Abstract—This paper aims to maximize algebraic connectivity of networks via topology design under the presence of constraints and an adversary. We are concerned with three problems. First, we formulate the concave-maximization topology design problem of adding edges to an initial graph, which introduces a nonconvex binary decision variable, in addition to subjugation to general convex constraints on the feasible edge set. Unlike previous approaches, our method is justifiably not greedy and is capable of accommodating these additional constraints. We also study a scenario in which a coordinator must selectively protect edges of the network from a chance of failure due to a physical disturbance or adversarial attack. The coordinator needs to strategically respond to the adversary’s action without presupposed knowledge of the adversary’s feasible attack actions. We propose three heuristic algorithms for the coordinator to accomplish the objective and identify worst-case preventive solutions. Each algorithm is shown to be effective in simulation and their compared performance is discussed.

I. INTRODUCTION

Motivation. Multi-agent systems are pervasive in new technology spaces such as power networks with distributed energy resources like solar and wind, mobile sensor networks, and large-scale distribution systems. In such systems, communication amongst agents is paramount to the propagation of information, which often lends itself to robustness and stability of the system. Network connectivity is well studied from a graph-theoretic standpoint, but the problem of designing topologies when confronted by engineering constraints or adversarial attacks is not well addressed by current works. We are motivated to study the NP-hard graph design problem of adding edges to an initial topology and to develop a method to solve it which has both improved performance and allows for direct application to the aforementioned constrained and adversarial settings.

Literature Review. The classic paper [7] by Miroslav Fiedler proposes a scalar metric for the algebraic connectivity of undirected graphs, which is given by the second-smallest eigenvalue of the graph Laplacian and is also referred to as the Fiedler eigenvalue. One of the main problems we are interested in studying is posed in [8], where the authors develop a heuristic for strategically adding edges to an initial topology to maximize this eigenvalue. Lower and upper bounds for the Fiedler eigenvalue with respect to adding a particular edge are found; however, the work is limited in that their approach is greedy and may not perform well in some cases. In addition, the proposed strategy does not address how to handle additional constraints that may be imposed on the network, such as limits on nodal degree or restricting costlier edges. The authors of [1] aim to solve the problem of maximizing connectivity for a particular robotic network scenario in the presence of an adversarial jammer, although the work does not sufficiently address scenarios with a more general adversary who may not be subject to dynamical constraints. The Fiedler eigenvector, which has a close relationship to the topology design problem, is studied in [11]. Many methods to compute this eigenvector exist, such as the cascadic method in [12]. However, these papers do not fully characterize how this eigenvector evolves from adding or removing edges from the network, which is largely unanswered by the literature. The authors of [6] study the spectra of randomized graph Laplacians, and [13] gives a means to estimate and maximize the Fiedler eigenvalue in a mobile sensor network setting. However, neither of these works consider the problem from a design perspective. In the celebrated paper by Goemans and Williamson [10], the authors develop a relaxation and performance guarantee on solving the MAXCUT problem, which has not yet been adapted for solving the topology design problem. Each of [5], [2] survey existing results related to the Fiedler eigenvalue and contain useful references.

Statement of Contributions. This paper considers three optimization problems and has two main contributions. First, we formulate the concave-maximization topology design problem from the perspective of adding edges to an initial network, subject to general convex constraints plus an intrinsic binary constraint. We then pose a scenario where a coordinator must strategically select links to protect from random failures due to a physical disturbance or malicious attack by a strategic adversary. In addition, we formulate this problem from the adversary’s perspective. Our first main contribution is a method to solve the topology design problem (and, by extension, the protected links problem). We develop a novel MAXCUT-inspired SDP relaxation to handle the binary constraint, which elegantly considers the whole problem in a manner where previous greedy methods fall short. Our next main contribution returns to the coordinator-adversary scenario. We first discuss the nonexistence of a Nash equilibrium in general. This motivates the development of an optimal preventive strategy in which the coordinator makes an optimal play with respect to any possible response by the adversary. We rigorously prove several auxiliary results about the solutions of the adversary’s computation-
ally hard concave-minimization problem in order to justify heuristic algorithms which may be used by the coordinator to search for the optimal preventive strategy. A desirable quality of these algorithms is they do not presuppose the knowledge of the adversary’s feasibility set, nor the capability of solving her problem. Rather, the latter two algorithms observe her plays over time and use these against her construct an effective preventive solution. Simulations demonstrate the effectiveness of our SDP relaxation for topology design and the performance of the preventive-solution seeking algorithms when applied to the adversarial link-protection problem.

II. Preliminaries

This section establishes some notation and preliminary concepts which will be drawn upon throughout the paper.

A. Notation

We denote by \( \mathbb{R} \) the set of real numbers. The notation \( x \in \mathbb{R}^n \) and \( A \in \mathbb{R}^{n \times m} \) indicates an \( n \)-dimensional real vector and an \( n \)-by-\( m \)-dimensional real matrix, respectively. The gradient of a real-valued multivariate function \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) with respect to a vector \( x \in \mathbb{R}^n \) is written \( \nabla_x f(x) \). The \( i \)-th component of a vector \( x \) is indicated by \( x_i \), all components of \( x \) not including the \( i \)-th component is indicated by \( x_{-i} \), and the \((i,j)\)-th element of a matrix \( A \) is indicated by \( A_{ij} \). The standard inner product is written \( \langle x, y \rangle = x^T y \) and the Euclidean norm of a vector \( x \) is denoted \( \|x\|_2 = \sqrt{x^T x} \). The closed Euclidean ball of radius \( \varepsilon \) centered at a point \( x \) is expressed as \( B_{\varepsilon}(x) \). Two vectors \( x \) and \( y \) are said to be parallel or equal if \( \langle x, y \rangle = 0 \), indicated by \( x \parallel y \), and the orthogonal complement to a span of vectors \( a_i \) is written \( \text{span}\{a_i\}^\perp \), meaning \( x \perp \forall y \in \text{span}\{a_i\} \). Elementwise multiplication is represented by \( x \circ y = (x_1 y_1, \ldots, x_n y_n)^T \). A symmetric matrix \( A \in \mathbb{R}^{n \times n} \) is said to be positive semidefinite if it is positive definite, written \( \text{sym}\{A_i\} \) if clarification is needed, with associated eigenvectors \( v_1, \ldots, v_n \) that are assumed to be of unit magnitude unless stated otherwise. A positive semi-definite matrix \( A \) is indicated by \( A \succeq 0 \).

