Inducing Equilibria in Networked Public Goods Games through Network Structure Modification

David Kempe
University of Southern California
david.m.kempe@gmail.com

Sixie Yu
Washington University in St. Louis
sixie.yu@wustl.edu

Yevgeniy Vorobeychik
Washington University in St. Louis
yvorobeychik@wustl.edu

ABSTRACT

Networked public goods games model scenarios in which self-interested agents decide whether or how much to invest in an action that benefits not only themselves, but also their network neighbors. Examples include vaccination, security investment, and crime reporting. While every agent's utility is increasing in their neighbors' joint investment, the specific form can vary widely depending on the scenario. A principal, such as a policymaker, may wish to induce large investment from the agents. Besides direct incentives, an important lever here is the network structure itself: by adding and removing edges, for example, through community meetings, the principal can change the nature of the utility functions, resulting in different, and perhaps socially preferable, equilibrium outcomes. We initiate an algorithmic study of targeted network modifications with the goal of inducing equilibria of a particular form. We study this question for a variety of equilibrium forms (induce all agents to invest, at least a given set S, exactly a given set S, at least k agents), and for a variety of utility functions. While we show that the problem is NP-complete for a number of these scenarios, we exhibit a broad array of scenarios in which the problem can be solved in polynomial time by non-trivial reductions to (minimum-cost) matching problems.

1 INTRODUCTION

Groups of individuals often encounter the following type of scenario. Each member of the group can decide whether or how much effort (or money) to invest for the common good; everyone in the group (including the individual) profits from all members' efforts, but the individual incurs a cost for the investment. Examples of such scenarios include decisions whether or not to vaccinate, report crime in a neighborhood, invest in security, chip in on department committee work, keep one's yard representable or sidewalk shoveled, or purchase a tool that one's friends or neighbors can share.

These and many other scenarios are modeled by *public goods games* [18, 22]. In many applications, including most of the ones listed above, the benefits of an individual's effort are not reaped by all group members, but only by those with whom the individual interacts. This naturally motivates the definition of *networked public*

goods games [3, 4, 9], in which a (given, known) network captures which individuals will benefit from which other individuals' efforts.

In networked public goods games, an agent's utility depends on (1) her[1] own investment decision, which incurs some cost (in terms of money, time, effort, or risk), and (2) the aggregate investment by the agent and her neighbors in the network. (Precise definitions of all concepts are given in Section 2.) While the agent's utility is always non-decreasing in the neighbors' joint investment, the specific functional form can vary widely. For example, it may suffice to have a single friend with a useful tool, but lowering crime rates in a neighborhood may require broad participation in crime reporting.

For practically all public goods scenarios, the equilibria involve significant underinvestment. That is, significantly fewer agents expend effort than would lead to a socially optimal outcome, a phenomenon closely related to the so-called "Tragedy of the Commons" and the "Bystander Effect." This kind of underinvestment is not only predicted by theory, but typically observed in practice as well. When the equilibrium outcomes are socially undesirable, a principal, such as a policymaker, may be interested in changing the parameters of the game so as to induce equilibrium outcomes that are better aligned with the social interest. A natural and traditional approach is to change the cost structure, by rewarding investment (decreasing investment costs) or punishing failure to invest (increasing the cost of non-investment). For many of the scenarios listed above, monetary fines, other types of penalties, or social pressure implement such rewards or punishments. Generally, the design of cost or reward structures and rules of encounter is at the heart of work in mechanism and market design [21].

In graphical games, such as the networked public goods game, however, there is an additional important parameter that may be subject to modification: the network structure itself. For example, to facilitate crime prevention, the principal may organize community meetings in particular neighborhoods, increasing the density of the social network among the community members. The principal may also add individual links, for example, by introducing individuals to one another, or weaken relationships (remove links) by adding hurdles to specific interactions (e.g., if the public goods game represents strategic interactions in a criminal organization, and law enforcement chooses which relationships to monitor).

We initiate an algorithmic study of targeted network structure modifications in networked public goods games with binary actions, with the goal of inducing pure strategy Nash equilibria (PSNE) with desirable properties. We will consider a principal who is aiming for "high investment." Each edge has a cost for addition/removal, and to induce a desirable equilibrium of the game, the principal can

[1]For clarity, we will always refer to agents using female pronouns, and the principal using male pronouns.

add/remove edges from the network, subject to an upper bound on the total cost.

Naturally, there are different concrete ways of capturing the goal of "high investment," and we consider the following four natural candidate properties which the target equilibrium should satisfy: (1) all agents invest, (2) exactly a given subset S of agents invest, (3) at least the agents in a given subset S invest, and (4) at least r agents invest. We study each of these objectives for games with a variety of utility functions, ranging from general monotone functions to generalized sigmoid functions, as well as convex and concave functions. While we show (in Section 3) that the problem is NP-complete for several of these scenarios, we exhibit (in Section 4) a broad array of settings in which it can be solved in polynomial time by non-trivial reductions to minimum-cost matching problems. The full summary of our main results is presented in Table 1.

	all	$= S$	$\supseteq S$	$\geq r$
general	hard	hard	hard	hard
sigmoid	poly	poly	hard	hard
convex	poly	poly	hard	hard
concave	poly	poly	hard	hard

Table 1: An overview of our results.

For all entries of the table marked "poly," our algorithms handle fully general edge cost structures. On the other hand, all of the hardness proofs apply already in very restrictive cases, allowing only one of addition and removal, and either allowing unlimited changes of the allowed type, or simply restricting the total *number* of added/removed edges.

Related Work: Our work is conceptually connected to three broad threads in the literature: graphical games, mechanism and market design, and network design. Graphical models of games capture various forms of structure in the players' utility functions which limit the scope of utility dependence on other players' actions [24]. An important class of these are *graphical games*, where a player's utility only depends on the actions of her network neighbors [16]. Networked public goods games are one important example of graphical games, with utilities only depending on the investment choices by a player's network neighbors [3, 10, 17]. Bramoullé and Kranton [3] studied the effects of network structure modification on a public goods game. Our results, however, are novel in several respects. First, they assumed that only a single edge is added to the underlying network and studied how the addition affects welfare. In contrast, we consider addition and deletion of sets of edges, and focus on the algorithmic aspects of the problem. Moreover, Bramoullé and Kranton only consider strictly concave utility functions, whereas we study convex, concave, and general sigmoid utility functions (detailed definitions are in Section 2). Galeotti et al. [10] also considered the effects of modifying the underlying network of a public goods game on equilibrium behavior and welfare. However, their analysis is restricted to convex or concave utilities which are degree symmetric (i.e., if two nodes have the same degree, they must have the same utility function), involves incomplete information of players about the network, and does not consider the associated algorithmic problem. In contrast, we

focus on algorithmic questions and allow heterogeneous utilities. Yu et al. [29] studied algorithmic aspects of binary public goods games They showed that in general, checking the existence of a pure-strategy Nash equilibrim is NP-complete, and also identified tractable cases based on restrictions of either the utility functions or the underlying network structure. They proposed a heuristic to compute approximate equilibria in general games. However, they did not consider modifying the network structure to induce certain Nash equilibria. Grossklags et al. [13] studied how economic agents invest in security through the lens of public goods games. The value of the public goods is the overall protection level. Each agent has two options: investing in self-protection, or investing in self-insurance. The former affects the overall protection level, as well as the loss incurred by the agent, while the latter only affects the agent's own loss. They analyzed the Nash equilibria under five economic settings, which characterize different threat models.

