and a known stochastic model for the adversarial attack, we propose two methods for minimizing the network vulnerability that leverage the principle of *eigenspectrum optimization*. We provide extensive numerical results analyzing the effects of both methods.

Index Terms—Network Graphs, Second Order Dynamics, Vulnerability Reduction

I. INTRODUCTION

In this paper we consider the phenomenon of runaway amplification of signal in a network due to *resonance*, which has implications on security of the network. This is possible if an adversarial agent pumps signal into one or more vertices of the network in a periodic manner at a frequency that matches or is very close to one of the *natural frequencies* of the network. This phenomenon is observed in networks with a second order signal dynamics.

While second order dynamics over networks has been studied in the past [1], [2], [3], [4], especially in context of power grids (since power transmission using alternating currents are described naturally using second-order dynamics), existing literature does not focus on controlling network parameters and topology for the purpose of mitigation of resonance.

We consider networks whose dynamics are governed by second order differential equations where the coefficients are functions of the graph Laplacian matrix. Assuming an adversarial signal source that obeys a known stochastic model, we develop two methods (Network Graph Optimization and Auxiliary Graph Optimization) for optimizing the network structure to reduce signal resonance under the following conditions respectively: (i) network structure can be altered by modifying the edge weights (representing the connection strength between two network nodes), (ii) edge weights of the network cannot be modified directly, but an auxiliary network can be attached to the system. The contributions of this paper are as follows:

- We provide a second-order dynamics model of the network behavior under external forcing (source of the adversarial signal), that is consistent with the network topology (Section III).
- We develop the notion of network vulnerability, measured by the expected resonance amplitude under stochastically modeled adversarial forcing (Section III).

- We propose two methods, namely Network Graph Optimization and Auxiliary Graph Optimization, both of which rely on the principle of eigenspectrum optimization (Sections IV and V).
- We analyze the performance of both methods through numerical experiments (Section VI).

II. RELATED WORK

The Laplacian dynamics on a graph, $\dot{\mathbf{x}} = -L\mathbf{x}$, as a linear signal transmission model is a model for transmission that represents *diffusion* across the network and occurs in applications frequently [5], [6]. In particular, if x_i is the signal value on i -th vertex, then this dynamics corresponds to its rate of change as a sum of the influx of the signals from its neighbors (scaled with the corresponding edge weights), minus the outflux to its neighbors.

While first-order signal dynamics is most well-studied in context of networks [5], [7], [8], higher-order dynamics has also been studied. A second-order dynamics over a network is relevant, for example, in context of distributed power grids, electrical circuits and consensus in such networks [3], [9], [10], where the dynamics of alternating electrical current and voltage are naturally second order. The motion dynamics of mobile agents (e.g., robots) is often governed by Newtonian dynamics, which gives rise to second-order dynamics over a network of such agents [11]. Second order dynamics can also be used to model transmission of information on social networks where the transmissibility of a signal depends both on its amount (how widespread it is) and its rate of change (how “viral” it is). The properties of second-order dynamics over networks has been well-studied in the literature (see [1], [2] for example), and model reduction in the context of such dynamics has been investigated [3], [4]. However, existing literature does not focus on active control of network parameters and topology for the purpose of prevention of resonance.

Optimization of the spectrum of the Laplacian matrix in order to affect the connectivity of a network has also been extensively studied [12], [13], [14]. However, most often, such optimization problems focus on the network connectivity in general, without explicitly addressing performance a second-order signal dynamics over the network.

In this paper we consider a general second-order dynamics over a network with external forcing. We particularly focus on developing methods for mitigating resonance attacks inflicted by an adversarial agent pumping oscillatory signal in a periodic manner at one or more vertices while trying to match a natural frequency of the network. To our knowledge, there has been no prior work on active detection and control of resonance in a general graphical network with a focus on increasing robustness of the network to adversarial attacks.

III. MOTIVATION & BACKGROUND

We consider a network (referred to as the *main network*) represented by a weighted undirected graph $G = (V, E, \mathbf{w})$ where V is the vertex set, $E \subseteq V \times_{\text{sym}} V$ is the edge set, and \mathbf{w} is a set of real weights on the edges. The vertices are indexed by natural numbers,

This work was supported by AFOSR award number FA9550-23-1-0046.

¹Department of Mechanical Engineering and Mechanics, Lehigh University, 19 Memorial Drive West, Bethlehem, PA 18015, U.S.A., [als421, sub216]@lehigh.edu.

²Department of Electrical and Computer Engineering, Lehigh University, 19 Memorial Drive West, Bethlehem, PA 18015, U.S.A., [ngk324, rb0f]@lehigh.edu.

$1, 2, \dots, n$ (where n is the number of vertices), and the set of neighbors of the k -th vertex is denoted $\mathcal{N}_k = \{j | (k, j) \in E\}$. The weight on an edge $(j, k) \in E$ is denoted by w_{jk} . We also assign a natural number indexing to the edges, $1, 2, \dots, m$ (where m is the number of edges), and with a little abuse of notation, w_l will refer to the weight on the l -th edge.

The signal on the k -th vertex is modeled as a complex number, $x_k \in \mathbb{C}$ (while in practice the signal may be real, in which case the real part of the signal and dynamics equations are of relevance, the equations and their general solutions are most compactly represented by a complex dynamics), which follows a second order linear dynamics coupled with the signals on the neighbors of the k -th vertex in G .

In it's simplest form, such a dynamics can be constructed as a natural extension of the first-order Laplacian dynamics, such that the second derivative of the signal on the k -th vertex is equal to the rate of *influx* of signal from the neighbors of the vertex minus the rate of *outflux* of signal to the neighbors, with the influx and outflux being proportional to the signal on the respective vertices. With the edge weights identified as the proportionality constants, this simple dynamics can be written as $\ddot{x}_k = \sum_{j \in \mathcal{N}_k} w_{jk} x_j - \sum_{j \in \mathcal{N}_k} w_{jk} x_k$. This dynamics can be compactly written as $\ddot{\mathbf{x}} + \mathbf{L}\mathbf{x} = \mathbf{0}$, where $\mathbf{x} \in \mathbb{C}^n$ is the *signal vector* (the k -th element of which is x_k) and $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is the weighted graph Laplacian matrix (\mathbf{A} is the *weighted adjacency matrix* and \mathbf{D} the *weighted degree matrix*). The Laplacian matrix satisfies the property that its (j, k) -th element is zero if there does not exist an edge connecting vertices k and j . This property of the Laplacian matrix ensures that the dynamics of signal at a vertex depends on the signals on the neighboring vertices only, and will be referred to as the property of being *consistent with the network topology*.

In this paper we consider a more general form of second-order linear dynamics for signals following second order differential equation [15]:

$$\ddot{\mathbf{x}} + \mathbf{\Gamma}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{f}e^{i\nu t} \quad (1)$$

where, \mathbf{K} and $\mathbf{\Gamma}$ are called the *stiffness* and *damping* matrices respectively that are consistent with the network topology (i.e., their (k, j) -th element is nonzero only if there exists an edge between the k -th and j -th vertices in the graph). The network is subject to an adversarial forcing specified by the *forcing vector* \mathbf{f} (with its k -th element, f_k , being the amplitude of adversarial signal forced on the k -th vertex) and *forcing frequency* ν .

The solution to (1), when there is no external forcing (i.e., $\mathbf{f} = \mathbf{0}$), exhibits oscillatory nature when the damping matrix is positive definite and the damping is small [15]. In line with the dynamics of a signal at a vertex being the signed sum of influx and outflux of signals weighed by edge weights, we choose the stiffness matrix to be $\mathbf{K} = \mathbf{L} + \epsilon \mathbf{I}$. The role of the $\epsilon \mathbf{I}$ term, for a small $\epsilon > 0$, is to ensure that \mathbf{K} is positive definite (all eigenvalue of \mathbf{K} are strictly greater than zero), which in turn prevents drift in the dynamics, since it is well-known that the weighted graph Laplacian, \mathbf{L} , has a non-trivial nullspace [16]. For notational convenience, we also define the matrix $\mathbf{\Omega}$ such that $\mathbf{\Omega}^2 = \mathbf{K} = \mathbf{L} + \epsilon \mathbf{I}$. We choose the damping matrix as $\mathbf{\Gamma} = 2\gamma\mathbf{\Omega}^2$ for some small real $\gamma > 0$, which corresponds to the fact that the damping over an edge is proportional to the edge weight (scaled by a factor of 2γ). This makes both \mathbf{K} and $\mathbf{\Gamma}$ consistent with the network topology. In the later sections, we will assume the damping multiplier γ to be small. Using these new notations, we can write the dynamics (1) as

$$\ddot{\mathbf{x}} + 2\gamma\mathbf{\Omega}^2\dot{\mathbf{x}} + \mathbf{\Omega}^2\mathbf{x} = \mathbf{f}e^{i\nu t} \quad (2)$$

The steady-state solution to equation (2) is given by [15]:

$$\mathbf{x}_s = (-\nu^2 \mathbf{I} + 2i\nu\gamma\mathbf{\Omega}^2 + \mathbf{\Omega}^2)^{-1} \mathbf{f} e^{i\nu t} \quad (3)$$

It is a well-known fact that if the forcing frequency ν matches one of the natural frequencies of the network (one of the eigenvalues

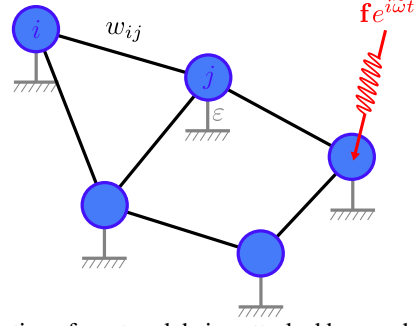


Fig. 1: Illustration of a network being attacked by an adversarial agent trying to cause resonance.

of $\mathbf{\Omega}$), that leads to resonance, where, with a small damping, the steady-state amplitude of the forced oscillations can get arbitrarily large. The objective of this paper is to minimize the expected steady-state amplitude under a probabilistic model for the distribution of the forcing frequency ν .

We assume that the adversarial agent tries to match its forcing frequency, ν , with one of the natural frequencies of the system (one of the eigenvalues of $\mathbf{\Omega}$), but, is subject to uncertainties, either due to an inability to precisely select the forcing frequency, or because of an imprecise knowledge of the natural frequencies of the system. In particular, we assume that ν is a stochastic variable with a probability density function dependent upon the natural frequencies of the system. We measure the *vulnerability of a network* against the attacks by the expected value of the squared 2-norm of the steady-state response, denoted as $\mathbb{E}_{\nu, \mathbf{f}}(\|\mathbf{x}_s\|_2^2)$.

