Finding Nontrivial Minimum Fixed Points in Discrete Dynamical Systems: Complexity, Special Case Algorithms and Heuristics

Zirou Qiu,[1,2] **Chen Chen,**[2] **Madhav V. Marathe,**[1,2] **S. S. Ravi,**[2,3]
Daniel J. Rosenkrantz,[2,3] **Richard E. Stearns,**[2,3] **Anil Vullikanti**[1,2]
[1] Computer Science Dept., University of Virginia. [2] Biocomplexity Institute and Initiative, University of Virginia.
[3] Computer Science Dept., University at Albany – SUNY.
{zq5au, vsakumar, marathe}@virginia.edu, {chenannie45, drosenkrantz, ssravi0, thestearns2}@gmail.com

Abstract

Networked discrete dynamical systems are often used to model the spread of contagions and decision-making by agents in coordination games. Fixed points of such dynamical systems represent configurations to which the system converges. In the dissemination of undesirable contagions (such as rumors and misinformation), convergence to fixed points with a small number of affected nodes is a desirable goal. Motivated by such considerations, we formulate a novel optimization problem of finding a nontrivial fixed point of the system with the minimum number of affected nodes. We establish that, unless **P = NP**, there is no polynomial time algorithm for approximating a solution to this problem to within the factor $n^{1-\epsilon}$ for any constant $\epsilon > 0$. To cope with this computational intractability, we identify several special cases for which the problem can be solved efficiently. Further, we introduce an integer linear program to address the problem for networks of reasonable sizes. For solving the problem on larger networks, we propose a general heuristic framework along with greedy selection methods. Extensive experimental results on real-world networks demonstrate the effectiveness of the proposed heuristics.

Reproducibility: A full version of the manuscript, source code and data are available at:
https://github.com/bridgelessqiu/NMIN-FPE.

1 Introduction

Discrete dynamical systems are commonly used to model the propagation of contagions (e.g., rumors, failures of subsystems in infrastructures) and decision-making processes in networked games (Valdez et al. 2020; Jackson 2010). Specifically, the states of nodes in such dynamical systems are binary, with state 1 indicating the adoption of a contagion. At each time step, the states of the nodes are updated using their *local functions*. When the local functions are threshold function, a node v acquires a contagion (i.e., v changes to state 1) if the number of v's neighbors that have adopted the contagion (i.e., v's peer strength) is at least a given threshold value. Conversely, an individual's adoption of a contagion is *reversed* (i.e., v changes to state 0) when the peer strength is below the threshold (Barrett et al. 2006). Since its introduction by Granovetter (1978), the threshold model has been extensively studied in many contexts including opinion dynamics (Auletta, Ferraioli, and Greco 2018), information diffusion (Cheng et al. 2018) and the spread of social conventions and rumors (Dong et al. 2019; Ye et al. 2021). The threshold model also captures decision patterns in networked coordination games (Ramazi, Riehl, and Cao 2016).

One important stage of the system dynamics is the convergence of nodes' states, where no individuals change states further; this is similar to an equilibrium in a networked game (Daskalakis and Papadimitriou 2007, 2015). Such a stage is called a **fixed point** of the dynamical system. Consider a scenario where a rumor is spreading in a community under the threshold model; here, an individual v chooses to believe the rumor if the number of believers in v's social circle is at least the threshold of v. Given the undesirable nature of rumors, identifying fixed points with minimum numbers of believers is desirable (Wang et al. 2017).

For some social contagions that are widely adopted in communities, it is often unrealistic to expect contagions to eventually disappear spontaneously. One example is the anti-vaccination opinion, which emerged in 1853 against the smallpox vaccine (Wolfe and Sharp 2002). Even today, the anti-vaccination sentiment persists across the world (Willis et al. 2021). Such considerations motivate us to study a more realistic problem, namely determining whether there are fixed points with at most a given number of contagion adoptions under the *nontriviality constraint* that the number of adoptions in the fixed point must be nonzero. We refer to this as the **nontrivial minimum fixed point existence** problem (NMIN-FPE).

Nontrivial minimum fixed points of a system, which are jointly determined by the network structure and local functions, provide a way of quantifying the system's resilience against the spread of negative information. In particular, the number of contagion adoptions in a nontrivial minimum fixed point provides the lower bound on the number of individuals affected by the negative contagion. Further, when the complete absence of a contagion is impractical, nontrivial minimum fixed points serve as desirable convergence points for control strategies (Khalil, Dilkina, and Song 2013). Similarly in coordination games, one is interested in finding equilibria wherein only a small number of players deviate from the strategy adopted by a majority of the players (Ramazi, Riehl, and Cao 2016).

As we will show, the main difficulty of the NMIN-FPE problem lies in its computational complexity. A related problem is that of influence minimization (e.g., (Yao et al. 2015)). The main differences between the two problems are twofold. First, the influence minimization problem is based on the progressive model where a node state can only change from 0 to 1 but not vice versa. Second, the influence minimization problem aims to find optimal intervention strategies (e.g., node/edge removal) to reduce the cascade size, while NMIN-FPE aims to find a minimum influenced group without changing the system.