We denote \( \mathbb{R}^n \) and \( 1_n \) refers to the \( n \)-dimensional vectors of all zeros and all ones, respectively. We use the notation \( \Pi_n := I_n - \frac{1_n 1_n^T}{n} \) and refer to \( \Pi_n \) as a pseudo-identity matrix; note that \( \text{null}(\Pi_n) = \text{span}\{1_n\} \). The operator \( \Pi_n \) for row arguments produces a diagonal matrix whose diagonal elements are the entries of the vector. The null space is denoted \( \emptyset \) and the cardinality of a finite set \( S \) is denoted \( |S| \). We express the \( m \)-Cartesian product of sets by means of a superscript \( m \), such as \( \overline{\mathcal{S}}_i^m \subset \mathbb{R}^{m\times n} \), where \( \overline{\mathcal{S}}_i, \mathcal{S}_i \in \mathbb{R} \). Probabilities and expectations are indicated by \( \mathbb{P} \) and \( \mathbb{E} \), respectively.

B. Graph Theory

We refer to [9] as a supplement for the concepts we describe throughout this subsection. In multi-agent engineering applications, it is useful to represent a network mathematically as a graph \( G = (\mathcal{N}, \mathcal{E}) \) whose node set is given by \( \mathcal{N} = \{1, \ldots, n\} \) and edge set \( \mathcal{E} \subset \mathcal{N} \times \mathcal{N} \), |\mathcal{E}| = m, which represents a physical connection or ability to transmit a message between agents. We refer to the set of nodes that node \( i \) is connected to as its set of neighbors, \( \mathcal{N}_i \subset \mathcal{N} \). We consider undirected graphs so \( (i, j) \in \mathcal{E} \) indicates \( j \in \mathcal{N}_i \) and \( i \in \mathcal{N}_j \). The graph has an associated Laplacian matrix \( L \in \mathbb{R}^{n \times n} \), whose elements are

\[
L_{ij} = \begin{cases} 
-1, & j \in \mathcal{N}_i, j \neq i, \\
|\mathcal{N}_i|, & j = i, \\
0, & \text{otherwise}.
\end{cases}
\]

Note that \( L \succeq 0 \). It is well known that the multiplicity of the zero eigenvalue is equal to the number of connected components in the graph [9]. To expand on this, connected graphs have a one-dimensional null space associated with the eigenvector \( 1_n \). The incidence matrix of \( L \) is given by \( E \in \{-1, 0, 1\}^{n \times n} \), where the \( i \)-th column of \( E \), given by \( e_i \in \{-1, 0, 1\}^n \), is associated with an edge \( l \sim (i, j) \). In particular, the \( i \)-th element of \( e_i \) is \(-1\), the \( j \)-th element is \(1\), and all other elements are zero. A vector \( x \in \{0, 1\}^n \) encodes the (dis)connectivity of the edges. In this sense, for \( l \in \{1, \ldots, m\} \), \( x_l = 0 \) indicates \( l \) is disconnected and \( x_l = 1 \) indicates \( l \) is connected. Then, \( L = E \text{diag}(x)E^\top \).

C. Set Theory

A limit point \( p \) of a set \( \mathcal{P} \) is a point such that any neighborhood \( B_{\varepsilon}(p) \) contains a point \( p' \in \mathcal{P} \). A set is closed if it contains all of its limit points, it is bounded if it is contained in a ball of finite radius, and it is compact if it is both closed and bounded. Let \( \mathcal{A}_i = \{ p \mid a_i^\top p \geq b_i \} \) be a closed half-space and \( \mathcal{P} = \mathcal{A}_1 \cap \cdots \cap \mathcal{A}_r \subset \mathbb{R}^m \) be a finite intersection of closed half-spaces. If \( \mathcal{P} \) is compact, we refer to it as a polytope. Consider a set of points \( \mathcal{F} = \{ p \in \mathcal{P} \mid a_i^\top p = b_i, i \in I \subseteq \{1, \ldots, r\} \} \). The affine hull of \( \mathcal{F} \) as \( \text{aff}(\mathcal{F}) = \{ p + w \mid p, w \in \mathbb{R}^m, p \in \mathcal{F}, w \perp \text{span}\{a_i\}_{i \in I} \} \) and define the relative interior of \( \mathcal{F} \) as \( \text{relint}(\mathcal{F}) = \{ p \mid \exists w > 0 : B_{\varepsilon}(p) \cap \text{aff}(\mathcal{F}) \subset \mathcal{F} \} \).

III. Problem Statements

This section formulates the three optimization problems that we study. The first problem aims to add edges to an initial topology to maximize algebraic connectivity of the final graph. The second problem introduces a coordinator who is charged with protecting some links in a network that are subject to an external disturbance or attack. The third problem takes the opposite approach of the latter problem by minimizing connectivity from an adversarial point of view.