The main objective of this work is to develop an approach that relies on the principle of eigenspectrum optimization to reduce the vulnerability of the network against adversarial attacks with a known stochastic model. We approach this problem in two different ways:

- (1) A direct optimization of the weights on the edges of the network that minimizes $\mathbb{E}_{\nu, \mathbf{f}}(\|\mathbf{x}_s\|_2^2)$. We refer to this approach as *Network Graph Optimization* (Section IV).
- (2) When it is not possible to alter the weights on the edges directly, we propose to attach an *auxiliary network* to the main network, and tune/optimize it such that this auxiliary network can effectively absorb and dissipate the excess energy from the resonance in the main network while minimizing the expected steady-state amplitude on the main network. We refer to this approach as *Auxiliary Graph Optimization* (Section V).

IV. NETWORK GRAPH OPTIMIZATION

Given an initial configuration of the main network specified via the graph G , the Network Graph Optimization (NGO), refers to the procedure of changing the main network graph's weights and/or topology in such a way that the vulnerability of the network is minimized against the adversarial agent's forcing behavior (forcing vector and frequency) obeying the stochastic model that will be explained in Sections IV-B and IV-A.

In this section, we derive an expression for the expected value of the squared 2-norm of the steady-state response (i.e. the vulnerability of the network), and formulate the eigenspectrum optimization problem to minimize the vulnerability of the network.

A. Stochastic Modeling of the Forcing Vector

We assume that the forcing vector \mathbf{f} is sampled from a uniform distribution over a $(n-1)$ -unit sphere.

Lemma 1. If $\mathbf{f} \in \mathbb{R}^n$ is sampled from an uniform distribution over a $(n-1)$ -unit sphere and M is a symmetric matrix, then

$$\mathbb{E}_{\mathbf{f}}(\|\mathbf{M}\mathbf{f}\|_2^2) = \frac{1}{n} \|\mathbf{M}\|_F^2$$

where $\|\cdot\|_F$ is the Frobenius norm.

The proof of this lemma is deferred to Appendix A for better readability. Using the result of this lemma and the identity $\|\mathbf{M}\|_F^2 = \text{Tr}(\mathbf{M}^H \mathbf{M}) = \sum_{i=1}^n |\lambda_i(\mathbf{M})|^2$ (where \mathbf{M}^H denotes the conjugate transpose and $\lambda_i(\mathbf{M})$ denotes the i -th eigenvalue of \mathbf{M}), the expected value of $\|\mathbf{x}_s\|_2^2$ (equation (3)) with respect to the random variable \mathbf{f} is thus computed as follows:

$$\begin{aligned} \mathbb{E}_{\mathbf{f}}(\|\mathbf{x}_s\|_2^2) &= \frac{1}{n} \|(-v^2 I + i2v\gamma\Omega^2 + \Omega^2)^{-1}\|_F^2 \\ &= \frac{1}{n} \sum_k \frac{1}{|-v^2 + i2v\gamma\omega_k^2 + \omega_k^2|^2} \\ &= \frac{1}{n} \sum_k \frac{1}{(\omega_k^2 - v^2)^2 + (2\gamma v\omega_k^2)^2} \end{aligned} \quad (4)$$

where ω_k are the eigenvalues of Ω .

As seen from equation (4), if v matches one of the eigenvalues of Ω , the corresponding term of the summation blows up when the damping, γ , is small, leading to *resonance*.

B. Stochastic Modeling of the Forcing Frequency

We assume that the adversarial agent has uncertain knowledge of the network (or equivalently precise knowledge of the network, but uncertainty/error in choosing a forcing frequency). This uncertainty/error manifests itself when the adversarial agent tries to pick a forcing frequency that matches with one of the natural frequencies of the network. We model this uncertainty by considering v to be a random variable with a probability density function, ρ , constructed as a uniformly weighted sum of multiple Cauchy distributions [17] each of which are centered at the natural frequencies, $\{\omega_j\}_{j=1,\dots,n}$, with a constant spread of h :

$$\rho(v) = \frac{1}{n} \sum_{j=1}^n \rho_{\omega_j}(v) = \frac{1}{n} \sum_{j=1}^n \frac{h/\pi}{(\omega_j - v)^2 + h^2} \quad (5)$$

The Cauchy distribution, as opposed to other probability distributions, allows the integral representing the expected value of $\|\mathbf{x}_s\|_2^2$ to converge. Figure 2 illustrates an example where three individual Cauchy distributions are summed up with uniform weights to obtain a composite probability distribution $\rho(v)$.

C. Derivation of the Optimization Objective

The expected value of $\|\mathbf{x}_s\|_2^2$ with respect to the random variables \mathbf{f} and v can then be calculated using equations (4) and (5) as follows:

$$\begin{aligned} \mathbb{E}_{\mathbf{f}, v}(\|\mathbf{x}_s\|_2^2) &= \int_{-\infty}^{\infty} \mathbb{E}_{\mathbf{f}}(\|\mathbf{x}_s\|_2^2) \rho(v) dv \\ &= \frac{h}{\pi n^2} \sum_{k,j} g(\omega_k, \omega_j) \end{aligned} \quad (6)$$

where

$$g(\omega_k, \omega_j) = \int_{-\infty}^{\infty} \frac{dv}{((\omega_k^2 - v^2)^2 + (2\gamma v\omega_k^2)^2)((\omega_j - v)^2 + h^2)} \quad (7)$$

If γ is non-zero, the poles of the integrand above lie away from the real line on the complex plane, and hence a closed-form expression for the integral $g(\omega_k, \omega_j)$ can be obtained using the Residue theorem [18] by

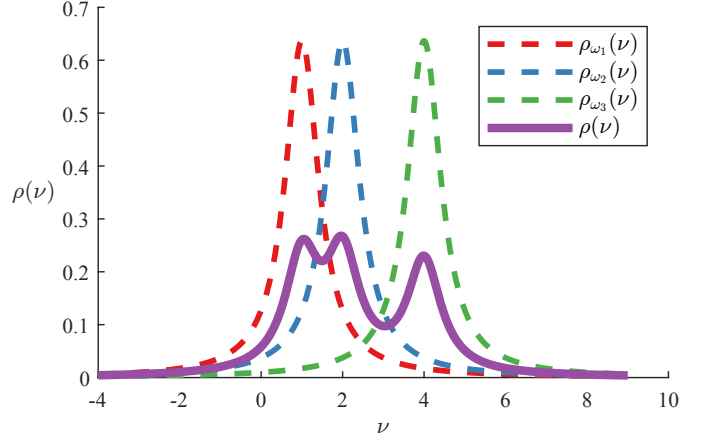


Fig. 2: Cauchy distributions centered at the natural frequencies $\omega_1 = 1$, $\omega_2 = 2$, and $\omega_3 = 4$ with a spread of $h = 0.5$. The probability density function $\rho(v) = \frac{1}{3} \sum_{i=1}^3 \rho_{\omega_i}(v)$ for the adversarial agent's choice of forcing frequency is obtained as the uniformly weighted sum of the Cauchy distributions each of which are centered at the natural frequencies of the network.

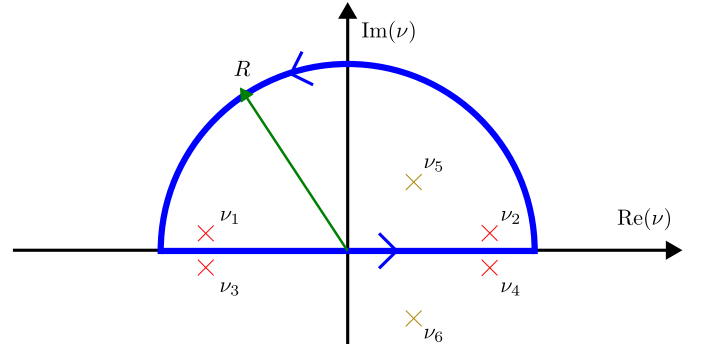


Fig. 3: Integration contour for equation (7). Poles v_1 to v_4 correspond to the forcing vector component $\mathbb{E}_{\mathbf{f}}(\|\mathbf{x}_s\|_2^2)$ and they collapse on to the real line as γ goes to zero. Poles v_5 and v_6 correspond to the forcing frequency component $\rho(v)$.

performing a contour integration over the real line and a semi-circular arc of radius $R \rightarrow \infty$ on the upper half of the complex plane (Figure 3).

Assuming $\gamma \ll h$, we compute the roots of the quartic polynomial in v in the denominator of the integrand in (7) using a symbolic algebra toolbox, and then apply the Residue theorem to obtain

$$g(\omega_k, \omega_j) = \frac{\pi}{2\gamma} \frac{h^2 + \omega_k^2 + \omega_j^2}{\omega_k^4 (h^4 + 2h^2(\omega_k^2 + \omega_j^2) + (\omega_k^2 - \omega_j^2)^2)} \quad (8)$$

From (6) this leads to the following expression for the expected value of the 2-norm of the steady-state amplitude:

$$\mathbb{E}_{\mathbf{f}, v}(\|\mathbf{x}_s\|_2^2) = \frac{h}{2\gamma n^2} \sum_{k,j} \frac{h^2 + \omega_k^2 + \omega_j^2}{\omega_k^4 (h^4 + 2h^2(\omega_k^2 + \omega_j^2) + (\omega_k^2 - \omega_j^2)^2)} \quad (9)$$

The objective is to minimize this expected value of the 2-norm of the steady-state amplitude, so as to mitigate the effects of resonance attacks on the network. We note that ω_k and ω_j are the eigenvalues of $\Omega = \sqrt{L + \varepsilon I}$, where the Laplacian matrix, $L = D - A$, depends on the weights on the edges of the graph. Thus $\mathbb{E}_{\mathbf{f}, v}(\|\mathbf{x}_s\|_2^2)$, as described in (9), is a function of the edge weights of the graph. We thus define the objective function, $J(\mathbf{w}) = \mathbb{E}_{\mathbf{f}, v}(\|\mathbf{x}_s\|_2^2)$ to be a function of the edge

weight vector, $\mathbf{w} \in \mathbb{R}^m$ (where m is the number of edges in the graph), given by (9). It can be checked that J is in general a non-convex function.

D. Eigenspectrum Optimization of the Main Network Graph

We define the *eigenspectrum optimization problem of the main network graph* as the problem of minimizing the expected steady-state amplitude of signal on the network under the described stochastic forcing:

$$\begin{aligned} & \underset{\mathbf{w}}{\text{minimize}} && J(\mathbf{w}) \\ & \text{subject to} && \mathbf{1}^T \mathbf{w} = w^{\text{tot}}, \\ & && \mathbf{w} \succeq w^{\min} \mathbf{1} \end{aligned} \quad (10)$$

where $\mathbf{w} \in \mathbb{R}^m$ is the vector of weights on the network graph edges. Here we treat the total sum of weights, $\sum_{j=1}^m w_j = w^{\text{tot}} \geq mw^{\min} \geq 0$, as a resource to be re-distributed among all edges, hence their sum is constrained to be equal to w^{tot} . w^{tot} is assumed to be specified by the initial weight distribution on the network graph G . We consider non-negative edge weights throughout the paper, which further imply $w^{\min} > 0$ to preserve the connectivity and network topology. Note that \mathbf{w} only contains the weights of the existing edges on the graph, thus it is not possible to remove existing edges or add non-existent edges during the optimization.