Mechanism and market design (e.g., [21]) also aim to change the parameters of a game to induce equilibrium outcomes favored by a principal. However, the specific ways in which the game's parameters are changed are vastly different; key approaches include the design of market structure, such as matching market mechanisms [14], payments, as in traditional mechanism design [21], or the structure of information available to the players [7].

Anoter relevant line of research is network design. An interesting subclass is network design with fair cost-sharing [2]. Network design has also been applied to model security applications. Hota and Sundaram [15] focused on designing an optimal network topology that minimizes the expected fraction of attacked nodes, where the probability of a node being attacked is a function of its security investment as well as the investments by its neighbors.

The idea of altering a (social) network in order to induce certain outcomes is present in a number of recent works, for a variety of different outcomes. Sheldon et al. [23] aim to modify the network so as to maximize the spread of cascades, while [6, 12, 27] aim to alter the spectral gap of the network to make it more or less connected. Along similar lines, Bredereck and Elkind [5] considered the converging state of simple information diffusion dynamics, with a specific focus on how the removal of edges can be used to manipulate the majority opinion in such outcomes. Similar ideas arise in a recent line of work (e.g., [19, 25] and the references therein) studying how the outcome of an election can be manipulated by altering network structures. In a sense, the converse problem is studied by Amelkin and Singh [1], who aim to reduce opinion control by recommending (i.e., adding) links to social network users. Similar ideas are present in the work of Garimella et al. [11], who aim to decrease opinion polarization by connecting pairs of individuals with differing opinions. Sless et al. [26] investigated the problem of coalition formation through adding links to the underlying social network. All of these works share the high-level goal of inducing (socially) preferable equilibrium behavior, but the specific optimization goals, and with them the algorithmic approaches, are vastly different.

An analysis of connections between equilibrium outcomes of games and network structure was carried out by Bramoullé et al. [4]. They found that the smallest eigenvalue $\lambda_{\min}$ of the network's adjacency matrix is a key quantity for equilibria; recall that spectral properties also play a role in several of the other related papers

discussed above. Bramoullé et al. also considered the effects of network structure modification; in particular, they investigated how the addition of edges affects $\lambda_{\min}$, which in turn provides a qualitative understanding of the effects of edge additions on equilibria. However, they did not investigate the algorithmic issues involved. Milchtaich [20] studied equilibrium existence as a function of network topology in weighted network congestion games.

2 MODEL

2.1 Binary Networked Public Goods Game

A *binary networked public goods (BNPG)* game is characterized by the following:

(1) A simple, undirected and loop-free graph $G = (V, E)$ whose nodes $V = \{1, 2, \ldots, n\}$ are the agents/players, and whose edges $E = \{(i, j) \mid i, j \in V\}$ represent the interdependencies among the players' payoffs.

(2) A binary strategy space $\{0, 1\}$ for each player i. Choosing strategy 1 corresponds to *investing* in a public good, while choosing 0 captures non-investment. We use x_i to denote the action chosen by player i, and $\mathbf{x} = (x_1, x_2, \ldots, x_m)$ for the joint pure strategy profile of all players.

(3) For each player i, a non-decreasing utility function $U_i(\mathbf{x})$.

As is standard in the literature on networked public goods [3], we assume that each player's utility depends only on (1) her own investment (for which she incurs a cost), and (2) the joint investment of herself and her neighbors in the network, which provides her with a positive externality. Formally, we capture this as follows. Let $\mathcal{N}_i^{(G)} = \{j \mid (j, i) \in E\}$ be the set of neighbors of i in the graph G; then, we can define $n_i^{(G, \mathbf{x})} = \sum_{j \in \mathcal{N}_i^{(G)}} x_j$ to be the number of i's neighbors who invest under $\mathbf{x}$. When $G, \mathbf{x}$ are clear from the context, we will omit them from this notation. We assume that each player i's utility function is of the following form:

$$U_i(\mathbf{x}) = U_i(x_i, n_i^{(\mathbf{x})}) = g_i(x_i + n_i^{(\mathbf{x})}) - c_i x_i. \tag{1}$$

The second term $(-c_i x_i)$ simply captures the cost that i incurs from investing herself. Each g_i is a non-negative and non-decreasing function (a standard assumption in the public goods games literature), capturing the positive externality that i experiences from her neighbors' (and her own) investments. In many scenarios, g_i will have additional properties, such as being concave or convex, and we discuss such properties in Section 2.3. Observe that each function g_i can be represented using $O(n)$ values, so the entire BNPG game (including the graph structure) can be represented using $O(n^2)$ values.

We are interested in inducing particular pure strategy Nash Equilibria of the game by modifying the network structure. Pure strategy Nash Equilibria are defined as follows:

Definition 2.1. In a BNPG game, a *pure strategy Nash Equilibrium (PSNE)* is an action profile $\mathbf{x} \in \{0, 1\}^n$ satisfying $U_i(x_i, n_i^{(\mathbf{x})}) > U_i(1 - x_i, n_i^{(\mathbf{x})})$, or $U_i(x_i, n_i^{(\mathbf{x})}) = U_i(1 - x_i, n_i^{(\mathbf{x})})$ and $x_i = 1$, for every player i. Thus, we are assuming that each player in equilibrium always breaks ties in favor of investing.

A given BNPG game may have multiple equilibria. We will be interested in modifying the graph G to ensure that at least one element of a given set $\mathcal{X}$ is a PSNE. For notational convenience, we interpret $\mathcal{X}$ both as a set of strategy vectors $\mathbf{x} \in \{0, 1\}^n$ and as the subset of investing players $S(\mathbf{x}) := \{i \mid x_i = 1\}$, whichever is notationally more convenient. We are interested in the following classes of PSNE:

all: Every player invests, i.e., $\mathcal{X} = \{\{1, 2, \ldots, n\}\}$.

$= S$: Exactly a given set S of players invests (and the other players do not), i.e., $\mathcal{X} = \{S\}$. All players investing is the special case $S = \{1, \ldots, n\}$.

$\supseteq S$: *At least* the set S of players invests; other players may also invest. Here, $\mathcal{X} = \{T \mid T \supseteq S\}$.

$\geq r$: At least r players invest. Here, $\mathcal{X} = \{T \mid |T| \geq r\}$.

In general, even without the ability to modify G, deciding if a BNPG has an equilibrium in $\mathcal{X}$ is NP-hard. This can be seen most directly with the following example (see also Section 3.2 of [3]): Each cost is $c_i = 1$, and each $g_i(z) = 2$ if $z \geq 1$, and $g_i(0) = 0$. Then, the PSNE are exactly the strategy profiles $\mathbf{x}$ in which independent sets of G invest. Therefore, if $\mathcal{X}$ is the set of all profiles in which at least r players invest (for given r), the problem of deciding if the game has a PSNE in $\mathcal{X}$ is equivalent to the INDEPENDENT SET problem.