A. Topology Design for Adding Edges

Consider a network of agents with some initial (possibly disconnected) graph topology characterized by an edge set \( \mathcal{E}_0 \) and Laplacian \( L_0 \). We would like to add \( k \) edges to \( \mathcal{E}_0 \), possibly subject to some additional convex constraints, so as to maximize the Fiedler eigenvalue of the final Laplacian \( L^* \). This problem is well motivated: the Fiedler eigenvalue
dictates convergence rate of many first-order distributed algorithms such as consensus and gradient descent. First, let \( \mathcal{E} \) be the complete edge set (not including redundant edges or self loops) with \( m = |\mathcal{E}| \). Consider the incidence matrix \( E \) associated with \( \mathcal{E} \) and the vector of edge connectivities \( x \), as described in Section IV. The constrained topology design problem is then formulated as

\[
\begin{align*}
\text{P1} : \max_{x,\alpha} & \quad \alpha, \\
\text{subject to} & \quad E(\text{diag}(x))E^\top \succeq \alpha \mathbb{I}_n, \\
& \quad \sum_{l=1}^m x_l \leq k + |\mathcal{E}_0|, \\
& \quad x \in \mathcal{X}, \\
& \quad x_l = 1, l \sim (i,j) \in \mathcal{E}_0, \\
& \quad x_l \in \{0,1\}, l \in \{1,\ldots,m\}. 
\end{align*}
\]

In P1, the solution \( \alpha^* \) is precisely the value for \( \lambda_2 \) of the final Laplacian solution given by \( L(x^*) = E(\text{diag}(x^*))E^\top \). This is encoded in the constraint (1b), where the pseudo-identity matrix \( \mathbb{I}_n \) has the effect of filtering out the fixed zero-eigenvalue of the Laplacian. A useful relation is \( \lambda_2 = \inf \{ z^\top L(x)z \mid z \perp \mathbb{I}_n, \|z\|_2 = 1 \} \), which shows that \( \lambda_2 \) as a function of \( x \) is a pointwise infimum of linear functions of \( x \) and is therefore concave. By extension, P1 is a concave-maximization problem in \( x \). The set \( \mathcal{X} \) is assumed compact and convex and may be chosen by the designer in accordance with problem constraints such as bandwidth/memory limitations, restrictions on nodal degrees, or restricting certain edges from being chosen. These constraints may manifest in applications such as communication bandwidth limitations amongst Distributed Energy Resource Providers for Real-Time Optimization in renewable energy dispatch [4]. The binary constraint (1c) is nonconvex and makes the problem NP-hard. Handling this constraint is one of the main objectives of this paper and will be addressed in Section IV.

As for existing methods of solving P1, one option is to solve it over the convex hull of the constraint set, which is given by \( [0,1]^m \cap \mathcal{X} \). Then, the problem may be solved in \( k \) steps by iteratively adding the edge \( l \sim (i,j) \) for which \( x_l \) is maximal in each step. This is discussed in [8] and the references therein. Although this method allows the designer to easily capture \( \mathcal{X} \), it is not a satisfying relaxation because the underlying characteristics of the connectivity are not well captured. The authors of [8] propose an alternate method which chooses the edge \( l \) for which \( \frac{\partial \lambda_2}{\partial x_l} = \frac{\partial L(x)}{\partial x_l}v_2 = v_2^\top e_l e_l^\top v_2 = (v_{2,i} - v_{2,j})^2 \) is maximal. This method is limited in that it is (a) greedy and (b) cannot account for \( \mathcal{X} \). We are motivated to develop a relaxation for P1 which improves on existing techniques in both performance and the capability of handling constraints.

**B. Topology Design for Protecting Edges**

We now formulate a problem which is closely related to P1 and interesting to study in its own right. Motivated by the scenario of guarding against disruptive physical disturbances or adversarial attacks, consider a coordinator who may protect up to \( k_s \) links from failing in a network. The failure of the links are assumed to be independent Bernoulli random variables whose probabilities are encoded by the vector \( p \in [0,1]^m \). Then, consider the coordinator’s decision vector \( s \in \mathcal{S} = \{0,1\} \cap \mathcal{S}' \), where \( \mathcal{S}' \) is assumed compact and convex. We write out the Laplacian as before, \( L(x) = E \text{diag}(x)E^\top \). Following a disturbance or attack, the probability that an edge \( l \) is (dis)connected is given by \( \mathbb{P}(x_l = 1) = (s_l - 1)p_l + 1 \) (resp. \( \mathbb{P}(x_l = 0) = (1 - s_l)p_l \)). The interpretation for the vector \( s \) here is that, if a particular element \( s_l = 1 \), it is deterministically connected and considered immune to the disturbance or attack. The coordinator’s problem is:

\[
\begin{align*}
\text{P2} : \max_{s,\alpha} & \quad \alpha, \\
\text{subject to} & \quad E(\text{diag}(x))E^\top \succeq \alpha \mathbb{I}_n, \\
& \quad \mathbb{P}(x_l = 1) = (s_l - 1)p_l + 1, \\
& \quad \mathbb{P}(x_l = 0) = (1 - s_l)p_l, \\
& \quad \sum_{l=1}^m s_l \leq k_s, \\
& \quad s \in \mathcal{S}', \quad s \in \{0,1\}^m. 
\end{align*}
\]

Due to the linearity of the expectation operator, (2c) is equivalent to \( E(\text{diag}((s - \mathbb{1}_m) \odot p + \mathbb{1}_m))E^\top \succeq \alpha \mathbb{I}_n \), which is an LMI (linear matrix inequality) in \( s \). We note that, as in P1, the objective of P2 may be thought of as a pointwise infimum of linear functions and, as such, is a binary concave-maximization problem in \( s \).

The formulation in P2 presupposes a fixed vector \( p \). However, it may be the case that a strategic attacker detects preventive action taken by the coordinator and adjusts her strategy to improve the likelihood of disconnecting the network. We now formulate the attacker’s problem for some known, fixed coordinator strategy \( s \):

\[
\begin{align*}
\text{P3} : \min_{p,\alpha} & \quad \alpha, \\
\text{subject to} & \quad \lambda_2(\mathbb{E}[E(\text{diag}(x))E^\top]) = \alpha, \\
& \quad \mathbb{P}(x_l = 1) = (s_l - 1)p_l + 1, \\
& \quad \mathbb{P}(x_l = 0) = (1 - s_l)p_l, \\
& \quad p \in \mathcal{P}, \quad p \in [0,1]^m. 
\end{align*}
\]

Notice here that the optimization is instead over \( p \) and is now a minimization of \( \alpha \). The constraint (3b) is now a nonlinear equality rather than an LMI, which manifests itself from this being a concave-minimization problem. This equality is not a convex constraint and will be addressed in Section IV. We assume \( \mathcal{P} \) is compact and convex.

