2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

Maximizing Algebraic Connectivity of Constrained Graphs in Adversarial Environments

Tor Anderson Chin-Yao Chang Sonia Martínez

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 nonconvex 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 [6] 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 [7], 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 [10]. Many methods to compute this eigenvector exist, such as the cascadic method in [11]. 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 [5] 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 [9], 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 [4], [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

Tor Anderson, Chin-Yao Chang, and Sonia Martínez are with the Department of Mechanical and Aerospace Engineering, University of California, San Diego, CA, USA. Email: {tka001, chc433, soniamd}@eng.ucsd.edu. This document has been modified to reduce size; a full version can be found at https://arxiv.org/abs/1711.04091. This research was supported by the Advanced Research Projects Agency - Energy under the NODES program, Cooperative Agreement DE-AR0000695.

results about the solutions of the adversary's computationally 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 function $f : \mathbb{R}^n \to \mathbb{R}$ with respect to $x \in \mathbb{R}^n$ is written $\nabla_x f(x)$. The i^{th} component of a vector x is indicated by x_i and the $(i,j)^{\text{th}}$ element of a matrix A is indicated by A_{ij}. The standard inner product is written $\langle x, y\rangle = x^\top y$ and the Euclidean norm of a vector x is denoted $\|x\|_2 = \sqrt{\langle x, x\rangle}$. The closed Euclidean ball of radius ε centered at a point x is expressed as $\mathcal{B}_\varepsilon(x)$. Two vectors x and y are perpendicular if $\langle x, y\rangle = 0$, indicated by $x \perp y$, and the orthogonal complement to a span of vectors a_i is written $\operatorname{span}\{a_i\}^\perp$, meaning $x \perp y, \forall x \in \operatorname{span}\{a_i\}, \forall y \in \operatorname{span}\{a_i\}^\perp$. Elementwise multiplication is represented by $x \diamond y = (x_1y_1, \dots, x_ny_n)^\top$. A symmetric matrix $A \in \mathbb{R}^{n\times n}$ has n real eigenvalues ordered as $\lambda_1 \le \dots \le \lambda_n$ with associated eigenvectors $v_1, \dots, v_n$ that are assumed unit magnitude. A positive semi-definite matrix A is indicated by $A \succeq 0$. We denote componentwise inequality as $x \succeq y$. The notation $\mathbf{0}_n$ and $\mathbf{1}_n$ refers to the n-dimensional vectors of all zeros and all ones, respectively. We use the notation $\mathbb{I}_n := I_n - \dfrac{\mathbf{1}_n\mathbf{1}_n^\top}{n}$ and refer to this matrix as a *pseudo-identity matrix*; note that $\operatorname{null}(\mathbb{I}_n) = \operatorname{span}\{\mathbf{1}_n\}$. The operator diag for vector arguments produces a diagonal matrix whose diagonal elements are the entries of the vector. The cardinality of a finite set $\mathcal{S}$ is denoted $|\mathcal{S}|$. We express the m Cartesian product of sets by means of a superscript m, such as $[\underline{s}_i, \overline{s}_i]^m \subset \mathbb{R}^m$, where $\underline{s}_i, \overline{s}_i \in \mathbb{R}$. Probabilities and expectations are indicated by $\mathbb{P}$ and $\mathbb{E}$, respectively.

B. Graph Theory

We refer to [8] as a supplement for this subsection. In multi-agent engineering applications, it is useful to represent a network as a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ of nodes $\mathcal{N} = \{1, \dots, n\}$ and edges $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$, $|\,\mathcal{E}\,| = m$, which represent a physical connection or ability to transmit a message between agents. We refer to the nodes that node i is connected to as its neighbor set, $\mathcal{N}_i \subset \mathcal{N}$. 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 a Laplacian matrix $L \in \mathbb{R}^{n\times n}$ whose elements $L_{ij} = -1$ for $j \in \mathcal{N}_i, j \neq i$; $L_{ii} = |\,\mathcal{N}_i\,|$, and $L_{ij} = 0$ otherwise.