The optimal edge weights are denoted by \mathbf{w}^* and the corresponding optimal weighted graph is $G^* = (V, E, \mathbf{w}^*)$. In Figure 4, we provide a histogram of eigenvalues of the graph Laplacian matrix (henceforth referred to as the *eigenvalue spectrum*) for both the initial network graph G and the optimized network graph G^* , where both graphs are complete (*i.e.*, there exists an edge between every pair of vertices in V). As can be seen, the optimization has the effect of *flattening* the eigenvalue spectrum, resulting in a more uniform distribution of the eigenvalues, compared to the initial *peaky* spectrum where the eigenvalues are accumulated around a specific value.

Observing that the eigenvalues of the graph Laplacian, $\{\lambda_k\}_{k=1,2,\dots,n}$, and the eigenvalues of Ω , $\{\omega_k\}_{k=1,2,\dots,n}$, are related monotonically as $\omega_k = \sqrt{\lambda_k + \varepsilon}$, the interpretation of this change in the eigenvalue spectrum is as follows: If a graph has a peaky spectrum, an adversarial agent will have a higher chance of success in causing resonance (high-amplitude oscillations) in the graph by choosing the frequency near the peak to pump its forcing signal into the graph. Whereas, with a flattened spectrum, it has less obvious peak to choose from, and hence the overall expected steady-state amplitude is lower.

V. AUXILIARY GRAPH OPTIMIZATION

We consider the scenario where the main network cannot be manipulated directly and the edge weights of the main graph G cannot be modified. An alternative to changing the network itself at the level of individual edges of the network is to connect the network with an *auxiliary network* that is tuned/optimized in a way that minimizes the vulnerability of the main network. This idea of using auxiliary systems to dampen certain frequencies of oscillation appear extensively in the study and design of mechanical and structural systems (such as the use of *tuned mass dampers* in prevention of mechanical vibrations in buildings [19]). We, however, develop the mathematical foundations and methods for designing analogous tuned auxiliary networks for mitigating resonance attacks on the network by an adversarial agent.

In this section, we first reformulate the dynamics equations and the definition of vulnerability based on the *combined network* (main network + auxiliary network). Then, we derive the corresponding objective function and formulate the eigenspectrum optimization problem to minimize the vulnerability of the main network.

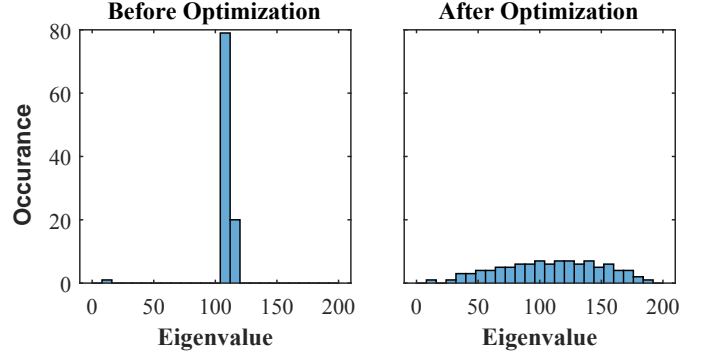


Fig. 4: Histograms of the Laplacian matrix eigenvalues for the initial network graph G and optimized network graph G^* . The initial network is modeled by a complete graph, whose edge weights are perturbed away from a uniform distribution by a small amount. The corresponding eigenspectrum (on the left) is *peaky*, whereas as a result of the eigenspectrum optimization, the eigenvalue spectrum (on the right) has become *flatter*.

A. Reformulation of Combined Dynamics

We denote the graph representation of the auxiliary network by \tilde{G} , and the combined network is denoted by $G \cup \tilde{G}$ (see Figure 5). A second-order unforced signal dynamics on the stand-alone auxiliary network is given by $\ddot{\tilde{\mathbf{x}}} + \tilde{\Gamma}\dot{\tilde{\mathbf{x}}} + \tilde{K}\tilde{\mathbf{x}} = 0$, where $\tilde{\mathbf{x}} \in \mathbb{C}^{\tilde{n}}$ is the signal vector on the vertices of the auxiliary-network, and $\tilde{\Gamma}$ and \tilde{K} are the damping and stiffness matrices respectively that are consistent with the topology of the auxiliary network (in particular, $\tilde{K} = \tilde{\Omega}^2 = \tilde{L} + \varepsilon I$ and $\tilde{\Gamma} = 2\tilde{\gamma}\tilde{\Omega}^2$ (where \tilde{L} is the weighted Laplacian matrix of the auxiliary network and $\tilde{\gamma}$ is the damping multiplier on the auxiliary network).

We make the following simplifying assumptions about the auxiliary network and its inter-connection with the main network:

- We assume the auxiliary network to have the same number of vertices as the main network (that is, $\tilde{n} = n$).
- The above assumption allows a one-to-one connection between the vertices of G and \tilde{G} . The indexing of the vertices of \tilde{G} is done in a way that the k -th vertex of G is assumed to be connected with (and only with) the k -th vertex of \tilde{G} .
- The inter-connecting edges between G and \tilde{G} are assumed to have stiffness (corresponding to a weight of c on those edges), but no damping, allowing the second derivative of the signal on a vertex in G to be coupled with the signal on the neighbor in \tilde{G} , but not its first derivative.
- It's assumed that the adversarial agent can attack vertices of the main network, but not the auxiliary network.
- The connectivity of the auxiliary graph is specified via one of the two types: (1) a *mirrored* auxiliary graph, which exactly mirrors the connectivity of the main graph, and (2) a *complete* auxiliary graph, which is a complete graph. Note that when the main graph is complete, both types correspond to the same auxiliary graph.

Since the auxiliary network is connected to the main network, with the purpose of mitigating the resonance on the main network under adversarial forcing, based on the above assumptions, the signal dynamics over G and \tilde{G} are coupled to give the following signal dynamics on $G \cup \tilde{G}$:

$$\begin{bmatrix} \ddot{\mathbf{x}} \\ \ddot{\tilde{\mathbf{x}}} \end{bmatrix} + \begin{bmatrix} \Gamma & 0 \\ 0 & \tilde{\Gamma} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\tilde{\mathbf{x}}} \end{bmatrix} + \begin{bmatrix} K + cI & -cI \\ -cI & \tilde{K} + cI \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \tilde{\mathbf{x}} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ 0 \end{bmatrix} e^{i\omega t} \quad (11)$$

where the terms cI represent coupling between the dynamics of the two networks due to the one-to-one connection between the vertices of G and \tilde{G} , and affects the stiffness matrix of the combined network,

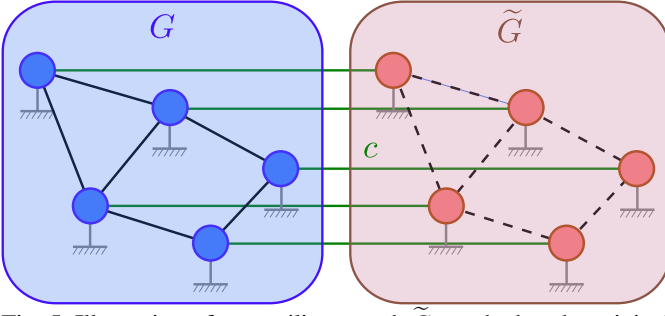


Fig. 5: Illustration of an auxiliary graph \tilde{G} attached to the original graph G with an aim to decrease vulnerability against adversarial attacks. The auxiliary graph is of type *mirrored* (has the same connectivity as the main graph). Green lines indicate the inter-graph connections with weights c .

but not the damping matrix. An illustration of a combined network is provided in Figure 5.

The steady-state solution to (11) is

$$\begin{bmatrix} \mathbf{x}_s \\ \tilde{\mathbf{x}}_s \end{bmatrix} = S^{-1} \begin{bmatrix} \mathbf{f} \\ 0 \end{bmatrix} e^{i\nu t} \quad (12)$$

where,

$$S = \begin{bmatrix} i\nu 2\gamma\Omega^2 + \Omega^2 + (c - \nu^2)I & -cI \\ -cI & i\nu 2\gamma\tilde{\Omega}^2 + \tilde{\Omega}^2 + (c - \nu^2)I \end{bmatrix} \quad (13)$$

However, we note that we are only interested in the response of the main network to the adversarial attacks, which from (12) and (13) is:

$$\mathbf{x}_s = [S^{-1}]_{11} \mathbf{f} e^{i\nu t} \quad (14)$$

where $[S^{-1}]_{11}$ is the top left $n \times n$ block of the inverse of the matrix S , which can be computed using Schur complement of a block matrix [20] as:

$$\begin{aligned} [S^{-1}]_{11} &= ([S]_{11} - [S]_{12}[S]_{22}^{-1}[S]_{21})^{-1} \\ &= \left((i\nu 2\gamma\Omega^2 + \Omega^2 + (c - \nu^2)I) \right. \\ &\quad \left. + c^2 (i\nu 2\gamma\tilde{\Omega}^2 + \tilde{\Omega}^2 + (c - \nu^2)I)^{-1} \right)^{-1} \end{aligned} \quad (15)$$

As a quick sanity check, note that when $c=0$ (which means that the main and the auxiliary networks are not connected), we have

$$[S^{-1}]_{11} = (i\nu 2\gamma\Omega^2 + \Omega^2 - \nu^2 I)^{-1} \quad (16)$$

indicating that the steady-state response on the main network is equivalent to the one derived in equation (3), as expected. In Section VI we use this theoretical result to perform further numerical sanity check on the Auxiliary Graph Optimization objective function.

Although it is not possible to establish a configuration for the auxiliary graph that will guarantee it, we assume that Ω and $\tilde{\Omega}$ are simultaneously diagonalizable. This, using (15), allows us to compute the eigenvalues of $[S^{-1}]_{11}$ as

$$s_k(\nu) = \frac{1}{- \nu^2 + i\nu 2\gamma\omega_k^2 + \omega_k^2 + c - \frac{c^2}{- \nu^2 + i\nu 2\gamma\tilde{\omega}_k^2 + \tilde{\omega}_k^2 + c}} \quad (17)$$

where $\omega_k, \tilde{\omega}_k$ are the k -th eigenvalues of Ω and $\tilde{\Omega}$ respectively. Later on, we will show that there will be an approximation error between the computed expected value and the average squared 2-norm of the steady-state response when Ω and $\tilde{\Omega}$ are not simultaneously diagonalizable.