2.2 Network Modifications

The main modeling contribution of our work is to assume that a principal can modify the network G (subject to a budget) with the goal of inducing equilibria from a class $\mathcal{X}$. Formally, an input graph $G' = (V, E')$ on the agents is given. Each node pair (i, j) has an associated cost $\gamma_{(i,j)} = \gamma_{(j,i)} \geq 0$. When $(i, j) \in E'$, this is the cost for removing the edge (i, j) from G', while for $(i, j) \notin E'$, it is the cost for adding the edge (i, j) to G'. When the principal produces a graph $G = (V, E)$, the cost of doing so is $\sum_{e \in E \triangle E'} \gamma_e = \sum_{e \in E \setminus E'} \gamma_e + \sum_{e \in E' \setminus E} \gamma_e$. The principal is given a budget B not to be exceeded. The goal is to solve the following problem:

Definition 2.2 (Network Design for BNPG). Given a BNPG instance, edge costs γ_e, desired PSNE class $\mathcal{X}$, and budget B, find an edge set E with $\sum_{e \in E \triangle E'} \gamma_e \leq B$ such that the BNPG game on (V, E) has at least one PSNE in $\mathcal{X}$.

The general costs $\gamma_{(i,j)}$ admit many natural special cases: by setting $\gamma_{(i,j)} = \infty$ for $(i, j) \in E'$ (or for $(i, j) \notin E'$), we can prohibit the removal (or addition) of edges. By setting $\gamma_{(i,j)} = 0$, we can allow unlimited removal (or addition) of edges. And by setting $\gamma_{(i,j)} = 1$, we can simply restrict the number of edges removed/added.

2.3 Utility Functions and Induced Degrees

In the fully general version of the model, the g_i can be arbitrary non-decreasing functions. We will show that at this level of generality, the Network Design problem is NP-hard for all four classes of PSNE we consider (Theorem 3.2). In most scenarios, g_i will have additional properties. Among the most common of these are:

Concavity When g_i is concave, the returns for additional investments of neighbors are diminishing. The incentive structures in binary *best-shot games* [10] can be captured by concave g_i.

Convexity When g_i is convex, the returns for additional investments of neighbors are increasing.

Sigmoid For many natural scenarios, such as the adoption of innovations [30], the g_i are neither concave nor convex on their entire domain. Instead, g_i begins convex, with increasing returns to more investors, but eventually reaches saturation and diminishing returns.

We call such a function g_i a *(generalized) sigmoid function*[2] if there exists some $\hat{z}$ such that $g_i(z)$ is convex on $\{z \mid z \leq \hat{z}\}$ and concave on $\{z \mid z \geq \hat{z}\}$. Note that sigmoid functions subsume both concave functions (with $\hat{z} = -\infty$) and convex functions (with $\hat{z} = \infty$).

The first useful observation is that we can capture all the relevant information about an agent i's utility function using the set of numbers of investing neighbors that would make i invest. We call such sets *investment degree sets*, and denote them by D_i. When g_i is convex/concave/sigmoid, the investment degree sets have particularly nice forms, captured by the following lemma:

LEMMA 2.3. *For every non-decreasing function $g_i : [0, n-1] \to \mathbb{R}_+$ and cost c_i, there exists a unique set $D_i \subseteq \{0, 1, \ldots, n-1\}$ such that $x_i = 1$ is a best response to n_i if and only if $n_i \in D_i$. Furthermore,*

(1) *When g_i is concave, D_i is a downward-closed interval.*
(2) *When g_i is convex, D_i is an upward-closed interval.*
(3) *When g_i is sigmoid, D_i is an interval.*

Conversely, for every set $D_i \subseteq \{0, 1, \ldots, n-1\}$, there exists a non-decreasing function $g_i : [0, n-1] \to \mathbb{R}_+$ and cost c_i such that $x_i = 1$ is a best response to n_i if and only if $n_i \in D_i$. Furthermore,

(1) *When D_i is a downward-closed interval, there exists such a g_i which is concave.*
(2) *When D_i is an upward-closed interval, there exists such a g_i which is convex.*
(3) *When D_i is an interval, there exists such a g_i which is sigmoid.*

PROOF. We write $\Delta g_i(z) = g_i(z + 1) - g_i(z)$ for the discrete derivative. By definition of best responses (recall our tie breaking rule), a player i invests if and only if $U_i(1, n_i) \geq U_i(0, n_i)$, which is equivalent to

$$\Delta g_i(n_i) \geq c_i. \qquad (2)$$

Thus, letting $D_i = \{z \mid \Delta g_i(z) \geq c_i\}$, we obtain that $x_i = 1$ is the best response to z iff $z \in D_i$, proving the first claim. We now consider the three special cases:

(1) When g_i is concave, Δg_i is non-increasing. Therefore, whenever $\Delta g_i(z) \geq c_i$, we also have $\Delta g_i(z-1) \geq c_i$, meaning that D_i is downward closed.
(2) When g_i is convex, Δg_i is non-decreasing. The rest of the argument is exactly as for the concave case.
(3) When g_i is sigmoid, Δg_i is non-decreasing on $[0, \hat{z}]$ and non-increasing on $[\hat{z}, n-1]$, with the maximum attained at $\hat{z}$. Therefore, D_i is an interval.

For the converse, given a set D_i, define a discrete derivative of $\Delta g_i(z) = 2$ if $z \in D_i$, and $\Delta g_i(z) = 0$ if $z \notin D_i$. Normalizing with $g_i(0) = 0$, and setting $c_i = 1$, player i will invest iff $z \in D_i$. If D_i is an interval, then g_i will start out as the constant 0, have slope 2 over the interval, and then become flat at the end of the interval. Thus, g_i is a sigmoid. If the interval is downward-closed, then the function is concave; if it is upward-closed, the function is convex. □

The characterization of Lemma 2.3 makes precise our intuition behind considering concave/convex g_i. It shows that when g_i is concave, then the *fewer* neighbors invest, the more i is prone to invest. On the other hand, when g_i is convex, then the *more* neighbors invest, the more i is prone to invest. The primary benefit of Lemma 2.3 is that the Network Design problem can now be considered solely in terms of D_i and induced numbers of investing neighbors, rather than utility functions, simplifying the arguments below.

Definition 2.4 (Network Design for Degree Sets (NDDS)). The problem NDDS $(\mathcal{P}, \mathcal{X})$ is defined as follows: Given a graph $G' = (V, E')$, investment degree sets D_i for all players i consistent with a function property $\mathcal{P}$ (such as convexity, concavity, sigmoid, or general), edge costs γ_e, desired PSNE class $\mathcal{X}$, and budget B, find an edge set E with $\sum_{e \in E \triangle E'} \gamma_e \leq B$ such that there exists a set $I \in \mathcal{X}$ of investing players with

$$|\mathcal{N}_i^{(G)} \cap I| \in D_i \qquad \text{for all } i \in I,$$
$$|\mathcal{N}_i^{(G)} \cap I| \notin D_i \qquad \text{for all } i \notin I.$$

Here, $G = (V, E)$ is the modified graph.

Because the investment degree sets D_i can be efficiently constructed from the g_i and c_i and vice versa, an algorithmic solution or a hardness result for the NDDS $(\mathcal{P}, \mathcal{X})$ problem immediately yields the same result for the corresponding Network Design problem from Definition 2.2, and vice versa.