Note that $L \succeq 0$. The multiplicity of the zero eigenvalue is equal to the number of connected components in the graph, and *connected* graphs have a one-dimensional null space associated with the eigenvector $\mathbf{1}_n$ [8]. The incidence matrix of L is given by $E \in \{-1, 0, 1\}^{n\times m}$, where the l^{th} column of E, given by $e_l \in \{-1, 0, 1\}^n$ is associated with an edge $l \sim (i,j)$. The i^{th} element of e_l is -1, the j^{th} element is 1, and all other elements are zero. A vector $x \in \{0,1\}^m$ encodes the (dis)connectivity of the edges. In this sense, for $l \in \{1, \dots, m\}$, $x_l = 0$ indicates l is disconnected and $x_l = 1$ indicates l is connected. Then, $L = E \operatorname{diag}(x) E^\top$.

C. Set Theory

A limit point p of a set $\mathcal{P}$ is a point such that any neighborhood $\mathcal{B}_\varepsilon(p)$ 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 $\mathcal{A}_i = \{p \mid a_i^\top p \ge b_i\}$ be a closed half-space and $\mathcal{P} = \mathcal{A}_1 \cap \dots \cap \mathcal{A}_r \subset \mathbb{R}^m$ 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 $\mathcal{F} = \{p \in \mathcal{P} \mid a_i^\top p = b_i, i \in \mathcal{I} \subseteq \{1, \dots, r\}; a_j^\top p \ge b_j, j \in \{1, \dots, r\}\backslash\mathcal{I}\}$ and let $h = \dim(\operatorname{span}\{a_i\})$ be the dimension of the subspace spanned by $\{a_i\}_{i\in\mathcal{I}}$. Then, we refer to $\mathcal{F}$ as an $(m-h)$-dimensional face of $\mathcal{P}$. Lastly, denote the affine hull of $\mathcal{F}$ as $\operatorname{aff}(\mathcal{F}) = \{p + w \mid p, w \in \mathbb{R}^m, p \in \mathcal{F}, w \perp \operatorname{span}\{a_i\}_{i\in\mathcal{I}}\}$ and define the relative interior of $\mathcal{F}$ as $\operatorname{relint}(\mathcal{F}) = \{p \mid \exists\varepsilon > 0 : \mathcal{B}_\varepsilon(p) \cap \operatorname{aff}(\mathcal{F}) \subset \mathcal{F}\}$.

III. Problem Statements

This section formulates the three optimization problems that we study which are related to adding edges, protecting edges, and attacking edges of a graph, respectively.

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 add k edges to $\mathcal{E}_0$ so as to maximize the Fiedler eigenvalue of the resulting Laplacian $L^\star$. This problem is well motivated: the Fiedler eigenvalue dictates convergence rate of many first-order distributed algorithms such as consensus and gradient descent. Let $\mathcal{E}$ be the complete edge set with $m = |\,\mathcal{E}\,|$. Consider the incidence matrix E associated with $\mathcal{E}$ and the vector of edge connectivities x, as in Section II-B. The topology design problem is:

$$
\begin{aligned}
\mathrm{P\,1}: \max_{x,\alpha} \quad & \alpha, && \text{(1a)}\\
\text{subject to} \quad & E(\operatorname{diag}(x))E^\top \succeq \alpha\,\mathbb{I}_n, && \text{(1b)}\\
& \sum_{l=1}^{m} x_l \le k + |\,\mathcal{E}_0\,|, && \text{(1c)}\\
& x \in \mathcal{X}, && \text{(1d)}\\
& x_l = 1,\ l \sim (i,j) \in \mathcal{E}_0, && \text{(1e)}\\
& x_l \in \{0,1\},\ l \in \{1, \dots, m\}. && \text{(1f)}
\end{aligned}
$$

The solution $\alpha^\star$ is precisely the value for λ_2 of the Laplacian $L(x^\star) = E(\mathrm{diag}(x^\star))E^\top$. 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 $\lambda_2 = \inf_z\{z^\top L(x)z \mid z \perp \mathbf{1}_n, \|z\|_2 = 1\}$, which shows that λ_2 as a function of x is a pointwise infimum of linear functions of x and is therefore concave. By extension, P 1 is a concave-maximization problem in x. The constraint (1c) captures the notion of adding k edges to the initial topology $\mathcal{E}_0$. 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 [**?**]. In (1e), the initial topology is translated to a constraint on x. The binary constraint (1f) is nonconvex; handling this constraint is one of the main objectives of this paper and is addressed in Section IV.