As before, we assume that \mathbf{f} is being uniformly sampled from $(n-1)$ -unit sphere and the adversarial agent only has imprecise information about the main graph as described in Section IV-B (i.e., it has no information about the auxiliary graph and hence the combined

network). Using the results from the derivation of the Network Graph Optimization method (Section IV), we compute the expected-value of the random variable $\|\mathbf{x}_s\|_2^2$, hence the vulnerability of the main network when it is attached to an auxiliary network as:

$$\begin{aligned} \mathbb{E}_{\mathbf{f}, \nu}(\|\mathbf{x}_s\|_2^2) &= \int_{-\infty}^{\infty} \mathbb{E}_{\mathbf{f}}[\|\mathbf{x}_s\|_2^2] \rho(\nu) d\nu \\ &= \frac{1}{n^2} \sum_{k,j=1}^n \int_{-\infty}^{\infty} s_k(\nu) \bar{s}_j(\nu) \rho_{\omega_j}(\nu) d\nu \end{aligned} \quad (18)$$

where \bar{s}_k denotes the complex conjugate of s_k and $\rho_{\omega_j}(\nu) = \frac{h/\pi}{(\omega_j - \nu)^2 + h^2}$. A closed form expression for the integral is obtained using the Residue theorem with the same contour as before as described in Section IV-B. In order to use the Residue theorem as described, however, one needs to compute the roots of the quartic polynomial in ν in the denominator of the integrand and determine whether those roots have positive or negative imaginary parts. In this case a direct computation of that, even using a symbolic algebra toolbox, was not feasible because of the complexity of the problem. In order to simplify computation of the roots, we use linearization with respect to γ . The details of the computation are provided in Appendix B. Corresponding calculations are performed using a symbolic mathematics toolbox. We omit the resulting expression for brevity.

Assuming that the main graph G and the parameters n, h, γ remain constant, the objective is to minimize $\mathbb{E}_{\mathbf{f}, \nu}(\|\mathbf{x}_s\|_2^2)$, which is a function of the eigenvalues of the auxiliary stiffness matrix $\tilde{\Omega}$ (which, in turn, are functions of the weights on the auxiliary graph edges, $\tilde{\mathbf{w}}$), the uniform inter-graph edge weight c and the auxiliary damping factor $\tilde{\gamma}$. For the purposes of this paper, we assume $\tilde{\gamma}$ to be a small constant, in order to allow signals transmitted over the network (non-adversarial) to persist and not get dissipated too quickly. The resulting objective function is thus defined as:

$$\tilde{J}(\tilde{\mathbf{w}}, c) = \mathbb{E}_{\mathbf{f}, \nu}(\|\mathbf{x}_s\|_2^2) \quad (19)$$

B. Eigenspectrum Optimization of the Auxiliary Network Graph

We define the eigenspectrum optimization problem of the auxiliary network graph as follows:

$$\begin{aligned} &\underset{\tilde{\mathbf{w}}, c}{\text{minimize}} && \tilde{J}(\tilde{\mathbf{w}}, c) \\ &\text{subject to} && \mathbf{0} \preceq \tilde{\mathbf{w}}, \\ & && 0 \leq c, \\ & && \mathbf{1}^T \tilde{\mathbf{w}} + nc \leq r_m w^{\text{tot}} \end{aligned} \quad (20)$$

Here, we assume that the weight resource is specified as a multiple of the total weights on the main graph (denoted by $r_m w^{\text{tot}}$) which is to be distributed among the auxiliary graph and inter-graph edges. We consider non-negative edge weights throughout, without any additional lower bound.

The optimal auxiliary graph edge weights are denoted by $\tilde{\mathbf{w}}^*$, the optimal inter-graph edge weight is c^* and the corresponding optimal auxiliary graph weight configuration is \tilde{G}^* .

Note that it is also possible to consider the case where the auxiliary damping multiplier $\tilde{\gamma}$ is a decision variable. We include further discussion on the effects of the auxiliary damping and experimental results in Section VI-C4.

VI. RESULTS

In this section, we present experiments conducted to accomplish the following:

- Validate the accuracy of the objective functions, J and \tilde{J} , in representing the network vulnerability measured by $\mathbb{E}(\|\mathbf{x}_s\|_2^2)$ for \mathbf{x}_s defined on G and \tilde{G} , as described equations (9) and (18) respectively.
- Analyze the effects of the problem parameters associated with the network dynamics and constraints on the relative vulnerability decrease that can be achieved via the proposed methods.
- Demonstrate the effectiveness of the proposed methods in decreasing the network vulnerability across a variety of problem instances.
- Perform numerical simulation of dynamics over a network to further validate the results achieved by the Network Graph Optimization.

A. Implementation Details and Setup

To solve the network graph and auxiliary graph eigenspectrum optimizations, we use the *interior-point* algorithm implemented within the *fmincon* function from MATLAB Optimization Toolbox [21].

1) *Network Graph Construction*: All algorithms are implemented and tested on three classes of network graphs:

- Random Complete Graphs* (“RCG”): Given a desired number of vertices, n , we establish an edge between every pair of vertices, thus resulting in a graph with $n_e = \dim \mathbf{w} = \binom{n}{2}$ edges. We then sample the weight for each edge from a uniform distribution on the interval $[1 - w_p, 1 + w_p]$, where w_p is a given *weight perturbation*.
- Random Incomplete Graphs* (“RIG”): Given a desired number of vertices, n , and a desired number of edges, $n_e = \dim \mathbf{w} < \binom{n}{2}$, we randomly chose n_e distinct pair of vertices to establish the edges between. Weights for the edges are sampled from a uniform distribution on the interval $[1 - w_p, 1 + w_p]$.
- Social Network Graphs* (“Social”): As a representative of real-world networks, we extracted subgraphs from the “Government” graph category of the Gemsec Facebook Dataset [22] which encompasses various graphs representing blue verified Facebook page networks. To generate the subgraphs, ego graphs with a radius of 2 were created. Nodes were randomly selected without replacement to serve as the center of each ego graph. Only the first 100 subgraphs containing between 25 and 200 vertices that were generated were selected, resulting in a set of 100 subgraphs with an average 109.82 vertices and 867.69 edges per subgraph.

2) *Adversarial Force Sampling*: For computing steady-state amplitudes for specific instances of simulation for a given graph, we need to sample the adversarial forcing vector, \mathbf{f} , and the adversarial forcing frequency, v .

As described in Section IV-A, we assume that the forcing vector \mathbf{f} is sampled from an $(n-1)$ -dimensional unit sphere. This is achieved by sampling each element of the vector from the standard normal distribution, and then normalizing the vector [23]. Here we highlight that the necessary number of forcing vector samples to cover the sphere surface increases exponentially as the size of the network, n , (the dimension of the forcing vector/unit sphere) increases, if the sample dispersion is to be maintained. This comes from the fact that the dispersion is inversely proportional to the sample size and the dimension [24], [25].

As described in Section IV-B, the forcing frequency needs to be sampled using a probability density function that is a uniformly weighted sum of multiple Cauchy distributions each of which are centered at the natural frequencies, $\{\omega_j\}_{j=1,\dots,n}$, with a constant spread of h . However, in order to perform this sampling, one needs to compute the inverse of the cumulative distribution function (c.d.f.) of the ρ described in (5), which is computationally difficult. We thus adopt a practical method that is used to generate samples from mixture models as explained in [26]: Using an uniform probability of $1/n$ on

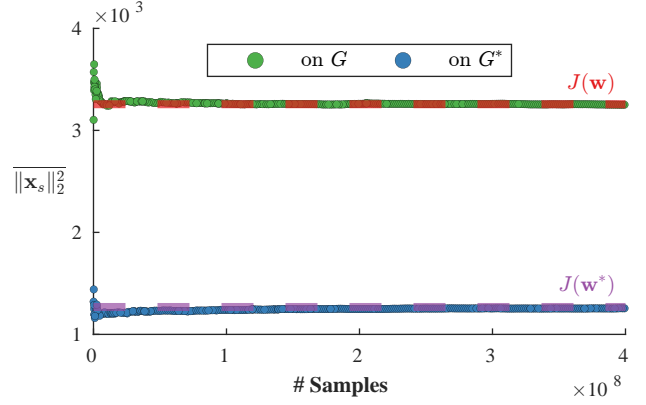


Fig. 6: Squared 2-norm of the steady state response \mathbf{x}_s is evaluated in closed-form using sampled external forcing. This plot shows the running average of $\|\mathbf{x}_s\|_2^2$ against the number of forcing samples. The running averages are computed using batches of 100K samples. Over large enough samples, the average $\|\mathbf{x}_s\|_2^2$ evaluated on the initial graph G and the optimized graph G^* converge to the values of the objective function evaluated at \mathbf{w} and \mathbf{w}^* respectively.

all the natural frequencies, $\{\omega_j\}_{j=1,\dots,n}$, we first sample one of the natural frequencies, say ω_s . Then the forcing frequency is sampled from a Cauchy distribution centred at ω_s and with a spread of h using the inverse c.d.f. of Cauchy distribution, $v = \omega_s + h \tan(\pi(p - 0.5))$, where p is sampled from an uniform distribution over the unit interval $[0, 1]$.

B. Network Graph Optimization

First we present the results from Network Graph Optimization.

1) *Validation of the Objective Function*: The objective function, J (equation (9)), for the network graph eigenspectrum optimization problem is the expected value of the squared 2-norm of the steady-state response of the dynamic network subject to adversarial forcing with the stochastic model explained in Sections IV-A, IV-B.

To validate the the accuracy of the objective function in representing the expected value, we generate 400M adversarial forcing samples (using the procedure explained in Section VI-A), evaluate the closed-form steady state response, \mathbf{x}_s , for each sample using equation (3) and compute the average squared 2-norm of the responses $\|\mathbf{x}_s\|_2^2$. For the validation study, we use a RCG with $n = 10$ and $w_p = 0.3$. The running average over the number of samples divided in multiple batches are provided in Figure 6.

It can be observed that over a large number of forcing samples, the average of the squared 2-norm of the steady-state responses is well approximated by the objective values for both initial and optimized graphs. Hence, $\mathbb{E}_{\mathbf{f}, v}(\|\mathbf{x}_s\|_2^2)$ is accurately represented by J . Consequently, it can be seen that on the optimized graph, the steady state responses have smaller amplitudes on average. Following this validation, we can use the objective value as a measure of a graph’s vulnerability to adversarial attacks, where a lower objective function indicates less vulnerability.