**IV. An SDP Relaxation for Topology Design**

This section aims to develop a relaxed approach to solve P1 in a computationally efficient manner. Ideally, such an approach may also be straightforwardly extended to problems of the form P2. To do this, we draw on intuition from
the randomized hyperplane strategy given in [10] for solving the well-studied MAXCUT problem.

There are two notable differences between P1 and MAXCUT: the entries of the decision vector in P1 take values in \( \{0, 1\} \), whereas in MAXCUT, the decision (let’s say \( z \)) takes values \( z_i \in \{-1, 1\} \). The latter is convenient because it is equivalent to \( z_i^2 = 1 \). Additionally, the enumeration in MAXCUT is symmetric in the sense that, if \( z^* \) is a solution, then so is \(-z^*\). However, P1 is asymmetric in the sense that, if \( x^* \) is a solution, it cannot be said that \(-2x^* + 1_{m} \) (effectively swapping the zeros and ones in the elements of \( x^* \)) is a solution. We rectify these issues with a transformation and variable lift, respectively. Introduce a vector \( y = 2x - 1_m \) and notice \( x \in \{0, 1\}^m \) maps to \( y \in \{-1, 1\}^m \). Then, define \( \tilde{Y} = yy^\top \) so that \( y_1^2 = 1 \) may be enforced via \( Y_l = 1, l \in \{1, \ldots, m\} \). In addition, define \( \tilde{Y} = \begin{bmatrix} y_1 & y_2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} y_1^\top \\ \frac{1}{1} \\ \frac{1}{1} \end{bmatrix} \) to capture the asymmetry in the original variable \( x \). Now, we are ready to reformulate P1 as an SDP in the variable \( y \):

\[
\begin{align*}
P_4 : \max_{Y, y, \alpha} & \quad \alpha, \quad (4a) \\
\text{subject to} & \quad \frac{1}{2} E(\text{diag}(y) + I_m)E^\top \geq \alpha I_n, \quad (4b) \\
& \quad \tilde{Y} = \begin{bmatrix} y_1 & y_2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} y_1^\top \\ \frac{1}{1} \end{bmatrix} \geq 0, \quad (4c) \\
& \quad \text{rank}(\tilde{Y}) = 1, \quad (4d) \\
& \quad Y_{ll} = 1, l \in \{1, \ldots, m\}, \quad (4e) \\
& \quad y \in \mathcal{Y}, \quad (4f) \\
& \quad y_1 = 1, l \sim (i, j) \in \mathcal{E}_0, \quad (4g) \\
& \quad \frac{1}{2} \sum_i (y_i + 1) \leq k + |\mathcal{E}_0|, \quad (4h)
\end{align*}
\]

where \( \mathcal{Y} \) is an affine transformation on the set \( \mathcal{X} \) in [14], and we have simply used the transformation and variable lift to rewrite the other constraints. The problem \( P_4 \) is equivalent to P1: the NP-hardness now manifests itself in the nonlinear constraint (4h). Dropping this constraint produces a relaxed solution \( \tilde{Y}^* \) with the rank of \( \tilde{Y}^* \) not necessarily one.

This also produces a solution \( y^* \) which can be mapped back to \( x^* \). Of course, \( x^* \) may not take binary values due to the dropped rank constraint. We now briefly recall the geometric intuition to approximate the solution to MAXCUT in [10] with many technical details omitted here for brevity. Let \( Z^* \in \mathbb{R}^{m \times m} \) be a rank-2 solution to the rank-relaxed MAXCUT SDP problem. Decompose \( Z^* = W^TW \) with \( W \in \mathbb{R}^{r \times m} \), and notice the columns of \( W \) given by \( w_i \in \mathbb{R}^r, i \in \{1, \ldots, m\} \), are vectors on the \( r \)-dimensional unit ball due to \( Z^*_i = 1, i \in \{1, \ldots, m\} \). In order to determine a solution \( z^* \in \{-1, 1\}^m \), generate a uniformly random unit vector \( p \in \mathbb{R}^r \) which may define an hyperplane. If the vector \( w_i \) lies on one side of the hyperplane, i.e. \( \langle w_i, p \rangle \geq 0 \), set the corresponding element \( z_i = 1 \). If it is on the other side of the hyperplane, \( \langle w_i, p \rangle < 0 \), set \( z_i = -1 \). Geometrically speaking, the stronger a vector \( w_i \) is aligned with \( p \), the more “correlated” (for lack of a better term) node \( i \) is with the set \( \{j \in \{1, \ldots, m\} \mid z_j = 1\} \) and vice-versa. From another perspective, consider the case \( r_m = 1 \) which implies the solution is equivalent to the nonrelaxed problem. Then, it can be seen that \( \langle w_i, p \rangle \in \{-1, 1\} \) and the approach gives the exact optimizer for MAXCUT.

For our problem, decompose \( \tilde{Y}^* = U^TU \) with \( U \in \mathbb{R}^{r \times m+1} \), and obtain unit-vectors \( u_l \in \mathbb{R}^r, l \in \{1, \ldots, m+1\} \), from the columns of \( U \). Because of the asymmetry of our problem, we do not implement a random approach to determine the solution. Instead, notice that the last column \( u_{m+1} \) is qualitatively different than \( u_l, l \in \{1, \ldots, m\} \) due to the variable lift. We have that \( y_l = \langle u_l, u_{m+1} \rangle, l \in \{1, m\} \). Thus, larger entries of \( y_l \) correspond to vectors \( u_l \) on the unit ball which are more “aligned” with \( u_{m+1} \), which hearkens to the geometric intuition for the MAXCUT solution and \( u_{m+1} \) may be thought of as the vector \( p \) from MAXCUT. The entries \( y_l \) give a quantitative measure of the effectiveness of adding edge \( l \sim (i, j) \).

We suggest iteratively choosing the edge \( l \) associated with the largest element of \( y \) for which \( l \notin \mathcal{E}_0 \). If a particular edge is infeasible, this is elegantly accounted for by \[4i\] and is reflected in the relaxed solution to \( P_4 \). This approach may be iterated \( k \) times, updating \( \mathcal{E}_0 \) and decrementing \( k \) each time in accordance with \[4h\] to construct a satisfactory solution to the original NP-hard binary problem. In addition, this formulation is easily adaptable to solve \( P_2 \) via a similar transformation and variable lift in \( s \).