As for existing methods of solving P 1, the authors of [7] propose an alternate method which chooses the edge l for which $\dfrac{\partial \lambda_2}{\partial x_l} = v_2^\top \dfrac{\partial L(x)}{\partial x_l} v_2 = v_2^\top e_l e_l^\top v_2 = (v_{2,i} - v_{2,j})^2$ 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 P 1 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 P 1 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 modelled as independent Bernoulli random variables with probabilities encoded by the vector $p \in [0,1]^m$. Then, consider the coordinator's decision vector $s \in \mathcal{S} = \{0,1\}^m \cap \mathcal{S}'$, where $\mathcal{S}'$ is assumed compact and convex. Following a disturbance or attack, the probability that an edge l is (dis)connected is given by $\mathbb{P}(x_l \equiv 1) = (s_l - 1)p_l + 1$ (resp. $\mathbb{P}(x_l \equiv 0) = (1 - s_l)p_l$). If a particular element $s_l = 1$, it is deterministically connected and considered *immune* to the disturbance or attack. The coordinator's problem is:

$$
\begin{aligned}
\mathsf{P}\,2: \max_{s,\alpha} \quad & \alpha, && \text{(2a)}\\
\text{subject to} \quad & \mathbb{E}\left[E(\mathrm{diag}(x))E^\top\right] \succeq \alpha\,\mathbb{I}_n, && \text{(2b)}\\
& \mathbb{P}(x_l \equiv 1) = (s_l - 1)p_l + 1, && \text{(2c)}\\
& \mathbb{P}(x_l \equiv 0) = (1 - s_l)p_l, && \text{(2d)}\\
& \sum_{l=1}^{m} s_l \le k_s, && \text{(2e)}\\
& s \in \mathcal{S},' \quad s \in \{0,1\}^m. && \text{(2f)}
\end{aligned}
$$

Due to linearity of the expectation, (2b) is equivalent to $E(\mathrm{diag}((s - \mathbf{1}_m) \diamond p + \mathbf{1}_m))E^\top \succeq \alpha\,\mathbb{I}_n$, which is an LMI (linear matrix inequality) in s. Note that P 2 is a binary concave-maximization problem in s, similarly to P 1.

The formulation in P 2 presupposes a fixed vector p. From a strategic attacker perspective, they may solve a problem P 3 in response to the strategy s. In the interest of space, we briefly describe how P 3 differs from P 2: s is now fixed and instead p becomes the decision variable; additionally, (2a) becomes a minimization, (2b) becomes a nonlinear equality instead of an LMI, and (2e)–(2f) become $p \in \mathcal{P} \subseteq [0,1]^m$. The nonlinear equality constraint manifests itself from P 3 being a concave-*minimization* problem. 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 P 1 and, by extension, P 2, in a manner similar to the well-studied MAXCUT problem [9].

There are two notable differences between P 1 and MAXCUT: the entries of the decision vector in P 1 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 $z_i^2 = 1$, and the enumeration in MAXCUT is *symmetric* in the sense that, if $z^\star$ is a solution, then so is $-z^\star$. However, P 1 is *assymmetric* in the sense that, if $x^\star$ is a solution, it *cannot* be said that $-2x^\star + \mathbf{1}_m$ (effectively swapping the zeros and ones in the elements of $x^\star$) is a solution. We rectify these issues with a transformation and variable lift, respectively. Introduce a vector $y = 2x - \mathbf{1}_m$ and notice $x \in \{0,1\}^m$ maps to $y \in \{-1,1\}^m$. Then, define $Y = yy^\top$ so that $y_l^2 = 1$ may be enforced via $Y_{ll} = 1$, $l \in \{1,\ldots,m\}$. In addition, define $\widetilde{Y} = \begin{bmatrix} y \\ 1 \end{bmatrix}\begin{bmatrix} y \\ 1 \end{bmatrix}^\top = \begin{bmatrix} Y & y \\ y^\top & 1 \end{bmatrix}$ to capture the asymmetry in the original variable x. Now, we are ready to reformulate P 1 as an SDP in the variable y:

$$
\begin{aligned}
\mathsf{P}\,4: \max_{Y,y,\alpha} \quad & \alpha, && \text{(3a)}\\
\text{subject to} \quad & \frac{1}{2}E(\mathrm{diag}(y) + I_m)E^\top \succeq \alpha\,\mathbb{I}_n, && \text{(3b)}\\
& \widetilde{Y} = \begin{bmatrix} Y & y \\ y^\top & 1 \end{bmatrix} \succeq 0, && \text{(3c)}\\
& \mathrm{rank}\,(\widetilde{Y}) = 1, && \text{(3d)}\\
& \widetilde{Y}_{ll} = 1,\ l \in \{1,\ldots,m\}, && \text{(3e)}\\
& y \in \mathcal{Y}, && \text{(3f)}\\
& y_l = 1,\ l \sim (i,j) \in \mathcal{E}_0, && \text{(3g)}\\
& \frac{1}{2}\sum_i (y_i + 1) \le k + |\,\mathcal{E}_0\,|, && \text{(3h)}
\end{aligned}
$$

where $\mathcal{Y}$ is an affine transformation on the set $\mathcal{X}$ in (1d), and we have simply used the transformation and variable lift to rewrite the other constraints. The problem P 4 is equivalent to P 1: the NP-hardness now manifests itself in the nonlinear constraint (3d). Dropping this constraint produces a *relaxed* solution $\widetilde{Y}^\star$ with the rank of $\widetilde{Y}^\star$ not necessarily one.

This also produces a solution $y^\star$ which can be mapped back to $x^\star$. Of course, $x^\star$ may not take binary values due to the dropped rank constraint. Recall the intuition of the random hyperplane approach to computing a solution to MAXCUT [9]. For our problem, decompose $\widetilde{Y}^\star = U^\top U$ 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 $y_l = \langle u_l, u_{m+1}\rangle, l \in \{1,\ldots,m\}$. 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 are a 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 (3f) 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 (3h) to construct a satisfactory solution to the original binary problem. This formulation is easily adaptable to solve P 2 via a transformation in s. We note that the computational complexity of solving semidefinite programs of the form P 4 is on the order of $O(m^{1/2})$ [12], so a conservative estimate of the complexity for k added edges is roughly $O(m^{1/2}k)$.

V. Protecting Links Against an Adversary

This section studies a game between the coordinator and attacker where they take turns solving P 2 and P 3. We study Nash equilibria and solutions of this game to develop methods for computing a *preventive* strategy for the coordinator.

A. Nash Equilibria

We begin by adopting the shorthand notation $L(s,p) = \mathbb{E}\left[E(\mathrm{diag}(x))E^\top\right]$ with x distributed as in (2c)–(2d), i.e. $L_{ij}(s,p) = (1-s_l)p_l - 1, l \sim (i,j) \in \mathcal{E}, i \neq j$ and $L_{ii}(s,p) = -\sum_{j\in\mathcal{N}_i} L_{ij}(s,p)$. This matrix may be interpreted as a weighted Laplacian whose elements are given by the righthand side of (2c). We also adopt the shorthand $\alpha(s,p) = \inf_z\{z^\top L(s,p)z \mid z \perp \mathbf{1}_n, \|z\|_2 = 1\}$ 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). Recall the first-order concavity relation [3]

$$\begin{aligned}\alpha(s^2,p) &\leq \alpha(s^1,p) + \nabla_s\alpha(s^1,p)^\top(s^2-s^1),\ \forall s^1,s^2\in\mathcal{S},\\ \alpha(s,p^2) &\leq \alpha(s,p^1) + \nabla_p\alpha(s,p^1)^\top(p^2-p^1),\ \forall p^1,p^2\in\mathcal{P}.\end{aligned} \tag{4}$$

To compute the gradient of α with respect to s or p, let v be the Fiedler eigenvector associated with the second-smallest eigenvalue (in this case, α) of $L(s,p)$. Then,

$$\frac{\partial\alpha}{\partial s_l} = v^\top\frac{\partial L(s,p)}{\partial s_l}v, = v^\top p_l e_l e_l^\top v = p_l(v_i - v_j)^2, \tag{5}$$

which is a straightforward extension of the computation shown near the end of Section III-A. Additionally,

$$\frac{\partial\alpha}{\partial p_l} = \begin{cases}-(v_i-v_j)^2, & s_l = 0\\ 0, & s_l = 1.\end{cases} \tag{6}$$

The gradient with respect to s and p is a vector with elements given by (5)–(6), which are nonnegative for s and nonpositive for p. Also, note that $v \neq \mathbf{1}_n$, $v \neq 0$, implying the quantity $(v_i - v_j)^2$ must be strictly positive for some edges $l \sim (i,j)$.