2) *Parameter Analysis*: Each eigenspectrum optimization problem on a network graph can be specified via a set of parameters regarding the second order dynamics of the network, the external forcing and the constraints. These parameters are the number of vertices on the graph (n), the number of edges on the graph (n_e), the minimum weight constraint (w^{min}), the stiffness constant (ϵ), the damping factor (γ), and the spread of the external agent’s frequency distribution (h).

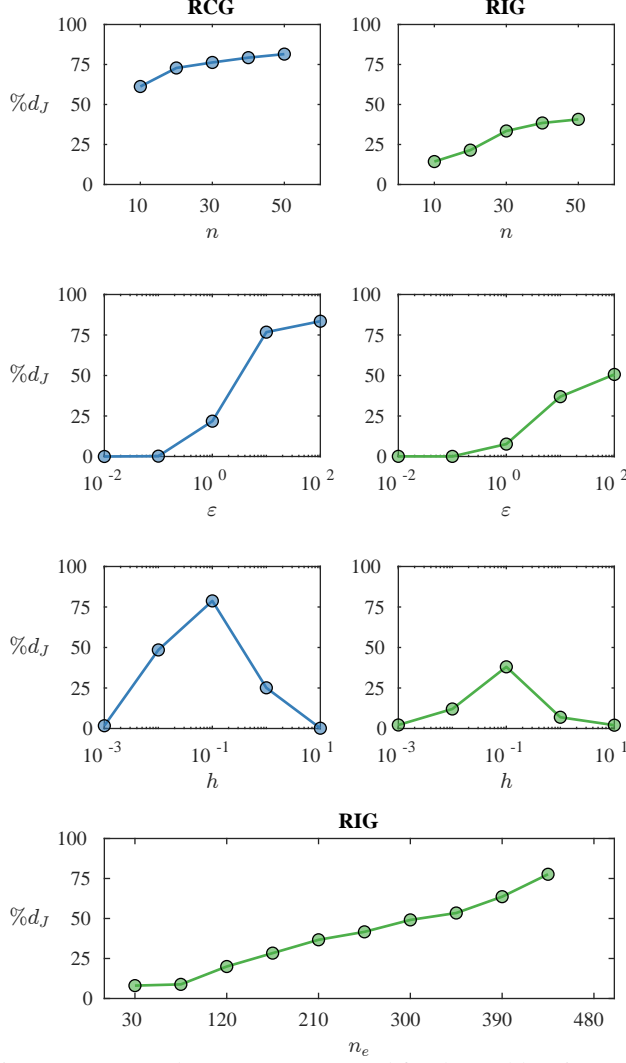


Fig. 7: A RCG and a RIG are generated for the problem instances specified by each set of parameters (only a RIG is generated for the case where the variable parameter is the number of edges). The optimization problem is solved for each instance, and the percentage decrease in objective values are plotted against the varying parameter.

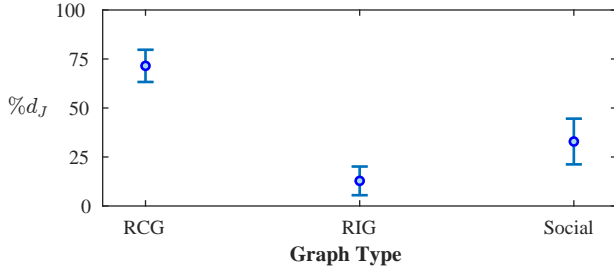


Fig. 8: The network graph eigenspectrum optimization problem is solved for instances featuring 100 RCGs, 100 RIGs and 100 social network graphs. On problems instances where the main network graphs are complete, the optimization consistently yielded larger relative decrease in the objective values.

We analyze the effects of these parameters on the percentage reduction of objective value that can be achieved via the eigenspectrum optimization, hence the reduction in the vulnerability of the main network graph using the Network Graph Optimization method. For this purpose, we start with set of parameter values, $n=30$, $n_e=225$, $w^{min}=10^{-3}$, $\varepsilon=10$, $\gamma=10^{-6}$, $h=0.1$, and generate problem instances featuring both RCGs and RIGs where we vary one parameter and keep the rest constant. We solve for each problem instance and compute the percentage reductions in objective as $\%d_J = \frac{|J^0 - J^*|}{J^0} \times 100$ (where superscripts 0 and * denote initial and optimal objective values), which are plotted against the varying parameter values in Figure 7. Note that by comparing the percentage decrease in the objectives instead of the final objective values achieved, we are trying to isolate the effect of the parameters on the effectiveness of Network Graph Optimization method in reducing the vulnerability of a graph instead of trying to find the set of problem parameters that make the network the least vulnerable.

It can be observed from Figure 7 that a larger decrease in the objective value can be achieved as the number of vertices or the number of edges increase. Intuitively, more vertices and more edges correspond to more flexibility in distributing the weight resources, thus resulting in larger improvements to the vulnerability of the graph. As expected, larger stiffness yields better results, where the effect gets more significant with increased orders of magnitude. The spread of the external agent's frequency distribution has a non-monotonic effect. As the spread gets smaller, the agent is able to pick the resonance frequencies more accurately, leaving the graph helpless against the attack, whereas a larger frequency spread corresponds to an agent that almost arbitrarily picks its frequencies, against which any modification of the graph based on reasoning would be less effective. Since the minimum weight constraint and the damping factor did not demonstrate a significant effect on the percentage decrease of the objective, corresponding plots are excluded. By observing the plots overall and the analysis on the number of edges, it is clear that the eigenspectrum optimization on a main network graph is more effective when the graph is complete. This behavior will become more apparent in the next section.

3) Demonstration of the Effectiveness of Network Graph Optimization: To demonstrate the overall effectiveness of eigenspectrum optimization on the main network graph in reducing the network vulnerability, we solve the optimization problem for RCGs, RIGs and *Social* graphs, and show that significant decrease in objective values can be achieved. We generate 100 RCGs and RIGs with n sampled uniformly from the interval $[10, 30]$ and w_p sampled uniformly from the interval $[0.1, 0.5]$. For the RIGs, we sampled n_e from the interval $[n, n^2/4]$. The parameters associated with the network dynamics are the minimum weight, $w^{min}=0.001$, the stiffness constant, $\varepsilon=10$, the damping coefficient, $\gamma=10^{-6}$, and the adversarial agent's frequency spread, $h=0.1$. The average percentage decrease in the objective value and the standard deviation across the problem instances featuring RCGs, RIGs and *Social* graphs are provided in Figure 8. Qualitatively, on the eigenspectrum of the graph, the optimization is manifested as a *flattening* of the eigenspectrum, as can be seen for the complete graph in Figure 15 and the *Social* graph in Figure 16.

As mentioned before, network graph eigenspectrum optimization is more successful at reducing the objective value relative to the initial value of the objective when it is performed on complete graphs. A reason for this behavior is the greater vulnerability of the complete graphs to the resonance attacks, due to the fact that the natural frequencies of a complete graph are heavily accumulated around a value resulting in a *peaky* spectrum, compared to a relatively flatter/uniform distribution of the natural frequencies on an incomplete graph. A fewer number of op-

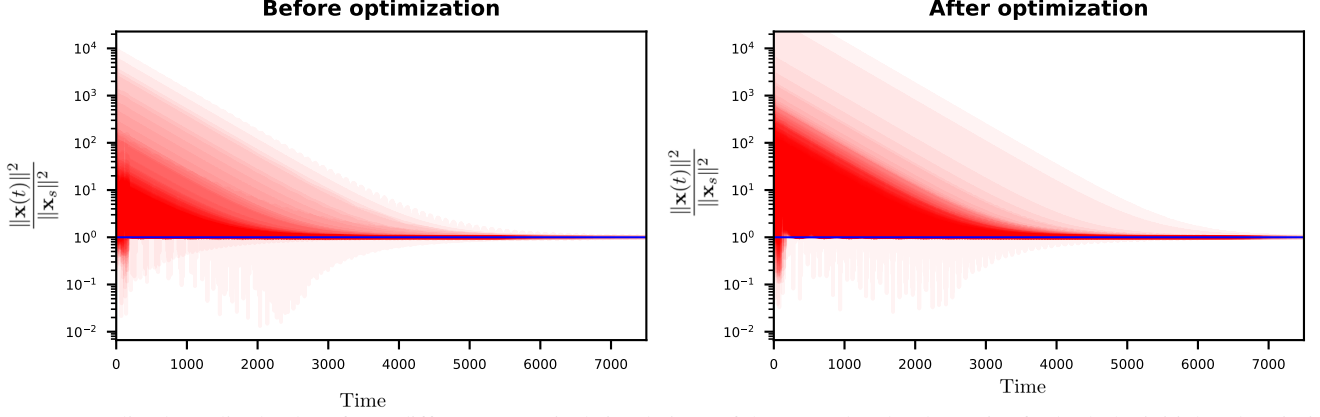


Fig. 9: Normalized amplitude plot of 100 different numerical simulations of the second-order dynamics for both the initial and optimized complete graph, displaying their corresponding two-norm squared amplitude values divided by the steady-state squared amplitude value calculated by the closed-form evaluation for the respective forcing vector and forcing frequency (i.e., $\frac{\|\mathbf{x}(t)\|_2^2}{\|\mathbf{x}_s\|_2^2}$). Since the two-norm squared amplitude value from the simulation of the second-order dynamics should converge to the same closed-form steady-state evaluation for the two-norm squared amplitude given the same corresponding forcing vector and forcing frequency, the values in the plot are expected to approach 1, as indicated by the blue line, which they indeed do. In these 100 simulations, the initial graph's mean steady-state squared amplitude value is 0.0899, while the optimized graph's mean steady-state squared amplitude value is 0.0145.

timization variables impose greater rigidity on incomplete graphs due to their fewer edges, whereas complete graphs, with their maximum possible number of edges, offer a greater flexibility in edge weight manipulations. Qualitatively, this is manifested by a lower relative flattening of the eigenspectrum in case of the incomplete *Social* graph (Figure 16) as compared to the complete graph (Figure 15). An embedding of an optimized *Social* graph is shown in Figure 17

4) *Numerical Second-Order Dynamics Simulation of the Main Network*: We simulated (performed numerical integration of) the second-order dynamics on 100 different graphs, with both the initial and optimized weights with varying forcing vectors and sampled forcing frequencies. The simulations were run until a steady state was achieved and the final steady-state amplitude was noted.

Complete Graph Numerical Simulations: We considered an unoptimized complete graph with uniform edge weights with an added perturbation as detailed in Section VI-A, as well as the corresponding optimized graph obtained using the network graph optimization method detailed in Section IV, and performed 100 numerical simulations on each of these graphs. The squared amplitude of \mathbf{x} as a function of time for each of the 100 simulations, each normalized by the closed-form steady-state squared amplitude $\|\mathbf{x}_s\|_2^2$, is shown in Figure 9. Besides observing that the steady-state amplitudes of the numerical simulations match the computed closed-form values, we note that the unsteady amplitude in relation to the steady-state amplitude has less variation in the optimized graph.