3 HARDNESS RESULTS

In this section, we prove the hardness results from Table 1. Hardness arises in different ways for different cases, and we treat them separately. For all versions, the problem is obviously in NP: a set I of investing agents forms a polynomial-sized witness, and it is easy to verify that (1) for each agent in I, investing is a best response, and (2) $I \in \mathcal{X}$.

NDDS (convex/concave, $\geq r$). When the goal is to get at least r agents to invest, NP-hardness follows from the discussion in Section 2.1. Even when all edge costs $\gamma_{(i,j)} = \infty$, i.e., the principal cannot add or remove any edges, it is NP-hard to decide whether the BNPG has an equilibrium in which at least r agents invest.

NDDS (concave, $\supseteq S$). When the goal is to get a superset of a given set S of agents to invest, it is NP-hard to decide whether a suitable equilibrium exists. Since concave functions are a special case of sigmoid functions, this result implies the hardness result for sigmoid functions as well.

THEOREM 3.1. *NDDS (concave, $\supseteq S$) is NP-hard.*

PROOF. The reduction is from INDEPENDENT SET. Given a graph $H = (V_H, E_H)$ and an integer k, the problem is to decide if H

contains an independent set of size (at least) k, i.e., a set $T \subseteq V_H$ such that no pair in T is connected by an edge. For the reduction, we add two nodes $u, \hat{u}$. u has an edge to $\hat{u}$ as well as to all nodes in V_H; no other edges are added to E_H. The degree sets are $D_v = 0$ for all $v \in V_H \cup \{\hat{u}\}$, $D_u = \{0, 1, \ldots, k\}$, and $S = \{\hat{u}\}$. We set all edge addition/removal costs to ∞.

If H has an independent set T of size at least k (without loss of generality, T is inclusion-wise maximal), then setting $I = T \cup \{\hat{u}\}$ gives us a superset of S. No $v \in I$ has a neighbor in I, so all of them invest. Each $v \in V_H \setminus I$ has at least one neighbor in I, so none of them invest. Finally, u has at least $k + 1$ neighbors in I, so u does not invest.

Conversely, if a superset $I \supseteq S = \{\hat{u}\}$ invests, then $u \notin I$. Therefore, u must have at least $k + 1$ neighbors in I; in particular, u has at least k neighbors in V_H. Because all of those neighbors are in I, their degrees within I must be 0, so they must form an independent set of size at least k. This completes the proof of NP-hardness. □

NDDS (general, all). For fully general functions g_i, we show that even the "easiest" goal — getting *all* agents to invest — is NP-hard. This immediately implies NP-hardness of the other cases (getting exactly or at least a subset S or at least r agents to invest), since their special cases $S = V$ or $r = n$ are hard.

Theorem 3.2. *NDDS (general, all) is NP-hard.*

Proof. We prove NP-hardness by a reduction from the Vertex Cover (VC) problem. In an instance of VC, we are given a graph $H = (V_H, E_H)$ and a positive integer k, and asked if H has a vertex cover of size at most k, i.e., a subset $S \subseteq V_H$ of at most k nodes such that each edge $e \in E_H$ has at least one endpoint in S. From H, k, we construct an instance of NDDS (general, all), consisting of a graph $G' = (V, E')$, investment degree sets D_i for each node $i \in V$, costs $\gamma_{(i,j)}$ for edge addition/removal, and a budget B.

The set of nodes V consists of V_H, one vertex u_e for each edge $e \in E_H$, and one additional vertex w. The edges E' are as follows:

- w is connected to all nodes in V_H.
- There is an edge between v and u_e if and only if v is an endpoint of e in H.

Let $d_H(v)$ be the degree of v in H. We define the investment degree sets for the agents as follows:

- For every $v \in V_H$, we let $D_v = \{0, d_H(v) + 1\}$.
- For every u_e, we let $D_{u_e} = \{1, 2\}$.
- $D_w = \{0, 1, \ldots, k\}$.

Finally, we set the costs $\gamma_e = 0$ for $e \in E'$, and $\gamma_{(i,j)} = \infty$ for $(i, j) \notin E'$. The budget is $B = 1$ (or really any non-negative number). Thus, the principal can remove as many edges as he wants, but cannot add any edges. This completes the reduction, which obviously runs in polynomial time.

First, assume that H has a vertex cover of size at most k. We show that there is a way to remove edges from E' such that each player i's degree ends up in D_i. Let S be the vertex cover of H. Let $E \subseteq E'$ be the set of all edges *not* incident on $V_H \setminus S$. Then, w is only incident on edges whose other endpoint is in S, so it has degree at most k. Each node $v \in V_H \setminus S$ has all its edges removed, so its degree is 0. Each node $v \in S$ is connected to $d_H(v)$ nodes corresponding to the $d_H(v)$ edges incident on v in H, plus its one edge to w, so

its degree is $d_H(v) + 1$. Finally, because S is a vertex cover, each node u_e is incident on at least one node $v \in S$, so its degree is 1 or 2. Thus, we have shown that each node $i \in V$ has degree in D_i.

For the converse direction, assume that there is a set $E \subseteq E'$ of edges such that in $G = (V, E)$, each node i has degree in D_i. Let $S = \{v \in V_H \mid (w, v) \in E\}$ be the set of vertices whose edge to w is kept. Because the degree of w is in D_w, we get that $|S| \leq k$. For each node $v \in V_H \setminus S$, at least the edge to w was removed, so its degree cannot be $d_H(v) + 1$. Therefore, its degree must be 0, so E cannot contain any edges incident on any $v \in V_H \setminus S$. For each node $v \in S$, at least the edge to w was retained, so its degree cannot be 0. Therefore, its degree must be $d_H(v) + 1$, so E must contain all edges incident on all $v \in S$. Finally, because each node u_e has degree in $\{1, 2\}$ in (V, E), each must have a neighbor in S. In other words, each edge $e \in E_H$ has at least one endpoint in S. This proves that S is a vertex cover of H. □

NDDS (convex, $\supseteq S$). Finally, we show that NDDS (convex, $\supseteq S$) is NP-hard. In contrast with the proof of Theorem 3.2, the hardness result for convex functions has to use edge costs other than 0 and ∞. The reason is that for convex sets, higher degrees are always preferable. Consequently, a principal will never remove edges (even if they are free to remove). On the other hand, if edge additions are free, the principal's optimal strategy is clearly to make G the complete graph. Either this will induce all nodes in S to invest, or no graph G will.

Theorem 3.3. *The problem NDDS (convex, $\supseteq S$) is NP-hard, even when $\gamma_{(i,j)} = 1$ for all $(i, j) \notin E'$, and $\gamma_e = 0$ for all $e \in E'$.*

Proof. We prove NP-hardness by a reduction from the k-Clique problem. In an instance of k-Clique, we are given a graph $H = (V_H, E_H)$ and a positive integer k, and asked if H has a clique of size at least k, i.e., a subset $S \subseteq V_H$ of at least k nodes such that $(u, v) \in E$ for all $u, v \in S, u \neq v$. From H, k, we construct an instance of NDDS (convex, $\supseteq S$), consisting of a graph $G' = (V, E')$, investment degree sets D_i for each node $i \in V$, costs $\gamma_{(i,j)}$ for edge addition/removal, and a budget B.