Consider a game where the coordinator and attacker take turns implementing the solutions of P 2 and P 3, respectively. A Nash equilibrium is a point $(s^\star, p^\star)$ with the property

$$\alpha(s,p^\star) \leq \alpha(s^\star,p^\star) \leq \alpha(s^\star,p), \forall s\in\mathcal{S}, \forall p\in\mathcal{P}, \tag{7}$$

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 $(s^\star, p^\star)$ satisfying (7) is not guaranteed to exist in general.

Proof. Proofs can be found at `https:///arxiv.org/abs/1711.04091` and are henceforth omitted for brevity. □

It is easy to construct counterexamples which demonstrate this, suggesting that Nash equilibria are unlikely to exist in meaningful scenarios. We find that the cases for which we can construct a Nash equilibrium are trivial: for example, if $\mathbf{1}_m \in \mathcal{S}$, the coordinator may choose $s^\star = \mathbf{1}_m$, and the attacker's solution set is trivially the whole set $\mathcal{P}$, i.e. the attacker is powerless to affect the value of α. Then, $(\mathbf{1}_m, p)$ are Nash equilibria for any 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 (7), we seek a point $(s^\star, p^\star)$ satisfying the following:

$$(s^\star, p^\star) = \operatorname*{argmax}_{s\in\mathcal{S}}\ \operatorname*{argmin}_{p\in\mathcal{P}}\ \alpha(s,p). \tag{8}$$

The interpretation of $s^\star$ solving (8) is that it provides the best-case solution for the coordinator given that the attacker makes the last play. In this sense, $s^\star$ is not optimal for $p^\star$; rather, it is an optimal play with respect to the whole set $\mathcal{P}$.

From the coordinator's perspective, the objective function $\operatorname*{argmin}_{p\in\mathcal{P}}\ \alpha(s,p)$ is a pointwise infimum of concave functions of s, and therefore the problem is a concave maximization. However, we have not assumed the coordinator knowledge of $\mathcal{P}$ or the ability to solve P 3. It would instead 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 $s^\star$ in the sense of (8).

Lemma 2. ***(Attacker's Solution Tends to be Noninterior).*** Consider the set of solutions $\mathcal{P}^\star \subseteq \mathcal{P}$ to P 3 for some s. If there exists point $p^\star \in \mathcal{P}^\star$ which is an interior point of $\mathcal{P}$, then $\mathcal{P}^\star = \mathcal{P}$.

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

Lemma 3. ***(Attacker's Solutions Tend Towards Low-Dimensional Faces).*** Let $\mathcal{P} = \mathcal{A}_1 \cap \cdots \cap \mathcal{A}_r \subset \mathbb{R}^m$ be a compact polytope with half-spaces $\mathcal{A}_i$ characterized by a_i, b_i for $i \in \{1, \ldots, r\}$, and let $\mathcal{F}$ be a face of $\mathcal{P}$ with $a_j^\top p = b_j$ for $j \in \mathcal{J} \subseteq \{1, \ldots, r\}$, $\forall p \in \mathcal{F}$. If a point $p^\star \in \operatorname{relint}(\mathcal{F})$ is a solution to $\mathsf{P}\,3$, then $\nabla_p \alpha(s, p^\star) \in \operatorname{span}\{a_j\}_{j \in \mathcal{J}}$.

To interpret the result of Lemma 3, notice that $p \in \operatorname{relint}(\mathcal{F})$ implies p does not belong to a lower dimensional face, and that the the dimension of $\operatorname{span}\{a_j\}_{j\in\mathcal{J}}$ becomes large only as the dimension of $\mathcal{F}$ becomes small. This intuitively suggests that the gradient of α at $p^\star$ may only belong to $\operatorname{span}\{a_j\}_{j\in\mathcal{J}}$ for a $p \in \mathcal{F}$ if this span is large in dimension. This allows us to conservatively characterize the solution set $\mathcal{P}^\star$, 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 $s^\star$.

C. Heuristics for Computing a Preventive Strategy

Recall from the previous subsection that the goal is to compute $s^\star$ as a solution to (8). 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 $\mathsf{P}\,2$.