C. Auxiliary Graph Optimization

For the Auxiliary Graph Optimization approach, we conduct similar experiments and provide additional analysis on the effects of auxiliary damping.

1) *Validation of the Objective Function*: The objective function \tilde{J} for the auxiliary graph eigenspectrum optimization problem is the expected value of the squared 2-norm of the steady-state response corresponding to the main network vertices when the dynamic network is subject to stochastic adversarial forcing. To validate the accuracy of the objective function in representing the expected value, we generate 800M adversarial forcing samples, evaluate the closed-form steady

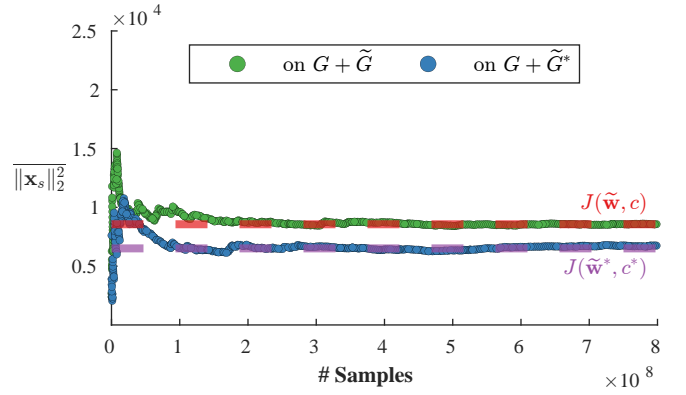


Fig. 10: Squared 2-norm of the steady state response corresponding to the main graph \mathbf{x}_s is evaluated in closed-form using sampled external forcing. This plot shows the running average of $\|\mathbf{x}_s\|_2^2$ against the number of forcing samples. The running averages are computed using batches of 100K samples. Over large enough samples, the average $\|\mathbf{x}_s\|_2^2$ evaluated on the unoptimized combined network system $G + \tilde{G}$ and the optimized system $G + \tilde{G}^*$ converge to the values of the objective function evaluated at $(\tilde{\mathbf{w}}, c)$ and $(\tilde{\mathbf{w}}^*, c^*)$ respectively.

state responses for each sample using equation (14) and compute the average squared 2-norm of the responses. For the validation study, we generate a RCG with $n=10$, with $w_p=0.3$ and use it as the main network graph. The running average over the number of samples divided in multiple batches are provided in Figure 10.

The problem instance generated for the validation study resulted in an optimized auxiliary graph for which Ω and $\tilde{\Omega}^*$ are simultaneously diagonalizable. As a consequence, we observe that $\|\mathbf{x}_s\|_2^2$ converge to the objective values for both the optimized and unoptimized combined networks. To demonstrate the fact that there will be an approximation error between $\|\mathbf{x}_s\|_2^2$ and \tilde{J} , when Ω and $\tilde{\Omega}^*$ are not simultaneously diagonalizable, we perform another auxiliary graph eigenspectrum optimization based on a RIG with $n=10$, $n_e=25$ and $w_p=0.3$. The running average over the number of samples divided in multiple batches are provided in Figure 11.

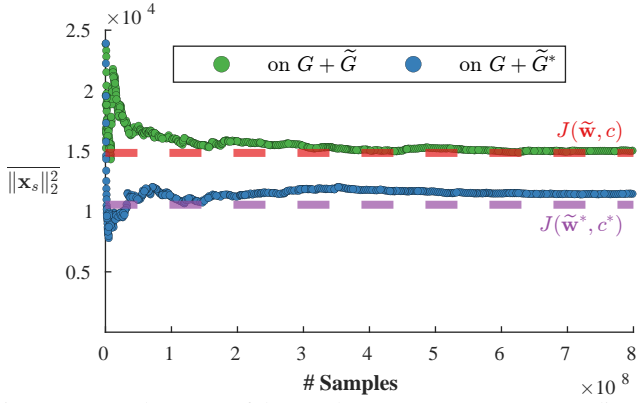


Fig. 11: Squared 2-norm of the steady state response corresponding to the main graph \mathbf{x}_s is evaluated in closed-form using sampled external forcing. This plot shows the running average of $\|\mathbf{x}_s\|_2^2$ against the number of forcing samples. The running averages are computed using batches of 100K samples. Over large enough samples, the average $\|\mathbf{x}_s\|_2^2$ evaluated on the unoptimized combined network system $G + \tilde{G}$ converges to the values of the objective function evaluated at $(\tilde{\mathbf{w}}, c)$. However, when evaluated on the optimized system $G + \tilde{G}^*$ the average $\|\mathbf{x}_s\|_2^2$ does not converge to the value of the objective function evaluated at $(\tilde{\mathbf{w}}^*, c^*)$, resulting in an approximation error.

From Figures 10 and 11, it can be observed that over a large number of forcing samples, the average of the squared 2-norm of the steady-state responses is well approximated by the objective values when Ω and $\tilde{\Omega}$ are simultaneously diagonalizable, whereas there exist an approximation error when these matrices are not simultaneously diagonalizable. Also, it can be seen that on the optimized graphs, the steady state responses have smaller amplitudes on average.

As a sanity check, we leverage the theoretical result provided in equation (16) and confirm that both objective functions J and \tilde{J} match when evaluated numerically for arbitrary choices of Ω and $\tilde{\Omega}$ when $c=0$.

Following the validation of the objective function \tilde{J} , we can use the objective value as a measure of a graph's vulnerability to adversarial attacks, where a lower objective function indicates less vulnerability.

2) *Parameter Analysis:* Parameters that specify an eigenspectrum optimization problem on an auxiliary graph is similar to those of network graph optimization. Since the auxiliary graph edges and inter-graph edges are assumed to have non-negative weights, we do not consider the minimum weight constraint (w^{min}) parameter in this case. However, in addition to the network graph optimization parameters, we must consider the effects of the following parameters associated with auxiliary graphs: the auxiliary connectivity type (*mirrored* or *complete*), the weights resource multiplier r_m , and the auxiliary damping factor $\tilde{\gamma}$. We defer the analysis of the auxiliary damping factor to Section VI-C4 and use a constant auxiliary damping factor of $\tilde{\gamma} = 10^{-6}$ throughout the parameter analysis.

We analyze the effects of these parameters on the percentage reduction of objective value that can be achieved via the auxiliary graph optimization, hence the relative decrease in the vulnerability of the graph using the AGO method. We start with the same set of parameter values with the addition of $r_m = 5$, and generate problem instances where we vary one parameter and keep the rest constant. We solve for each problem instance and compute the percentage reductions in objective as $\%d_J = \frac{|J^0 - \tilde{J}|}{J^0} \times 100$, which are plotted against the varying parameter values in Figure 12. Note that for the problem instances where the main network graph is a RIG, we provide two sets of results achieved

with a mirrored auxiliary graph and a complete auxiliary graph.

Here we highlight that the percentage reduction of the objective value is computed based on the value of the objective before the auxiliary graph is attached, that is J^0 , instead of the objective value evaluated using an unoptimized auxiliary graph, that is $\tilde{J}^0 = \tilde{J}(\tilde{\mathbf{w}}, c)$. The individual effects of attaching an arbitrary auxiliary graph, and the optimization of the auxiliary graph will be presented in the next section.

For all parameters, effects are similar to those on the network graph optimization. However, even for the parameter values for which the NGO was less effective, the AGO method can achieve larger decreases in the objective, which makes the approach less sensitive to the choice of the parameters. The same insensitivity is observed to the weight resource multiplier parameter. For the instances where the network graph was incomplete, some of the optimizations of the mirrored auxiliary graph failed to converge in the maximum number of iterations considered, which is indicated by a 0% decrease in the plots.

3) *Demonstration of the Effectiveness of AGO:* To demonstrate the overall effectiveness of eigenspectrum optimization on the auxiliary graph in reducing the network vulnerability, we solve the optimization problem for RCGs and RIGs and show that significant decrease in objective values can be achieved. We use the same problem instances generated for the NGO, with $r_m = 5$ and $\tilde{\gamma} = 10^{-6}$ and using complete auxiliary graphs. To demonstrate the effects of attaching an arbitrary auxiliary graph and the optimization of this auxiliary graph separately, we provide the average and the standard deviation of the percentage decrease in the objective calculated as (1) $\%d_J = \frac{|J^0 - \tilde{J}|}{J^0} \times 100$ (decrease achieved by going from network configuration G to $G + \tilde{G}$), and (2) $\%d_J = \frac{|J^0 - \tilde{J}^*|}{J^0} \times 100$ (decrease achieved by going from network configuration G to $G + \tilde{G}^*$) across the problem instances featuring complete and incomplete main network graphs are provided in Figure 13.

It can be seen that attaching even an arbitrary auxiliary graph decreases the vulnerability of the network significantly. However, performing the optimization over the auxiliary edge weights and inter-graph edges results in a further decrease of the vulnerability and provides more consistent behavior.

4) *Effect of the Auxiliary Damping and Auxiliary Damping Optimization:* Assuming that the auxiliary graph weights and the inter-graph edge weights are constant, the auxiliary objective function \tilde{J} becomes a function of the auxiliary damping factor $\tilde{\gamma}$ only. Furthermore, if the auxiliary damping is uniform across all auxiliary vertices, \tilde{J} is a single-variable function. To visualize the effect of the auxiliary damping, we evaluate \tilde{J} on an optimized combined network (specified by G, G^*, c^*) with $\tilde{\gamma}$ varying logarithmically on the interval $[10^{-6}, 10^5]$. The objective values are plotted against the auxiliary damping factor in Figure 14.

We observe that the objective function \tilde{J} is highly sensitive to the value of the auxiliary damping $\tilde{\gamma}$ and that one can significantly decrease the objective value by setting the auxiliary damping to be larger than the damping on the main network. However, simply setting the auxiliary damping to the maximum allowed value does not yield the smallest objective value as observed from Figure 14. To the best of our understanding, as the auxiliary damping gets larger than the optimal value, the auxiliary network loses the ability to dissipate the signal that is being transmitted from the main network and the signal tends to bounce back causing a resonance. For this reason, optimizing over the variable $\tilde{\gamma}$ could provide further improvements if the goal is to achieve the least possible vulnerability in the network.