V. Protecting Links Against an Adversary

This section begins by studying the Nash equilibria of a game between the coordinator and attacker where they take turns solving \( P_2 \) and \( P_3 \). We first study the (non)existence of the Nash equilibria of this game, and use this result to motivate the development of a preventive strategy for the coordinator. We then provide some auxiliary results about the solutions of \( P_3 \) and use these to justify methods for finding such a preventive strategy.

A. Nash Equilibria

We begin by adopting the shorthand notation \( L(s, p) = \mathbb{E}\left[E(\text{diag}(x))E^\top\right] \) with \( x \) distributed as in \[2c, 2d\], i.e. \( L_{ij}(s, p) = \langle 1 - s_i \rangle p_{l-1}, l \sim (i, j) \in \mathcal{E}, i \neq j \) and \( L_{ii}(s, p) = -\sum_{j \in \mathcal{N}} L(s, p)_{ij} \). This matrix may be interpreted as a weighted Laplacian whose elements are given by the righthand side of \[2c, 2d\]. We also adopt the shorthand \( \alpha(s, p) = \inf_{z} \left( z^\top L(s, p)z \mid z \perp \perp 1_n, ||z||_2 = 1 \right) \) to refer to the Fiedler eigenvalue of \( L(s, p) \), and note that \( \alpha(s, p) \), as a pointwise infimum of bilinear functions, is concave-concave in \((s, p)\). From this, recall the first-order concavity relation \[3\]

\[
\alpha(s^2, p^2) \leq \alpha(s^1, p) + \nabla_s \alpha(s^1, p)^\top (s^2 - s^1), \forall s^1, s^2 \in \mathcal{S}, \nonumber
\]

\[
\alpha(s, p^2) \leq \alpha(s, p^1) + \nabla_p \alpha(s, p^1)^\top (p^2 - p^1), \forall p^1, p^2 \in \mathcal{P}. \nonumber
\]

To get a better grasp on this, we compute the gradient of \( \alpha \) with respect to both \( s \) and \( p \). Let \( \nabla \) be the Fiedler eigenvector
associated with the second-smallest eigenvalue (in this case, \(\alpha\)) of \(L(s, p)\). Then,
\[
\frac{\partial \alpha}{\partial s_l} = v^\top \frac{\partial L(s, p)}{\partial s_l} v, \quad \forall s_l \in S, \quad p_l = \frac{v_i - v_j}{v_i - v_j}^2, \tag{6}
\]
which is a straightforward extension of the computation shown near the end of Section 3A. Additionally,
\[
\frac{\partial \alpha}{\partial p_l} = \begin{cases} -(v_i - v_j)^2, & s_l = 0, \\ 0, & s_l = 1. \end{cases} \tag{7}
\]

The gradient with respect to \(s\) and \(p\) is a vector with elements given by (6)–(7), which are nonnegative for \(s\) and nonpositive for \(p\). Also, note that \(v \neq 1_n, v \neq 0\), implying the quantity \((v_i - v_j)^2\) must be strictly positive for some edges \(l \sim (i, j)\).

Now, consider a game where the coordinator and attacker take turns solving and implementing the solutions of \(\alpha\) associated with the second-smallest eigenvalue (in this case, Lemma 1 suggests that Nash equilibria are also unlikely to \(\exists s\) that \(s\) that
\[
\alpha(s^*, p^*) \leq \alpha(s^*, p^*), \quad \forall s \in S, \quad \forall p \in P, \tag{8}
\]
which is a stationary point of the aforementioned game. We now state a lemma to motivate the remainder of this section.

**Lemma 1.** *(Nonexistence of Nash Equilibrium).* A Nash equilibrium point \((s^*, p^*)\) satisfying (8) is not guaranteed to exist in general.

**Proof.** To show this result, we provide a simple counterexample. Consider a complete graph with \(n = 3\) nodes and \(m = 3\) edges, \(l \in \{1, 2, 3\} \sim \{1, 2, 1, 2, 3\} = E\), and the coordinator and attacker feasibility sets \(S = \{0, 1\}^3 \cap \{s \mid \sum s_l \leq 2\}, \quad P = \{p \mid \sum p_l \leq 1\}. \)

We consider three cases of the strategies \((s, p)\) to show the nonexistence of a Nash equilibrium here.

**Case 1:** The attacker play \(p^*\) is such that \(p_l = 1\) for \(l\) such that \(s_l = 1\). This point violates the right-hand side of (8). To see this, notice \(\exists \ell \neq l\) s.t. \(s_{\ell}^* = 0\) and that the play \(p^*\) with \(p_{\ell}^* = 1/2\) is optimal for \(P_3\).

**Case 2:** The attacker play \(p_l^*\) is such that \(p_l^* = 1\) for \(l\) such that \(s_l = 0\). This point violates the left-hand side of (8), and the coordinator recomputes an optimal play \(s^*\) with \(s_{1}^* = 1\).

**Case 3:** The attacker play \(p^*\) is such that \(\exists \ell \neq l\) with \(p_{\ell}^* = 1\). This point violates the right-hand side of (8) and the attacker recomputes an optimal play \(p^*\) with \(p_{l}^* = 1\) for the \(l\) corresponding to \(s_l^* = 0\).

The simple counterexample employed in the proof of Lemma 1 suggests that Nash equilibria are also unlikely to exist in more meaningful scenarios. This result should not come as a surprise: the solutions to \(P_2\) and \(P_3\) are in direct conflict with one another, and playing sequentially has the effect of the coordinator “chasing” the attacker around the network. We find that the cases for which we can construct a Nash equilibrium are trivial: for example, if \(1_m \in S\), the coordinator may choose \(s^* = 1_m\), and the attacker’s solution set is trivially the whole set \(P\) with the interpretation that the attacker is powerless to affect the value of \(\alpha\). Then, \((1_m, p)\) are Nash equilibria \(\forall p \in P\), and are not interesting.