The graph G' consists of H, with a node-disjoint clique on nk nodes added. We call the set of new nodes V', and write $V = V' \cup V_H$. Thus, the new graph has $nk + n$ nodes. The investment degree of every node i is $D_i = \{nk + k - 1, \ldots, nk + n\}$. The cost of adding any non-existing edge (i, j) is $\gamma_{(i,j)} = 1$, and the cost for removing any existing edge e is $\gamma_e = 0$. The budget is $B = nk^2$. Notice that the D_i by Lemma 2.3 indeed correspond to convex functions g_i. Finally, the goal is to get a superset of V' to invest.

First, assume that H has a clique S of k nodes. Let E consist of all edges of G', plus a complete bipartite graph between V' and S. This added bipartite graph contains $kn \cdot k = k^2 n$ edges, so it satisfies the budget constraint. It is now immediate that each node in $V' \cup S$ has degree at least $nk + k - 1$. The nodes in $V_H \setminus S$ have degree at most $n - 1$. As a result, setting $I = V' \cup S$ satisfies Definition 2.4.

Conversely, let E be a set of edges with $|E \setminus E'| \leq nk^2$, and $I \subseteq V'$ a set of vertices such that in the graph (V, E), each node $v \in I$ has at least $nk + k - 1$ neighbors in I, and each node $v \notin I$ has at most $nk + k - 2$ neighbors in I. First, because each node $v \in V'$ started out with degree $nk - 1$ and has at least $nk + k - 1$ neighbors in I, $E \setminus E'$ must contain at least k incident edges for each such v. And

because E' already contained a clique on V', these edges must be between V' and V_H, so none of them are incident on two nodes of V'. Therefore, $E \setminus E'$ contains exactly k incident edges on each $v \in V'$.

Let S be the set of neighbors of V' in V_H. First, $|S| \geq k$, because each node $v \in V'$ is adjacent to k nodes in V_H. Next, we claim that $I = V' \cup S$. First, all nodes in S must be in I. The reason is that each $v \in V'$ has degree exactly $nk + k - 1$, so if even one of its neighbors were not in I, it couldn't have the required $nk + k - 1$ neighbors in I. Second, no node in $V_H \setminus S$ can be in I, because its degree is at most $n - 1 < nk + k - 1$.

For every node $v \in S$, let $k(v)$ be the number of neighbors of v in $S \cup V'$ in the graph $G = (V, E)$. Because $S \subseteq I$, we can lower-bound $k(v) \geq nk + k - 1$. On the other hand, $\sum_{v \in S} k_v \leq |S| \cdot (|S| - 1) + nk^2$, and because the minimum k_v is at most the average, we get that $nk + k - 1 \leq \min_{v \in S} k_v \leq (|S| - 1) + \frac{nk^2}{|S|}$. Rearranging this inequality gives us that $|S|(nk + k - |S|) \leq nk^2$. The left-hand side is a strictly concave function of $|S|$, and therefore attains its minimum at one of its endpoints $|S| \in \{k, n\}$. At $|S| = k$, the inequality holds with equality, while at $|S| = n$, it is violated. Therefore, $|S| = k$ is the only feasible solution of the inequality. Because each $v \in S$ only has nk neighbors in V', and none of its neighbors in $V_H \setminus S$ are in I, each $v \in S$ must have $k - 1$ neighbors in S. In other words, S is a clique of size k in H. □

4 TRACTABLE CASES

In this section, we give polynomial-time algorithms for the corresponding cases in Table 1. At the core of our algorithms lies a construction for the NDDS (sigmoid, *all*) problem, which is based on a reduction to the MINIMUM-COST PERFECT MATCHING problem. This reduction is a significant generalization of Tutte's reduction for finding a subgraph with a given degree sequence. The PERFECT MATCHING problem is polynomial-time solvable by utilizing the Blossom Algorithm proposed by Edmonds [8]. The remaining cases are either special cases of NDDS (sigmoid, *all*), or can be reduced to NDDS (sigmoid, *all*) fairly directly.

4.1 Tractability of NDDS (sigmoid, *all*)

Consider an instance of NDDS (sigmoid, *all*), consisting of a graph $G' = (V, E')$, investment degree sets D_i, costs $\gamma_{(i,j)}$ for edge addition/removal, and a budget B. The principal wants to modify G' to $G = (V, E)$ (at total cost at most B), such that all agents invest in a PSNE of the corresponding game. We construct an instance of the weighted perfect matching problem on a graph $H = (V_H, E_H)$ and show that the principal has a graph modification of cost at most B available iff G has a perfect matching of total cost at most B. Our construction generalizes Tutte [28]. We begin by describing Tutte's construction, and then present our generalization.

Tutte's reduction applies to the special case when edges can only be removed, and furthermore, each investment degree set $D_v = \{d_v\}$ is a singleton, called the *desired degree* of v. Each node v must have exactly $\Delta d_v := d_{G'}(v) - d_v$ of its edges removed. To encode this, Tutte's construction adds a node set X_v^- of Δd_v nodes $x_{v,j}^-, j = 1, \ldots, \Delta d_v$. Furthermore, it adds two nodes $y_{e,v}^-, y_{e,u}^-$ for every edge $e = (u, v)$. These two nodes are connected to each other, and to all nodes in their respective sets X_v^-, X_u^-. It is clear that

the construction takes polynomial time. Any perfect matching has to match all of the nodes in each X_v^-, capturing exactly the edges incident on v to be deleted. Because for each edge $e = (u, v)$, both $y_{e,v}^-$ and $y_{e,u}^-$ must be matched, either they are matched to each other (encoding that the edge is not deleted), or they must *both* be matched with nodes from the corresponding X_v^- and X_u^- sets. It is now straightforward that the new graph has a perfect matching iff the desired degree sequence can be obtained by edge removals. Edge removal costs can be assigned to the edges between X_v^- and the $y_{e,v}^-$.

Because the addition of edges corresponds to the removal of edges in the complement graph, a practically identical construction can be used directly if the goal is only to *add*, rather than remove edges. However, in NDDS (sigmoid, *all*), the principal can both add and remove edges. Furthermore, the investment degree sets D_i can be intervals containing multiple values. This necessitates significant extensions to Tutte's construction.

We now describe our generalized construction, where a graph $H = (V_H, E_H)$ is constructed from G'. For every agent $i \in V$, the degree in G' is $d_{G'}(i)$, and the degree set is $D_i = \{L_i, \ldots, R_i\}$, where L_i (resp., R_i) is the minimum (resp., maximum) of D_i. If any set D_i is empty, then the instance clearly has no solution, and this is easy to diagnose. From now on, we assume that $D_i \neq \emptyset$ for all i. At the core of the construction is the union of the Tutte construction for both additions and removals of edges. Thus, for each edge $e = (i, i') \in E'$ (a candidate for removal), we add two nodes $y_{e,i}^-$ and $y_{e,i'}^-$ with an edge between them; similarly for each node pair $e' = (i, i') \notin E'$ (a candidate for addition), we add two nodes $y_{e',i}^+$ and $y_{e',i'}^+$ with an edge between them.