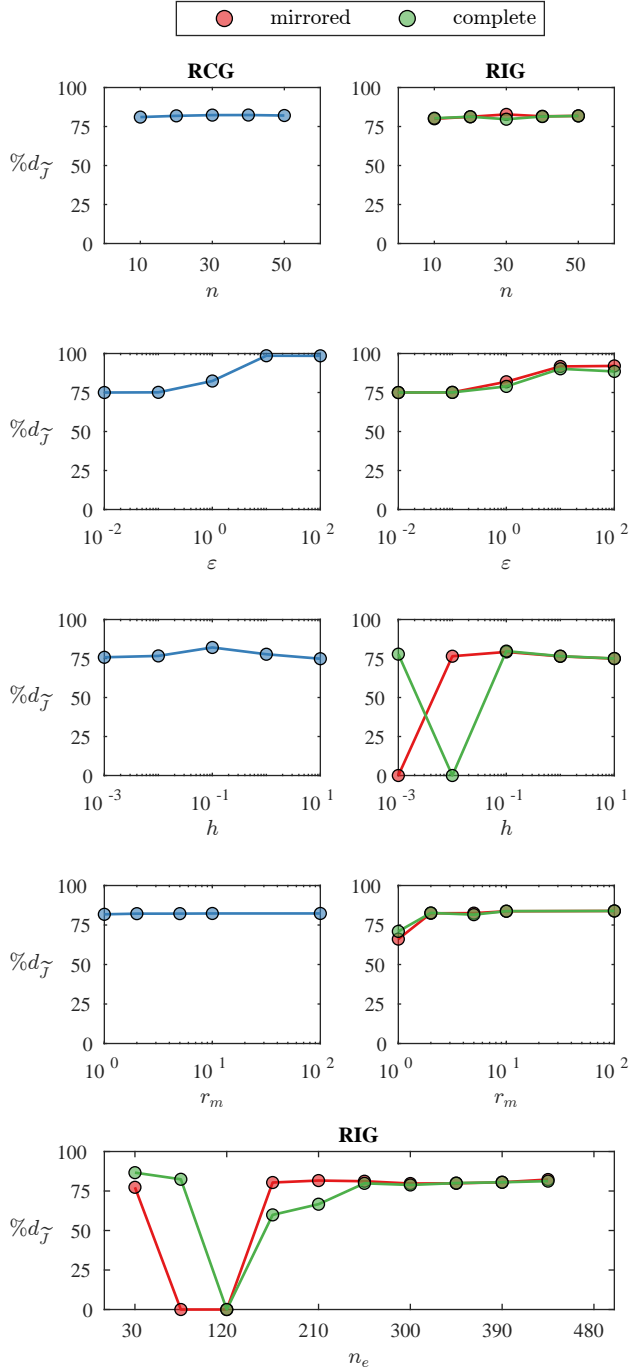


Fig. 12: A RCG and a RIG are generated for the problem instances specified by each set of parameters (only a RIG is generated for the case where the variable parameter is the number of edges). The optimization problem is solved for each instance (using both mirrored and complete auxiliary graphs for instances where the network graph is incomplete), and the percentage decrease in objective values ($\%d_{\tilde{J}}$) are plotted against the varying parameter. Data points where the percentage decrease is at 0 indicate the instances where the optimization failed to converge within the maximum number of iterations.

VII. CONCLUSION AND DISCUSSIONS

In this paper, we developed the notion of vulnerability of a network with second order signal dynamics under adversarial forcing that obeys a known stochastic model. To minimize the network vulnerability, we proposed two methods that optimize the network

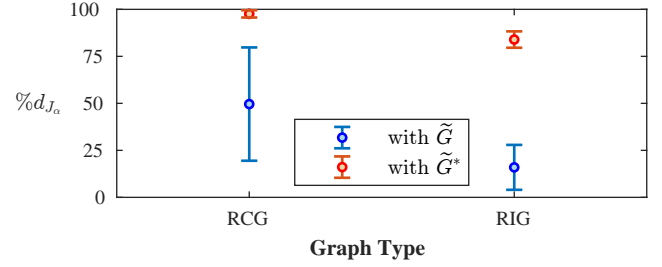


Fig. 13: The auxiliary graph eigenspectrum optimization problem is solved for instances featuring 100 RCGs and 100 RIGs. We report the average and standard deviation of the percentage decrease in the objective achieved by both going from the network configuration G to $G + \tilde{G}$ and from the network configuration G to $G + \tilde{G}^*$. Success rates for running AGO on RCGs and RIGs were %100 and %97 respectively.

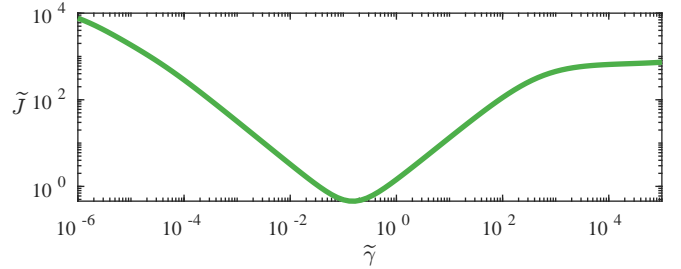


Fig. 14: Value of the auxiliary objective function \tilde{J} evaluated on an optimized combined network (specified by G, \tilde{G}^*, c^*) with auxiliary damping factor $\tilde{\gamma}$ on the interval $[10^{-6}, 10^5]$.

structure: *i*. The Network Graph Optimization (NGO) method provides an optimal set of network edge weights under the condition that the edge weights can be directly manipulated, and, *ii*. The Auxiliary Graph Optimization (AGO) method allows us to design an auxiliary network that can be attached to the main network with the purpose of minimizing the vulnerability, when the main network edge weights cannot be adjusted directly. We conducted numerical experiments to analyze the two methods in detail.

Currently, the notion of vulnerability and the optimization problems posed in this work depend on a linear model of the signal dynamics and a specific stochastic model of adversarial forcing. While the adaptation of some aspects of the model to other setting (e.g., a different stochastic model of adversarial forcing) can be straightforward re-derivation of the objective functions, a more general formulation that encompasses more complicated signal models, forcing models, and potentially nonlinear signal dynamics, is within the scope of future work. The optimization formulations presented in this paper lead to generally non-convex problems which are in turn solved by gradient based solvers. A formal analysis of the optimization landscape would be necessary to provide any guarantees on the quality of the solution being returned and is within the scope of future work.

The current optimization problem is formulated as a centralized one that assumes complete knowledge of the network graph edge weights. A potential future work involves the development of a distributed optimization scheme in which each vertex would use information about its local subgraph and would only adjust weights on its incident edges in order to optimize the network. A distributed method would allow the approach to scale to larger networks and generalize to settings where global information regarding the network may not be available due to privacy restrictions.

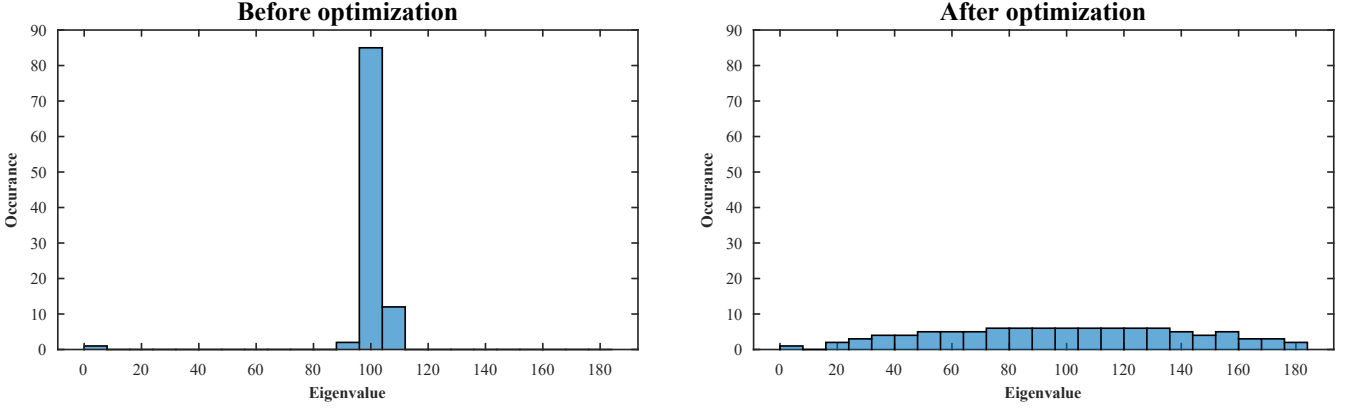


Fig. 15: Example of eigenvalue spectrum of a complete graph before and after optimization represented as histograms of the eigenvalues of the networks stiffness matrix, $L + \epsilon I$. The objective value significantly decreased from 1.378 to 0.3778, yielding a 72.58% decrease as a result of the network graph optimization on a 100-node RCG. Qualitatively, as a result of the network graph optimization, the eigenvalue spectrum has become *flatter*.

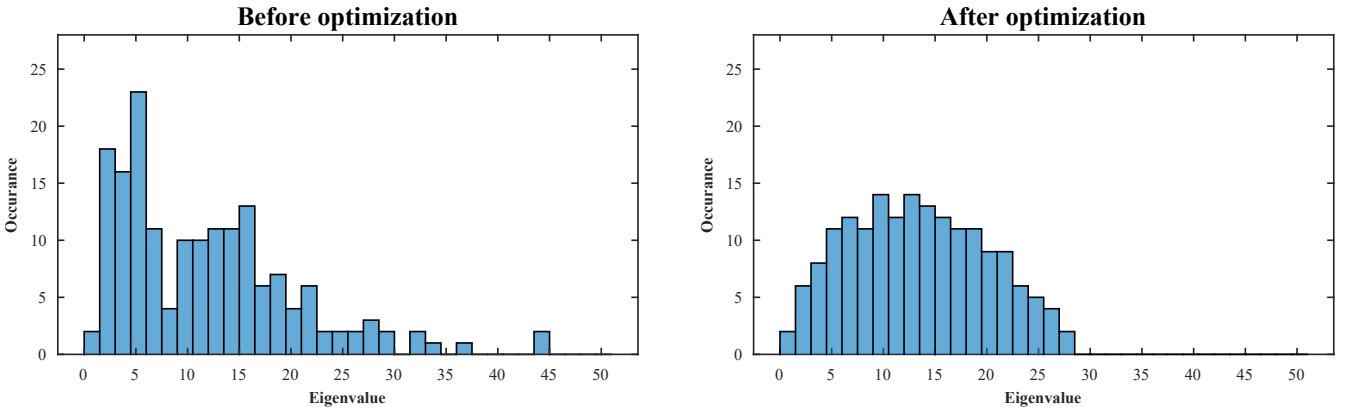


Fig. 16: Example of eigenvalue spectrum of a representative Facebook Social subgraph before and after optimization represented as histograms of the eigenvalues of the networks stiffness matrix, $L + \epsilon I$. The objective value significantly decreased from 6.868 to 2.467 as a result of the network graph optimization on the 173-node Facebook social subgraph, corresponding to a 64.089% decrease in the objective value. As a result of the optimization, the eigenvalue spectrum has become smoother and flatter.