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 $\mathsf{P}\,3$ 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 and can compute $\alpha(s, p)$.

Assumption 4. ***(Only Last Play Matters).*** The objective $\alpha(s, p)$ (and, by extension, λ_2) is only consequential once both the coordinator and attacker have chosen their final strategy and do not make additional plays.

We now describe three heuristics for computing $s^\star$, 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.

For brevity, we omit the formal statements of each Algorithm and briefly describe them instead. In Algorithm 1, for each $t \in \{1, \ldots, T\}$, start with $s(t) = \mathbf{0}_m$. Choose a uniformly random edge l which may feasibly be protected and update $s(t)$ accordingly. Iterate until no feasible edges remain. For each play $s(t), t \in \{1, \ldots, T\}$, observe $\alpha(s(t), p(t))$ and return the $s(t^\star)$ for which this quantity is maximized. This algorithm is simple to implement, although it does not utilize information about the attacker's plays.

Algorithm 2 makes an arbitrary initial play $s(1)$. The attacker's response $p(1)$ is recorded, and a new play $s(2)$ solving $\mathsf{P}\,2$ for $p = p(1)$ is implemented. From here on, following the attacker's response $p(t)$, the coordinator's strategy $s(t+1)$ is computed as the solution to $\mathsf{P}\,2$ with p being a convex combination of all previously observed $p(k), k \in \{1, \ldots, t\}$. When the algorithm is terminated at time T, return $s(t^\star)$ where $t^\star = \underset{t\in\{1,\ldots,T\}}{\operatorname{argmax}}\ \alpha(s(t), p(t))$. Recall Lemmas 2–3 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 Assumption 5, a stronger version of Assumption 1.

Assumption 5. ***(Coordinator's Problem is Solvable Over Finitely-Many Points).*** Given a finite set of points $\overline{\mathcal{P}} \subset \mathcal{P}$, the coordinator may compute the solution $s^\star = \underset{s\in\mathcal{S}}{\operatorname{argmax}}\ \underset{p\in\overline{\mathcal{P}}}{\min}\ \alpha(s, p)$.

It operates similarly to Algorithm 2, but instead of solving $\mathsf{P}\,2$ with respect to a convex combination of previous plays, it computes the solution $s(t)$ as the optimal play with respect to all previous attacker plays $\overline{\mathcal{P}} = \bigcup_{k=1}^{t-1} p(k)$. The best observed play $s(t^\star)$ is returned as in Algorithm 1–2. This is more computationally demanding than Algorithms 1 and 2, but it is strongly rooted in the theoretical understanding of the problem we have developed in the following sense: the convex hull of these points $\operatorname{co}(\overline{\mathcal{P}})$ at time t is a compact polytope whose vertices are defined by the points $p(k)$. Applying Lemma 3, it stands to reason that points in the interior or in higher-dimensional faces of $\operatorname{co}(\overline{\mathcal{P}})$ are uncommon solutions. We expect $\operatorname{co}(\overline{\mathcal{P}})$ to grow in each loop of the algorithm and effectively reconstruct the attacker's feasibility set $\mathcal{P}$.

We state the following trivial lemma for completeness.

Lemma 4. ***(Nondecreasing Performance of Algorithms 1–3).*** Let $p^\star(T)$ solve $\mathsf{P}\,3$ for $s = s^\star(T)$, where $s^\star(T)$ is the returned strategy of Algorithm 1, 2, or 3 truncated at time T. For all $T > 1$, $\alpha(s^\star(T), p^\star(T)) \geq \alpha(s^\star(T-1), p^\star(T-1))$.

VI. Simulations

We now examine our proposed SDP relaxation for solving $\mathsf{P}\,1$. For the ease of comparison with the Fiedler vector heuristic given in [7], we do not include any additional convex constraints beyond (1c). For a network with 14 nodes, $|\,\mathcal{E}_0\,| = 28$ initial edges generated randomly, we implement our SDP method, the Fiedler vector method [7], and a naive approach of solving $\mathsf{P}\,1$ over the convex hull of its feasibility set and iteratively adding edges with maximal values of x_l. 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. The performance of one such instance for $k = 25$ added edges is plotted in Figure 1, where it can be seen that our method outperforms the Fiedler vector heuristic in later iterations (this is common behavior across other initializations), which can be attributed to the non-greedy nature of our method.