**B. Coordinator’s Preventive Strategy**

Lemma 1 motivates the study of an optimal preventive strategy for the coordinator under the assumption that the attacker may always make a play in response to the coordinator’s action. Instead of a Nash equilibrium satisfying (8), we seek a point \((s^*, p^*)\) satisfying the following:
\[
(s^*, p^*) = \underset{s \in S}{\text{argmax}} \underset{p \in P}{\text{argmin}} \alpha(s, p). \tag{9}
\]

The interpretation of \(s^*\) solving (9) is that it provides the best-case solution for the coordinator given that the attacker makes the last play. In this sense, \(s^*\) is not optimal for \(p^*\); rather, it is an optimal play with respect to the whole set \(P\).

From the coordinator’s perspective, the objective function \(\alpha(s, p)\) is a pointwise infimum of concave functions of \(s\), and therefore the problem is a concave maximization. However, computing such a point may be dubious in practice, particularly since we have not assumed the coordinator has the capability of solving the concave minimization problem \(P_3\) or even knowledge of \(P\). It would, however, be convenient to use the attacker’s solutions to \(P_3\) against herself.

To do this, we establish some lemmas to gain insight on the solution sets of the attacker. This helps us construct heuristics for computing \(s^*\) in the sense of (9). We assume \(S, P \neq \emptyset\).

**Lemma 2.** *(Attacker’s Solution Tends to be Noninterior).* Consider the set of solutions \(P^* \subseteq P\) for some \(s\). If there exists point \(p^* \in P^*\) which is an interior point of \(P\), then \(P^* = P\).

**Proof.** Consider \(p^* \in P^*\) which is an interior point of \(P\). Pick \(\varepsilon > 0\) so \(\varepsilon L_r(s^*) \subseteq P\). From (5), the point \(p' = p^* - \varepsilon \nabla p_s(s^*, p^*)/\|\nabla p_s(s^*, p^*)\|_2 \in P\) violates the condition that \(p^*\) is a solution to \(P_3\) unless \(\nabla p_s(s^*, p^*) = 0_m\), so this must be the case. Then, in accordance with (5), all \(p \in P\) are solutions, and the statement \(P^* = P\) follows.

This lemma implies the solution set \(P^*\) consists of noninterior point(s) of \(P\) except in trivial cases. We now provide a stronger result in the case where \(P\) is a polytope, which shows that solutions tend to be contained in low-dimensional faces of \(P\) such as edges (line segments) and vertices.

**Lemma 3.** *(Attacker’s Solutions Tend Towards Low-Dimensional Faces).* Let \(P = A_1 \cap \cdots \cap A_r \subset \mathbb{R}^m\) be a compact polytope with half-spaces \(A_i\) characterized by \(a_i, b_i\) for \(i \in \{1, \ldots, r\}\), and let \(F\) be a face of \(P\) with \(a_j \perp p = b_j\) for \(j \in J \subseteq \{1, \ldots, r\}, \forall p \in F\). If a point \(p^* \in \text{relint}(F)\) is a solution to \(P_3\), then \(\nabla p_s(s^*, p^*) \in \text{span}\{a_j\}_{j \in J}\).

**Proof.** Suppose that \(\nabla p_s(s^*, p^*) \notin \text{span}\{a_j\}_{j \in J}\) and decompose \(\nabla p_s(s^*, p^*) = \beta \perp \zeta, \beta \in \text{span}\{a_j\}_{j \in J}, \zeta \in \text{span}\{a_j\}_{j \in J}, \zeta \perp 0_m\). Now consider the point \(p' = p^* - \varepsilon \zeta/\|\zeta\|_2 \in F\) for some \(\varepsilon > 0\). From (5), we have the relation \(\alpha(s, p') \leq \alpha(s^*, p^*) - \varepsilon \|\zeta\|_2\) with \(\varepsilon > 0\), and \(\|\zeta\|_2 > 0\), which contradicts \(p^*\) being a solution and completes the proof.

To interpret the result of Lemma 3 notice that \(p \in \text{relint}(F)\) implies \(p\) does not belong to a lower dimensional face, and that the dimension of \(\text{span}\{a_j\}_{j \in J}\) becomes.
large only as the dimension of $\mathcal{F}$ becomes small. This intuitively suggests that the gradient of $\alpha$ at $p^*$ may only belong to $\text{span}\{a_j\}_{j \in \mathcal{J}}$ for a $p \in \mathcal{F}$ if this span is large in dimension. This allows us to conservatively characterize the solution set $\mathcal{P}^*$, and the result gives credence to the notion that solutions take values in low-dimensional faces of $\mathcal{P}$.

It is our intent to use Lemmas 2 and 3 to establish intuition for the problem and justify solution strategies to the hard problem of computing a preventive $s^*$. Before proceeding, we establish one more simple lemma and provide discussion on deterministically connected graphs.

**Lemma 4. (Deterministic Connectivity).** Assume $\exists p \in \mathcal{P}$ with $p \succ 0_m$ and that the attacker makes the last play. Let $L(x)$ indicate the random matrix whose elements are distributed as in (3c)–(3d). Then, a coordinator’s strategy $s$ gives $\lambda_2 > 0$ of $L(x)$ with probability 1 if and only if the elements of $s$ equal to 1 are associated with edges of a connected graph.

**Proof.** $\Rightarrow$ We do not need to assume the attacker has made an optimal play with respect to $P$ 3. Instead, consider any play $p \succ 0_m$ and note that each edge $l$ associated with $s_l = 0$ has a nonzero probability of being disconnected. Then, there is a nonzero probability that $x_l = 0$ for each $l$ with $s_l = 0$, and the remaining protected edges do not form a connected graph. Then, if the elements of $s$ equal to one are not associated edges defining a connected graph, $\lambda_2 = 0$ is an event with nonzero probability.

$\Leftarrow$ This direction is trivial: $x_l = 1$ with probability 1 for edges $l$ corresponding to $s_l = 1$. If those edges form a connected graph, then $\lambda_2 > 0$ with probability 1. $\square$

The consequence of Lemma 4 is obvious: if the coordinator does not have the resources to protect edges which form a connected graph and the attacker targets all edges, then there is no guarantee the resulting graph will be connected. A subject of future work is to provide some insight on the lower bound of $\lambda_2$ for particular cases of $\mathcal{S}$ and $\mathcal{P}$.