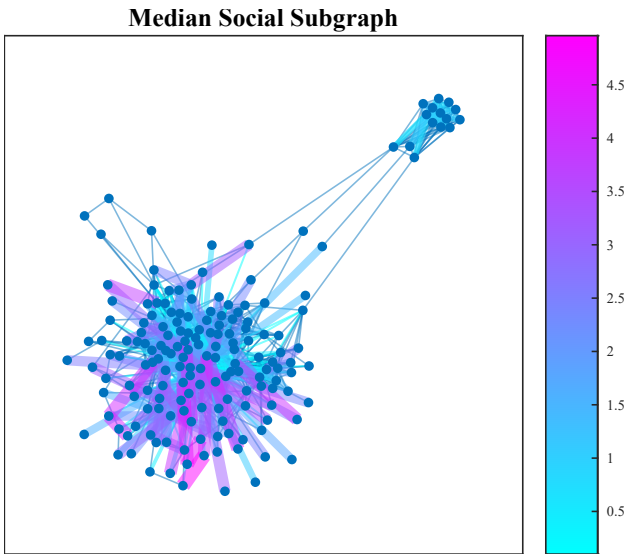


Fig. 17: An embedding of an example Facebook Social Subgraph with 173 vertices. The colors on the edges indicate weights after optimization.

APPENDIX

A. Proof of Lemma 1

Statement of the Lemma If $\mathbf{f} \in \mathbb{R}^n$ is sampled from an uniform distribution over a $(n-1)$ -unit sphere and M is a symmetric matrix, then

$$\mathbb{E}_{\mathbf{f}}(\|\mathbf{M}\mathbf{f}\|_2^2) = \frac{1}{n} \|\mathbf{M}\|_F^2$$

where $\|\cdot\|_F$ is the Frobenius norm.

Proof. Suppose M is diagonalized by the orthogonal matrix, U , so that $M = UDU^T$, where $D = \text{diag}(d_1, d_2, \dots, d_n)$ is the diagonal matrix of the eigenvalues of M .

Because of rotational symmetry of the distribution of \mathbf{f} (uniform distribution over a sphere), the expected value of $\|\mathbf{M}\mathbf{f}\|_2^2$ is independent of the choice of (an orthonormal) basis, and in particular, is the same in the basis of the eigenvectors of M . Thus,

$$\mathbb{E}_{\mathbf{f}}(\|\mathbf{M}\mathbf{f}\|_2^2) = \mathbb{E}_{\mathbf{f}}(\|\mathbf{D}\mathbf{f}\|_2^2) = \mathbb{E}_{\mathbf{f}}\left(\sum_{j=1}^n d_j^2 f_j^2\right) = \sum_{j=1}^n d_j^2 \mathbb{E}_{\mathbf{f}}(f_j^2) \quad (21)$$

where $\mathbb{E}_{\mathbf{f}}(f_j^2)$ is the expected value of the square of the j -th component of \mathbf{f} .

However, we note that because of the spherical symmetry of the distribution of \mathbf{f} , we must have $\mathbb{E}_{\mathbf{f}}(f_1^2) = \mathbb{E}_{\mathbf{f}}(f_2^2) = \dots = \mathbb{E}_{\mathbf{f}}(f_n^2) =: \xi$. Thus,

$$\begin{aligned} \mathbb{E}_{\mathbf{f}}(\|\mathbf{f}\|_2^2) &= 1 = \sum_{j=1}^n \mathbb{E}_{\mathbf{f}}(f_j^2) = n\xi \\ \Rightarrow \xi &= 1/n \end{aligned} \quad (22)$$

Hence from (21) we have, $\mathbb{E}_{\mathbf{f}}(\|\mathbf{M}\mathbf{f}\|_2^2) = \sum_{j=1}^n d_j^2 / n = \frac{1}{n} \|\mathbf{M}\|_F^2$. \square

B. Approximate Root Computation Using Linearization

Consider a polynomial in the variable $x \in \mathbb{C}$, given by $Q(x, \gamma)$, where $\gamma \in \mathbb{R}$ is a parameter involved in the coefficients of the polynomial. We are interested in approximately computing the roots of the polynomial for a general small, positive parameter value, γ , given the roots of the polynomial when $\gamma=0$ (which is presumed to be easier to compute).

If $\{r_k(\gamma)\}_{k=1,2,\dots,n}$ are the roots of the polynomial $Q(x, \gamma)$ (possibly with multiplicity), we have

$$\begin{aligned} Q(x, \gamma) &= \prod_{k=1}^n (x - r_k(\gamma)) \\ \Rightarrow \frac{\partial Q}{\partial \gamma}(x, \gamma) &= - \sum_{l=1}^n r'_l(\gamma) \prod_{k \neq l} (x - r_k(\gamma)) \end{aligned}$$

Evaluating the above at $x = r_j(\gamma)$,

$$\begin{aligned} \frac{\partial Q}{\partial \gamma}(r_j(\gamma), \gamma) &= -r'_j(\gamma) \prod_{k \neq j} (r_j(\gamma) - r_k(\gamma)) \\ \Rightarrow r'_j(\gamma) &= - \frac{\frac{\partial Q}{\partial \gamma}(r_j(\gamma), \gamma)}{\prod_{k \neq j} (r_j(\gamma) - r_k(\gamma))} \end{aligned}$$

This gives 1st order approximations for $r_j(\gamma)$ in the neighborhood of $\gamma=0$

$$\begin{aligned} r_j(\gamma) &\approx r_j(0) + r'_j(0)\gamma \\ &= r_j(0) - \frac{\frac{\partial Q}{\partial \gamma}(r_j(0), 0)}{\prod_{k \neq j} (r_j(0) - r_k(0))} \gamma \end{aligned}$$

ACKNOWLEDGEMENTS

We gratefully acknowledge the support of AFOSR award number FA9550-23-1-0046.

We would like to thank Dr. Brian M. Sadler, Senior Research Fellow, University of Texas at Austin, for his valuable insights and discussions on the motivation and potential applications of this work during the course of writing this paper.

REFERENCES

- [1] A. J. van der Schaft and B. M. Maschke, "Port-hamiltonian systems on graphs," *SIAM Journal on Control and Optimization*, vol. 51, no. 2, pp. 906–937, 2013. [Online]. Available: <https://doi.org/10.1137/110840091>
- [2] J. Chow and P. Kokotovic, "Time scale modeling of sparse dynamic networks," *IEEE Transactions on Automatic Control*, vol. 30, no. 8, pp. 714–722, 1985.
- [3] D. Romeres, F. Dörfler, and F. Bullo, "Novel results on slow coherency in consensus and power networks," in *2013 European Control Conference (ECC)*, 2013, pp. 742–747.
- [4] X. Cheng, Y. Kawano, and J. M. A. Scherpen, "Reduction of second-order network systems with structure preservation," *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 5026–5038, 2017.
- [5] I. Mirzaev and J. Gunawardena, "Laplacian dynamics on general graphs," *Bulletin of mathematical biology*, vol. 75, no. 11, pp. 2118–2149, 2013.
- [6] L. Pan, H. Shao, and M. Mesbahi, "Laplacian dynamics on signed networks," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, 2016, pp. 891–896.
- [7] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.
- [8] W. Ren, R. Beard, and E. Atkins, "A survey of consensus problems in multi-agent coordination," in *Proceedings of the 2005, American Control Conference, 2005.*, 2005, pp. 1859–1864 vol. 3.
- [9] F. Dörfler, J. W. Simpson-Porco, and F. Bullo, "Electrical networks and algebraic graph theory: Models, properties, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 977–1005, 2018.
- [10] S. V. Nagpal, G. G. Nair, F. Parise, and C. L. Anderson, "Designing robust networks of coupled phase oscillators with applications to the high voltage electric grid," *IEEE Transactions on Control of Network Systems*, vol. 10, no. 2, pp. 1046–1057, 2022.
- [11] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.
- [12] M. C. De Gennaro and A. Jadbabaie, "Decentralized control of connectivity for multi-agent systems," in *Proceedings of the 45th IEEE Conference on Decision and Control*, 2006, pp. 3628–3633.
- [13] L. Zhang, B. M. Sadler, R. S. Blum, and S. Bhattacharya, "Inter-cluster transmission control using graph modal barriers," *arXiv preprint arXiv:2010.04790*, Oct 2020, arXiv:2010.04790 [cs.RO].
- [14] C. Sun, R. Dai, and M. Mesbahi, "Weighted network design with cardinality constraints via alternating direction method of multipliers," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 4, pp. 2073–2084, 2018.
- [15] L. Meirovitch, *Fundamentals of vibrations*. Waveland Press, 2010.
- [16] C. Godsil, C. Royle, and G. Royle, *Algebraic Graph Theory*, ser. Graduate Texts in Mathematics. Springer, 2001. [Online]. Available: <https://books.google.com/books?id=gk60QgAACAAJ>
- [17] K. F. Riley, M. P. Hobson, and S. J. Bence, *Mathematical methods for physics and engineering: a comprehensive guide*. Cambridge university press, 2006.
- [18] E. B. Saff, *Fundamentals of Complex Analysis with Applications to Engineering*. Pearson, 2013.
- [19] A. M. Aly, "Proposed robust tuned mass damper for response mitigation in buildings exposed to multidirectional wind," *The Structural Design of Tall and Special Buildings*, vol. 23, no. 9, pp. 664–691, 2014. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/tal.1068>
- [20] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- [21] The MathWorks, *Optimization Toolbox User's Guide*, r2023b ed., The MathWorks, Inc., Natick, MA, USA, 2023. [Online]. Available: <https://www.mathworks.com/help/optim/>
- [22] B. Rozemberczki, R. Davies, R. Sarkar, and C. Sutton, "Gemsec: Graph embedding with self clustering," in *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2019*. ACM, 2019, pp. 65–72.
- [23] M. E. Muller, "A note on a method for generating points uniformly on n-dimensional spheres," *Communications of the ACM*, vol. 2, no. 4, pp. 19–20, 1959.
- [24] A. Sukharev, "Optimal strategies of the search for an extremum," *USSR Computational Mathematics and Mathematical Physics*, vol. 11, no. 4, pp. 119–137, 1971. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0041555371900085>
- [25] P. Deheuvels, "Strong bounds for multidimensional spacings," *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, vol. 64, no. 4, pp. 411–424, Dec 1983. [Online]. Available: <https://doi.org/10.1007/BF00534948>
- [26] A. Moitra, *Gaussian Mixture Models*. Cambridge University Press, 2018, p. 107–131.