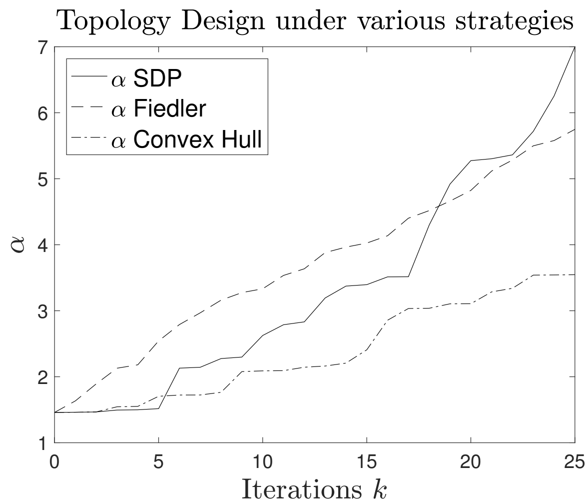

Fig. 1. Performance each method over $k = 25$ iterations.

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 a uniform convex weighting $1/t$ being used for Algorithm 2. We choose $\mathcal{P} = [0.25, 0.75]^{11} \cap \{p \mid \sum_l p_l \leq 4.25\}$ and $\mathcal{S} = \{0, 1\}^{11} \cap \{s \mid \sum_l s_l \leq 5\}$. We run the algorithms for $T = 30$ iterations and plot the results at each iteration in Figure 2.

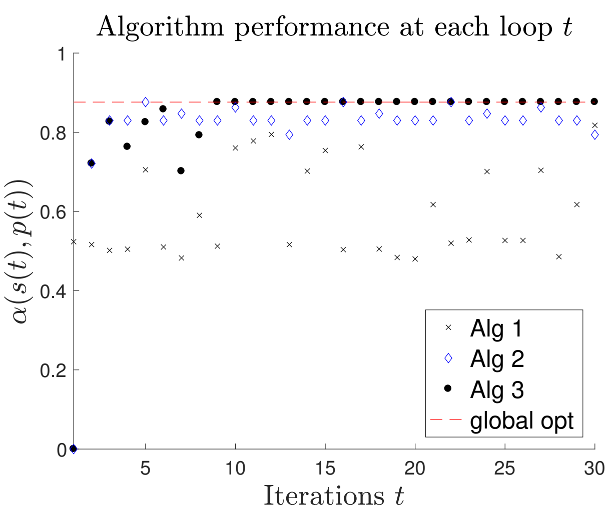

Fig. 2. Performance of Algorithms 1–3 at each loop t.

Algorithm 1 does not improve across iterations as it does not utilize information from previous iterations. It achieves a maximum value of $\alpha(s(t^\star), p(t^\star)) = 0.8175$. Algorithm 2 achieves the global optimum $\alpha(s(t^\star), p(t^\star)) = 0.8762$ the fastest, at $t^\star = 5$, although it never reaches this point again and instead oscillates around suboptimal points, indicating that optimality may not be reliably attained in general. The performance of Algorithm 3 does not improve monotonically in t due to nonconvexities, but once the algorithm achieves the global optimum at $t^\star = 9$ it remains optimal. 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 nonconvexity 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] N. de Abreu. Old and new results on algebraic connectivity of graphs. *Linear Algebra and Its Applications*, 423:53–73, 2006.

[5] X. Ding and T. Jiang. Old and new results on algebraic connectivity of graphs. *The Annals of Applied Probability*, 20(6):2086–2117, 2010.

[6] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(98):298–305, 1973.

[7] A. Ghosh and S. Boyd. Growing well-connected graphs. In *IEEE Int. Conf. on Decision and Control*, pages 6605–6611, San Diego, USA, 2006.

[8] C. D. Godsil and G. F. Royle. *Algebraic Graph Theory*, volume 207 of *Graduate Texts in Mathematics*. Springer, New York, 2001.

[9] 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.

[10] R. Merris. Laplacian graph eigenvectors. *Linear Algebra and Its Applications*, 278(1–3):221–236, 1998.

[11] J. Urschel, J. Xu, X. Hu, and L. Zikatanov. A cascadic multigrid algorithm for computing the fiedler vector of graph laplacians. *Journal of Computational Mathematics*, 33(2):209–226, 2015.

[12] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

[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.