**C. Heuristics for Computing a Preventive Strategy**

Recall from the previous subsection that the goal is to compute $s^*$ as a solution to (3). In this subsection, we describe three approaches to computing a satisfactory solution and formally adopt the following assumptions.

**Assumption 1. (Coordinator’s Problem is Solvable).** Given a known vector $p$, the coordinator can find the optimal solution of $P$ 2.

**Assumption 2. (Attacker Plays Optimally and Last).** The attacker’s play $p$ belongs to a convex, compact set $\mathcal{P}$. In addition, she always makes optimal plays which solve $P$ 3 given the coordinator’s play $s$, and she may always play in response to the coordinator changing his decision.

**Assumption 3. (Available Information).** The coordinator is cognizant of Assumption 2 and has access to the current attacker play $p$. He may compute $\alpha(s, p)$ for a particular play $(s, p)$.

**Assumption 4. (Only Last Play Matters).** The objective $\alpha(s, p)$ (and, by extension, $\lambda_2$) is only consequential once both the coordinator and attacker have chosen their final strategy and do not make additional plays.

We now construct three heuristics for computing $s^*$, the latter two of which observe the attacker plays $p(t)$ over a time horizon $t \in \{1, \ldots, T\}$ and iteratively construct a solution.

**Algorithm 1 Random Sampling**

1: procedure RAND$(\mathcal{S}, \mathcal{E}, T)$
2: Compute $(v_i - v_j)^2$ for each $l \sim (i, j) \in \mathcal{E}$
3: for $t = 1, \ldots, T$ do
4: $\mathcal{M} \leftarrow \{1, \ldots, m\}$
5: $s(t) \leftarrow 0_m$
6: done $\leftarrow false$
7: while done $= false$ do
8: Choose $l \in \mathcal{M}$ uniformly randomly
9: if $s(t) \leftarrow 1$ $\Rightarrow s(t) \in \mathcal{S}$ then
10: $s(t) \leftarrow 1$
11: end if
12: $\mathcal{M} \leftarrow \mathcal{M} \setminus \{l\}$
13: if $\exists l \in \mathcal{M}$ s.t. $s(t) \leftarrow 1$ $\Rightarrow s(t) \in \mathcal{S}$ then
14: done $\leftarrow true$
15: end if
16: end while
17: Play $s(t)$
18: Store $s(t)$ and $\alpha(s(t), p(t))$
19: end for
20: $t^* \leftarrow \text{argmax}\, \alpha(s(t), p(t))$
21: return $s^* = s(t^*)$
22: end procedure

Algorithm 1 is simple and doesn’t utilize the attacker’s plays. For each $t$, it constructs $s(t)$ by picking edges uniformly randomly. Each loop terminates when no feasible edges remain. The value of $\alpha$ is recorded following the attacker’s response $p(t)$, and the best $s(t)$ is returned.

Algorithm 2 utilizes a convex weighting function $\eta$ such that $\sum_{k=1}^{T} \eta(k) = 1$. We suggest three possible choices for $\eta$:

- $\eta_1(k) = 1/t,$
- $\eta_2(k) = \frac{\gamma^{(t-k)}}{\sum_{k} \gamma^{(t-k)}}, \gamma \in (0, 1),$
- $\eta_3(k) = \frac{\alpha(s(k), p(k))}{\sum_{k} \alpha(s(k), p(k))}.$

These may be interpreted as a uniform weighting of each observation $p(t)$, a recency-biased weighting, and a penalty-biased weighting, respectively. Algorithm 2 is motivated by a few observations. Firstly, recall Lemmas 2 and 3 and note that $p(t)$ may jump around extreme points of $\mathcal{P}$ as $s(t)$ evolves. Successive convex combinations of the solutions $p(t)$ effectively push the coordinator’s decision towards responding to the vulnerable parts of the space over time.

Algorithm 3 adopts the following stronger version of Assumption 1.
Algorithm 2 Convex Combinations

Require: Convex weighting function η with \( \sum_{k=1}^{t} η(k) = 1 \)
1: procedure CvX(S, E, T)
2: \( s(1) \leftarrow 0 \)
3: Store \( s(1), p(1) \) and \( α(s(1), p(1)) \)
4: \( p_0(1) \leftarrow p(1) \)
5: for \( t = 2, \ldots, T \) do
6: \( s(t) \leftarrow \arg\max_{s \in S} α(s, p_0(t - 1)) \)
7: Store \( s(t), p(t) \), and \( α(s(t), p(t)) \)
8: for \( k = 1, \ldots, t \) do
9: \( θ(k) \leftarrow η(k) \)
10: end for
11: \( p_0(t) \leftarrow \sum_{k=1}^{t} θ(k)p(k) \)
12: end for
13: \( t^* \leftarrow \arg\max_{t} α(s(t), p(t)) \)
14: return \( s^* = s(t^*) \)
15: end procedure

Algorithm 3 Constructing \( P \) via Pointwise Search

1: procedure Search(S, E, T)
2: \( s(1) \leftarrow 0 \)
3: Store \( s(1), p(1) \) and \( α(s(1), p(1)) \)
4: \( \overline{P} \leftarrow p(1) \)
5: for \( t = 2, \ldots, T \) do
6: \( s(t) \leftarrow \arg\min_{s \in S, p \in \overline{P}} α(s, p) \)
7: Store \( s(t), p(t) \) and \( α(s(t), p(t)) \)
8: \( \overline{P} \leftarrow \overline{P} \cup \{p(t)\} \)
9: end for
10: \( t^* \leftarrow \arg\max_{t} α(s(t), p(t)) \)
11: return \( s^* = s(t^*) \)
12: end procedure

Assumption 5. (Coordinator’s Problem is Solvable Over Finitely-Many Points). Given a finite set of points \( \mathcal{P} \subset \mathcal{P} \), the coordinator may compute the solution \( s^* = \arg\max_{s \in S} \min_{p \in \mathcal{P}} α(s, p) \).