Next, we describe the node gadget for a node i. An illustrative example is shown in Figure 1. We add a set X_i^+ of $\min(R_i, n - d_{G'}(i) - 1)$ nodes $x_{i,j}^+$ (blue nodes in Figure 1), corresponding to additions of edges, and a set X_i^- of $\min(n - L_i - 1, d_{G'}(i))$ nodes $x_{i,j}^-$ (green nodes in Figure 1), corresponding to edge removals. These are hard upper bounds on the number of possible edge additions/removals: for $|X_i^+|$, the first term arises because even if all existing edges were deleted, no more than R_i new edges can be safely added; the second term is because there are only $n - d_{G'}(i) - 1$ potential edges for addition. The justification is similar for $|X_i^-|$. As in Tutte's construction, we add an edge between each node $x_{i,j}^+$ and each $y_{e',i}^+$. Similarly, we add an edge between each node $x_{i,j}^-$ and each $y_{e,i}^-$. Finally, we add a complete bipartite graph between X_i^- and X_i^+.

As in Tutte's construction, including an edge between $x_{i,j}^+$ and $y_{e',i}^+$ in a matching corresponds to adding the edge e' (increasing the degree of i), and including the edge $(x_{i,j}^-, y_{e,i}^-)$ corresponds to removing the edge e, decreasing the degree of i. Because no other edges are incident on $y_{e',i}^+, y_{e,i}^-$, for any edge $e = (i, i')$, either $y_{e,i}^-$ is matched with $y_{e,i'}^-$, or both are matched with nodes from X_i^- (resp., $X_{i'}^-$); similarly for the $y_{e,i}^+$ nodes. The complete bipartite graph between X_i^- and X_i^+ allows us to encode that adding one fewer edge and removing one fewer edge has the same effect on i's degree as adding and removing one more edge.

We now expand the gadget to encode the set D_i. The intuition for the generalized gadget is the following: if k_i^+ nodes in X_i^+ are matched with nodes $y_{e',i}^+$, and k_i^- nodes in X_i^- are matched with
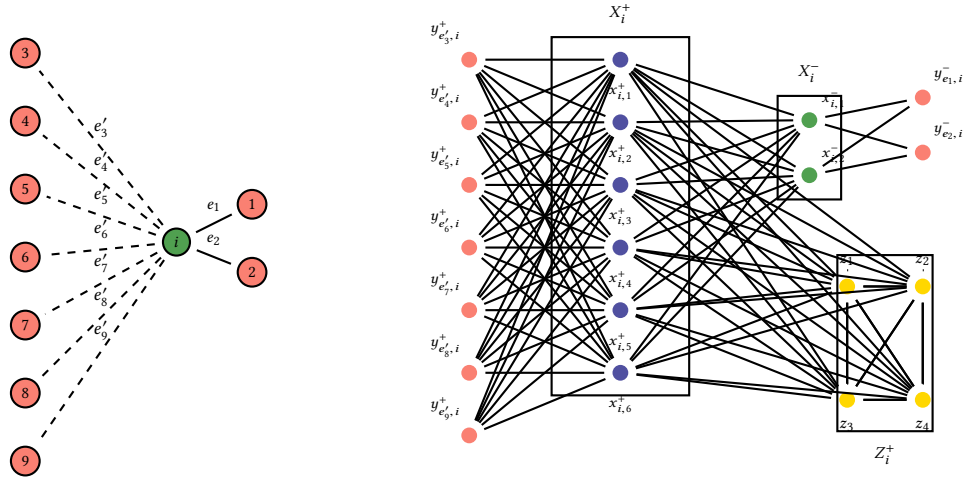
Figure 1: Example to illustrate our generalized construction. Left: An example graph where $d_{G'}(i) = 2$, $D_i = \{4, 5, 6\}$, and $n = 10$; Right: the subgraph $H(i)$ associated with i.

nodes $y^-_{e,i}$, then the new degree of i is $d_{G'}(i) + k^+_i - k^-_i$. We want to force this number to be in $D_i = [L_i, R_i]$ for every perfect matching. If we let $\hat{k}^+_i = |X^+_i| - k^+_i$ and $\hat{k}^-_i = |X^-_i| - k^-_i$ be the number of nodes in X^+_i, X^-_i that are matched differently (i.e., not with $y^+_{e',i}$, $y^-_{e,i}$), then the necessary/sufficient condition can be expressed as $\hat{k}^-_i - \hat{k}^+_i \in [L_i - (d_{G'}(i) + |X^+_i| - |X^-_i|), R_i - (d_{G'}(i) + |X^+_i| - |X^-_i|)]$. Furthermore, notice that our gadget will only need to work if at least one of $\hat{k}^-_i$, $\hat{k}^+_i$ is 0, since the complete bipartite graph between X^-_i, X^+_i can always be used to ensure this condition.

Let $\sigma_i = d_{G'}(i) + |X^+_i| - |X^-_i|$. A case distinction on the possible cases of the minimum in the definitions of $|X^+_i|$, $|X^-_i|$ shows that we always have $L_i \leq \sigma_i \leq R_i$. Therefore, $L_i - \sigma_i \leq 0 \leq R_i - \sigma_i$. We generate two more node sets Z^+_i, Z^-_i. Z^+_i consists of $\sigma_i - L_i$ nodes $z^+_{i,j}$, and Z^-_i consists of $R_i - \sigma_i$ nodes $z^-_{i,j}$. There is a complete bipartite graph between Z^+_i and X^+_i, as well as between Z^-_i and X^-_i. In addition, there is a complete graph on the union of all of the Z^+_i and Z^-_i, for all i. If the total number of nodes in the construction is odd, then we add one more node $\hat{z}$ and connect it to all nodes in all of the Z^+_i and Z^-_i. The $z^+_{i,j}$, $z^-_{i,j}$ are there to match any otherwise unmatched nodes $x^+_{i,j}$, $x^-_{i,j}$. Whichever ones of them are not needed can be matched with each other and with $\hat{z}$.

Finally, for every edge $e = (i, i') \in G'$, we assign a cost of $\gamma_e/2$ to the edges $(x^-_{i,j}, y^-_{e,i})$ and $(x^-_{i',j}, y^-_{e,i'})$ (for all j); similarly, for every edge $e' = (i, i') \notin G'$, we assign a cost of $\gamma_{e'}/2$ to the edges $(x^+_{i,j}, y^+_{e',i})$ and $(x^+_{i',j}, y^+_{e',i'})$ (for all j). All other edges have cost 0. The cost bound for the perfect matching is the given budget B. The correctness of this reduction is captured by the following theorem:

THEOREM 4.1. *The graph H has a perfect matching of total cost at most B if and only if there is an edge modification E of the input graph G' such that in (V, E), the degrees of all nodes i are in their respective investment sets D_i.*

PROOF. First, we assume that there is an edge set $E \subseteq V \times V$ such that $\sum_{e \in E' \triangle E} \gamma_e \leq B$, and in the graph $G = (V, E)$, every node i has degree $d_G(i) \in D_i$. We define a perfect matching M in H.

For edges $e = (i, i') \in E \cap E'$, the matching includes the edge $(y^-_{e,i}, y^-_{e,i'})$; similarly, for edges $e' = (i, i') \notin E, e' \notin E'$, it contains the edge $(y^+_{e',i}, y^+_{e',i'})$.

Now focus on one node i. Let k^+_i, k^-_i be the numbers of edges that were added to (resp., removed from) i, i.e., the numbers of edges incident on i in $E \setminus E'$ and $E' \setminus E$. Let $e'_1, \ldots, e'_{k^+_i}$ be an enumeration of the added edges (in arbitrary order), and $e_1, \ldots, e_{k^-_i}$ an enumeration of the removed edges in arbitrary order. For each e'_j, the matching M includes the edge $(x^+_{i,j}, y^+_{e'_j,i})$; similarly, for each e_j, M includes the edge $(x^-_{i,j}, y^-_{e_j,i})$. Doing this for all i ensures that all nodes $y^+_{e',i}$, $y^-_{e,i}$ are matched, and the total cost of all edges is exactly $\sum_{e \in E' \triangle E} \gamma_e \leq B$. This cost will not change by the inclusion of later edges, since they all have cost 0.

Next, let $m_i := \min(|X^+_i| - k^+_i, |X^-_i| - k^-_i)$. Notice that $m_i \geq 0$, because our definition of $|X^+_i|$, $|X^-_i|$ ensured that no edge set E with $d_G(i) \in D_i$ could add/remove more than $|X^+_i|$ (resp., $|X^-_i|$) edges. We next add a perfect matching of m_i edges $(x^+_{i,k^+_i+j}, x^-_{i,k^-_i+j})$ for $j = 1, \ldots, m_i$. At this point, at least one of the sets X^+_i, X^-_i is completely matched. For the remaining description, assume that X^-_i is fully matched — the other case is symmetric. Now, there are

$$|X^+_i| - k^+_i - m_i \leq |X^+_i| - |X^-_i| + (k^-_i - k^+_i)$$
$$= (\sigma_i - d_{G'}(i)) + (k^-_i - k^+_i)$$
$$= \sigma_i - d_G(i)$$

unmatched nodes in X^+_i. Because $d_G(i) \in D_i$, it must satisfy $d_G(i) \geq L_i$; therefore, because Z^+_i contains $\sigma_i - L_i \geq \sigma_i - d_G(i)$ nodes, it has enough nodes to perfectly match the remaining nodes of X^+_i — we add such a perfect matching. Finally, we add a perfect matching on the unmatched nodes of all Z^+_i, Z^-_i (and $\hat{z}$) — this

is possible, because H contains a complete graph on these nodes, the total number of nodes in H is even, and the number of nodes matched so far is (by definition of a matching) even. Thus, we have shown that H contains a perfect matching of the desired cost.

For the converse direction, we assume that H contains a perfect matching M of cost at most B. Define edge sets $E^+ = \{e' = (i, j) \notin E' \mid (y_{e',i}^+, y_{e',i'}^+) \notin M\}$ and $E^- = \{e = (i, j) \in E' \mid (y_{e,i}^-, y_{e,i'}^-) \notin M\}$. That is, E^+ consists of the edges for which addition was encoded in the Tutte reduction part, and E^- of the edges for which removal was encoded in the Tutte reduction part. Let $E = E' \cup E^+ \setminus E^-$. Because M is a perfect matching, it must include edges of the form $(x_{i,j}^+, y_{e',i}^+), (x_{i',j}^+, y_{e',i'}^+)$ for all edges $e' = (i, i') \in E^+$, and edges of the form $(x_{i,j}^-, y_{e,i}^-), (x_{i',j}^-, y_{e,i'}^-)$ for all edges $e = (i, i') \in E^-$. In particular, the total cost of $E \triangle E'$ is exactly B.

It remains to show that in the graph $G = (V, E)$, each node i has degree $d_G(i) \in D_i$. Let k_i^+ be the number of edges in E^+ incident on i, and k_i^- the number of edges in E^- incident on i. Then, because the $y_{e',i}^+$ for $e' \in E^+$ are *not* matched to $y_{e',i'}^+$, they must be matched to some $x_{i,j}^+$; similarly, the $y_{e,i}^-$ for $e \in E^-$ are matched to some $x_{i,j}^-$. In particular, this means that $k_i^+ \leq |X_i^+|, k_i^- \leq |X_i^-|$. Furthermore, because X_i^+ and X_i^- are completely matched, and they can only be matched with each other and $y_{e',i}^+$ and Z_i^+ ($y_{e,i}^-$, or Z_i^-, respectively), we infer that $(|X_i^+| - k_i^+) - (|X_i^-| - k_i^-) \leq \sigma_i - L_i$ and $(|X_i^-| - k_i^-) - (|X_i^+| - k_i^+) \leq R_i - \sigma_i$. Substituting the definition of σ_i, these inequalities rearrange to $L_i \leq d_{G'}(i) + k_i^+ - k_i^- = d_G(i)$ and $R_i \geq d_{G'}(i) + k_i^+ - k_i^- = d_G(i)$. Thus, we have shown that $d_G(i) \in [L_i, R_i]$, so the degree constraint for i is met. Since this holds for all i, the proof is complete. □

The reduction clearly runs in polynomial time (and is in fact fairly straightforward), and the Minimum-Cost Perfect Matching problem is known to be solvable in polynomial time [8]. Thus, we obtain a polynomial-time algorithm for the NDDS (sigmoid, *all*) problem, as claimed. Because convex and concave functions are special cases of sigmoid functions, NDDS (convex/concave, *all*) are also polynomial-time solvable.

4.2 Tractability of NDDS (sigmoid, $= S$)

Finally, we leverage the algorithm from Section 4.1 for the more general problem NDDS (sigmoid, $= S$).

Consider a hypothetical solution $G = (V, E)$. Then, for every node $i \notin S$, we must have $|\mathcal{N}_i^{(G)} \cap S| \notin D_i$. Edges between node pairs $i, i' \notin S$ do not matter. Similarly, because exactly the nodes of S are supposed to invest, for the purpose of investment decisions of nodes $i \in S$, edges to nodes not in S do not matter. Thus, as a first step, an algorithm can add/remove edges between S and $V \setminus S$ of minimum total cost to ensure that $|\mathcal{N}_i^{(G)} \cap S| \notin D_i$ for all $i \notin S$. This can be accomplished easily node by node: when considering node i, either the principal will add $R_i + 1 - d_{G'}(i)$ edges or remove $d_{G'}(i) - (L_i - 1)$ edges. In both cases, the minimum-cost edges incident on i will be chosen. If these additions/removals exceed the budget B, then no solution is possible. Otherwise, they will be performed, and the budget updated to the remaining budget.

After the removal of these edges, the agents in $V \setminus S$ are irrelevant; the sole goal is to alter the edges within S at minimum cost to

meet the degree constraints. This is an instance of the problem NDDS (sigmoid, *all*) on the induced graph $G'[S]$, which can be solved using the algorithm from Section 4.1. Thus, we have proved the following theorem:

Theorem 4.2. *The problem NDDS (sigmoid, $= S$) is polynomial-time solvable.*

Again, since convex and concave functions are special cases of sigmoid functions, the tractability of NDDS (convex/concave, $= S$) follows from Theorem 4.2.