VI. SIMULATIONS

We now examine our proposed SDP relaxation for solving P 1. For the ease of comparison with the Fiedler vector heuristic given in [8], we do not include any additional convex constraints beyond \( || \mathcal{E}_0 || = 28 \). For a network with 14 nodes, \( |\mathcal{E}_0| = 28 \) initial edges generated randomly, we implement the Fiedler method, our SDP method, and the approach of taking the convex hull of the feasibility set of P 1. We compute 100 trials with different topology initializations for each of the \( k \in \{25, 40\} \) edge-addition cases. Our SDP design outperforms the Fiedler vector heuristic in 75 trials for the \( k = 25 \) case, 80 trials for the \( k = 40 \) case, and it outperforms the convex hull approach in all 100 trials of both cases. This improved performance is observed over a variety of network sizes and initial connectivities, and we observe that increasing \( k \) increases the likelihood that our method outperforms the alternatives.

One such instance of \( k = 25 \) added edges is plotted in Figure 1 and the performance is plotted in Figure 2. Firstly, note that both our SDP method and the Fiedler vector method greatly outperform the simple convex hull approach. Our SDP method is outperformed by the Fiedler method in early iterations. In later iterations, the performance of our method catches up with and surpasses the Fiedler vector heuristic (this is common behavior across other initializations). We contend that the reason for this is that the relaxed solution of P 4 at each iteration is cognizant of the entire problem horizon, as opposed to the Fiedler vector heuristic which greedily chooses edges in accordance with the direction of steepest ascent in \( λ_2 \).

Next, we study a small network of 7 nodes and 11 edges so that the solutions to P 2 and P 3 may be brute-forcibly computed to test Algorithms 1-3 with \( η_i(k) \) being used for Algorithm 2. We choose \( P = [0.25, 0.75]^{11} \cap \{p | \sum_{i=1}^{11} p_i \leq 4.25\} \) and \( S = \{0, 1\}^{11} \cap \{s | \sum_{i=1}^{11} s_i \leq 5\} \). We run the algorithms for \( T = 30 \) iterations and plot the results at each iteration in Figure 3.

Clearly, Algorithm 1 does not improve across iterations as it does not utilize information about the attacker’s plays from previous iterations, and it achieves a maximum value of \( α(s(t^*), p(t^*)) = 0.8175 \), which is a bit below the global optimum \( α(s^*, p^*) = 0.8762 \). Algorithm 2 achieves the global optimum \( α(s(t^*), p(t^*)) = 0.8762 \) the fastest, at \( t^* = 5 \), although it never reaches this point again and instead oscillates around suboptimal points, indicating that optimality may not be reliably attained in general. Counter-intuitively, the performance of Algorithm 3 does not improve...
monotonically in $t$, although this should be expected: early solutions $s(t)$ may get “lucky”, in some sense, but the subsequent iteration may allow the attacker to jump to a new vulnerable part of the space. Once the algorithm achieves the global optimum at $t^* = 9$, it does not dip below this for the remainder of the time horizon. Finally, we note that the behavior of each algorithm observed here is typical when implemented on other small graphs.

VII. CONCLUDING REMARKS

This paper introduced three related problems motivated by studying the algebraic connectivity of a graph by adding edges to an initial topology or protecting edges under the case of a disturbance or attack on the network. We developed a novel SDP relaxation to address the NP-hardness of the design and demonstrated in simulation that it is superior to existing methods which are greedy and cannot accommodate general constraints. In addition, we studied the dynamics of the game that may be played between a network coordinator and strategic attacker. We developed the notion of an optimal preventive solution for the coordinator and proposed effective heuristics to find such a solution guided by characterizations of the solutions to the attacker’s problem. Future work includes characterizing the performance of our SDP relaxation and developing an algorithm which provably converges to the optimal preventive strategy.

REFERENCES

[1] S. Bhattacharya, A. Gupta, and T. Basar. Jamming in mobile networks: a game-theoretic approach. Numerical Algebra, Optimization, and Control, 3(1):1–30, 2013.
[2] S. Boyd. Convex optimization of graph Laplacian eigenvalues. In Proc. Int. Congress of Mathematicians, volume 3, pages 1311–1319, 2006.
[3] S. Boyd and L. Vandenberghe. Convex Optimization. Cambridge University Press, 2004.
[4] CAISO Business Practice Manual for Market Operation, 2017. Version 51. Available at https://bpmcm.caiso.com/Pages/BPMDetails.aspx?BPM=Market%20Operations
[5] N. M. M. de Abreu. Old and new results on algebraic connectivity of graphs. Linear Algebra and Its Applications, 423(1):53–73, 2006.
[6] X. Ding and T. Jiang. Spectral distributions of adjacency and Laplacian matrices of random graphs. The Annals of Applied Probability, 20(6):2086–2117, 2010.
[7] M. Fiedler. Algebraic connectivity of graphs. Czechoslovak Mathematical Journal, 23(98):298–305, 1973.
[8] A. Ghosh and S. Boyd. Growing well-connected graphs. In IEEE Int. Conf. on Decision and Control, pages 6605–6611, San Diego, USA, 2006.
[9] C. D. Godsil and G. F. Royle. Algebraic Graph Theory, volume 207 of Graduate Texts in Mathematics. Springer, New York, 2001.
[10] M. Goemans and D. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. Journal of the Association for Computing Machinery, 42(6):1115–1145, 1995.
[11] R. Merris. Laplacian graph eigenvectors. Linear Algebra and Its Applications, 278:221–236, 1998.
[12] J. Urschel, X. Hu, J. Xu, and L. Zikatanov. A cascadic multigrid algorithm for computing the Fiedler vector of graph Laplacians. Journal of Computational Mathematics, 33(2):209–226, 2015.
[13] P. Yang, R. A. Freeman, G. J. Gordon, K. M. Lynch, S. S. Srinivasa, and R. Sukthankar. Decentralized estimation and control of graph connectivity for mobile sensor networks. Automatica, 46(2):390–396, 2010.