5 CONCLUSION

The problem of modifying elements of a game structure to achieve desired outcomes has a long history and interest in both economics and computing, with mechanism design the classic variation. In mechanism design, a key design parameter is the payment scheme for the players. The somewhat more recent literature on market design is often focused on settings where payments are infeasible, and aims to design market structure, such as the rules of the matching markets. An even more recent thread considers the problem of designing signals that modify information available to the players, thereby inducing particular desirable outcomes. We suggest considering a fourth element of the game in settings where strategic dependencies among players are mediated by a network: the design of the network structure. Such design decisions are commonly inherently constrained by an already existing network, and we specifically consider the simplest and most natural design action: adding and removing links. Additionally, to elucidate both the process and the associated algorithmic mechanics, we further delve deeply into a study of network design for networked public goods games, with the goal of inducing desired pure strategy equilibrium outcomes. The significance of our work is thus both in proposing a novel framework for designing the rules of encounter specific to networked game theoretic scenarios, and elucidating the algorithmic complexity of this problem in the particular context of networked public goods games.

Our work provides an initial step, but leaves open a number of research questions. First, our focus on adding and removing edges with an additive addition/removal cost clearly limits the scope of applicability. In general, one would encounter numerous complications. For example, if the means for adding edges is through the design of events, then the cost would be incurred for adding a collection of edges (i.e., organizing an event), rather than adding each edge independently. Indeed, one could consider a broad space of reasonable cost functions that generalize additivity, such as submodular costs. Second, the problem of inducing equilibria through network modifications is interesting far more broadly than just networked public goods games. For example, such network design issues arise in congestion games. Third, we only considered the issue of inducing pure strategy Nash equilibria. It is, of course, natural to study other equilibrium concepts, such as mixed-strategy equilibria and correlated equilibria.

REFERENCES

[1] Victor Amelkin and Ambuj K. Singh. 2019. Fighting opinion control in social networks via link recommendation. In *Proc. 25th*. 677–685.

[2] Elliot Anshelevich, Anirban Dasgupta, Jon Kleinberg, Eva Tardos, Tom Wexler, and Tim Roughgarden. 2008. The price of stability for network design with fair cost allocation. *SIAM J. Comput.* 38, 4 (2008), 1602–1623.

[3] Yann Bramoullé and Rachel Kranton. 2007. Public goods in networks. *Journal of Economic Theory* 135, 1 (2007), 478–494.

[4] Yann Bramoullé, Rachel Kranton, and Martin D'Amours. 2014. Strategic Interaction and Networks. *American Economic Review* 104, 3 (2014), 898–930.

[5] Robert Bredereck and Edith Elkind. 2017. Manipulating Opinion Diffusion in Social Networks. In *Proc. 26th*. 894–900.

[6] Chen Chen, Hanghang Tong, B. Aditya Prakash, Tina Eliassi-Rad, Michalis Faloutsos, and Christos Faloutsos. 2016. Eigen-Optimization on Large Graphs by Edge Manipulation. *ACM Transactions on Knowledge Discovery from Data* 10, 4 (2016), 49:1–49:30.

[7] Shaddin Dughmi. 2017. Algorithmic Information Structure Design: A Survey. *ACM SIGecom Exchanges* 15, 2 (2017), 2–24.

[8] Jack Edmonds. 1965. Paths, trees, and flowers. *Canadian Journal of Mathematics* 17 (1965), 449–467.

[9] Michal Feldman, David Kempe, Brendan Lucier, and Renato Paes Leme. 2013. Pricing Public Goods for Private Sale. In *Proc. 14th*. 417–434.

[10] Andrea Galeotti, Sanjeev Goyal, Matthew O. Jackson, Fernando Vega-Redondo, and Leeat Yariv. 2010. Network Games. *The Review of Economic Studies* 77, 1 (2010), 218–244.

[11] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. 2018. Reducing Controversy by Connecting Opposing Views. In *Proc. 27th*. 5249–5253.

[12] Arpita Ghosh and Stephen Boyd. 2006. Growing Well-Connected Graphs. In *Proc. 45th IEEE Conference on Decision and Control (CDC)*. 6605–6611.

[13] Jens Grossklags, Nicolas Christin, and John Chuang. 2008. Security and insurance management in networks with heterogeneous agents. In *Proc. 9th*. 160–169.

[14] Guillaume Haeringer. 2018. *Market Design: Auctions and Matching.* The MIT Press.

[15] Ashish R. Hota and Shreyas Sundaram. 2016. Optimal network topologies for mitigating security and epidemic risks. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton).* IEEE, 1129–1136.

[16] Michael Kearns, Michael L. Littman, and Satinder Singh. 2001. Graphical models for game theory. In *Proc. 17th*. 253–260.

[17] Vadim Levit, Zohar Komarovsky, Tal Grinshpoun, and Amnon Meisels. 2018. Incentive-based search for efficient equilibria of the public goods game. *Artificial Intelligence* 262 (2018), 142–162.

[18] Andreu Mas-Collel, Michael D. Whinston, and Jerry R. Green. 1995. *Microeconomic Theory.* Oxford University Press.

[19] Nicola Gatti Matteo Castiglioni, Diodato Ferraioli. 2020. Election Control in Social Networks via Edge Addition or Removal. In *Proc. 34th*. AAAI Press.

[20] Igal Milchtaich. 2015. Network topology and equilibrium existence in weighted network congestion games. *International Journal of Game Theory* 44, 3 (2015), 515–541.

[21] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani (Eds.). 2007. *Algorithmic Game Theory.* Cambridge University Press.

[22] Paul A. Samuelson. 1954. The pure theory of public expenditure. *Review of Economics and Statistics* 36 (1954), 387–389.

[23] Daniel Sheldon, Bistra Dilkina, Adam N. Elmachtoub, Ryan Finseth, Ashish Sabharwal, Jon Conrad, Carla Gomes, David Shmoys, William Allen, Ole Amundsen, and William Vaughan. 2010. Maximizing the Spread of Cascades Using Network Design. In *Proc. 26th*. 517–526.

[24] Yoav Shoham and Kevin Leyton-Brown. 2009. *Multiagent Systems.* Cambridge University Press.

[25] Sigal Sina, Noam Hazon, Avinatan Hassidim, and Sarit Kraus. 2015. Adapting the social network to affect elections. In *Proc. 14th*. 705–713.

[26] Liat Sless, Noam Hazon, Sarit Kraus, and Michael Wooldridge. 2014. Forming coalitions and facilitating relationships for completing tasks in social networks. In *Proc. 13th*. 261–268.

[27] Hanghang Tong, B. Aditya Prakash, Tina Eliassi-Rad, Michalis Faloutsos, and Christos Faloutsos. 2012. Gelling, and melting, large graphs by edge manipulation. In *Proc. 21st*. 245–254.

[28] William Thomas Tutte. 1954. A short proof of the factor theorem for finite graphs. *Canadian Journal of Mathematics* 6 (1954), 347–352.

[29] Sixie Yu, Kai Zhou, P. Jeffrey Brantingham, and Yevgeniy Vorobeychik. 2020. Computing Equilibria in Binary Networked Public Goods Games. In *Proc. 34th*. AAAI Press.

[30] Haifeng Zhang, Yevgeniy Vorobeychik, Joshua Letchford, and Kiran Lakkaraju. 2016. Data-driven agent-based modeling, with application to rooftop solar adoption. *Journal of Autonomous Agents and Multiagent Systems* 30, 6 (2016), 1023–1049.