In this work, we study the NMIN-FPE problem on synchronous dynamical systems (SyDS) with threshold local functions, where the nodes update states simultaneously in each time-step. Our main contributions are as follows:

1. **Formulation.** We formally define the Nontrivial Minimum Fixed Point Existence Problem (NMIN-FPE) from a combinatorial optimization perspective.
2. **Intractability.** We establish that unless **P** $=$ **NP**, NMIN-FPE cannot be approximated to within the factor $n^{1-\epsilon}$ for any $\epsilon > 0$, even when the graph is bipartite. We also show that the NMIN-FPE is **W[1]**-hard w.r.t. the natural parameter (i.e., the number of nodes in state 1 in any nontrivial fixed point) of the problem.
3. **Algorithms.** We identify several special cases for which NMIN-FPE can be solved in polynomial time. To obtain an optimal solutions for networks of moderate size, we present an integer linear program (ILP) formulation for NMIN-FPE. For larger networks, we propose a heuristic framework along with three greedy selection strategies that can be embedded into the framework.
4. **Evaluation.** We conduct extensive experiments to study the performance of our heuristics on real-world networks under various scenarios. Our results demonstrate that the proposed heuristics are exceptionally effective and outperform baseline methods significantly, despite the strong inapproximability of NMIN-FPE.

2 Related work

Fixed points. Fixed points of discrete dynamical systems have been widely studied. Goles and Martinez (2013) show that that for any initial configuration, a threshold SyDS always converges to either a fixed point or a cycle with two configurations in a polynomial number of time steps. Barrett et al.(2007) show that determining whether a system has a fixed point (FPE) is **NP**-complete for symmetric sequential dynamical systems and that the problem is efficiently solvable for threshold sequential dynamical systems. More recently, Chistikov et al. (2020) study fixed points in the context of opinion diffusion; they show that determining whether a system reaches a fixed point from a given configuration is **PSPACE**-complete for SyDSs on general directed networks, but can be solved in polynomial time when the underlying graph is a DAG. In Rosenkrantz et al. (2021), we investigate convergence and other problems for SyDSs whose underlying graphs are DAGs. In particular, we show that the convergence guarantee problem (i.e., determining if a system reaches a fixed point starting from any configuration) is **Co-NP**-complete for SyDSs on DAGs.

Influence minimization. Existing works on influence minimization focus on reducing the prevalence via control strategies. Yang, Li and Giua (2019) study the problem of finding a k-subset of active nodes such that the converged influence value is minimized and a target set of nodes are active. They provide an integer program of the problem and suggest two heuristics. Wang et al. (2017) propose a new rumor diffusion model and optimize blocking the contagion by considering an Ising model. Zhu, Ni, and Wang (2020) estimate the influence of nodes and minimize the adoption of negative contagions by disabling nodes. Other approaches focus on blocking the spread via node removal (Kimura, Saito, and Nakano 2007; Yao et al. 2015; Chen et al. 2015; Kuhlman et al. 2015) or edge removal (Kimura, Saito, and Motoda 2008; Khalil, Dilkina, and Song 2013; Chen et al. 2016; Kuhlman et al. 2013) and enhance network resilience (Chen et al. 2015).

Coordination games. Agent decision-making in coordination games coincides with threshold-based cascade of contagions. Adam et al. (2012) study the best response dynamics of coordination games and analyze the convergences and propose a new network resilience measure. Ramazi et al. (2016) study both coordination and anticoordination games and show that such games always reach equilibria in a finite amount of time. Other aspects of equilibria in networked games (such as developing control strategies and determining the existence of equilibria) have also been studied ((Yu et al. 2020; Cao, Ertin, and Arora 2008; Anderson, Goeree, and Holt 2001; Salehisadaghiani and Pavel 2018)).

3 Preliminaries and Problem Definition

3.1 Preliminaries

We follow the definition of discrete dynamical systems from previous work (Rosenkrantz et al. 2021). A **synchronous dynamical system** (SyDS) $\mathcal{S}$ over the Boolean domain $\mathbb{B} = \{0,1\}$ of state values is defined as a pair $(G_{\mathcal{S}}, \mathcal{F})$ where (1) $G_{\mathcal{S}} = (V, E)$ is the underlying graph of $\mathcal{S}$ with $n = |V|$ and $m = |E|$, and (2) $\mathcal{F} = \{f_1, ..., f_n\}$ is a collection of functions for which f_i is the *local transition function* of node $v_i \in V, 1 \leq i \leq n$. In general, $f_i \in \mathcal{F}$ specifies how $v_i \in V$ updates its state throughout the evolution of $\mathcal{S}$. In this work, we study SyDSs over the Boolean domain with threshold functions as local functions. Following (Barrett et al. 2006), We denote such a system by $(\text{BOOL}, \text{THRESH})$-SyDS.

Update rules. In $(\text{BOOL}, \text{THRESH})$-SyDSs, each node $v_i \in V$ has a fixed integer threshold value $\tau_{v_i} \geq 0$. At each time step $t \geq 0$, each node $v_i \in V$ has a state value in $\mathbb{B}$. While the initial state of any node v_i (at time 0) can be assigned arbitrarily, the states at time steps $t \geq 1$ are determined by v_i's local function f_i. Specifically, v_i transitions to state 1 at time t if the number of state-1 nodes in its closed neighborhood $N(v_i)$ (which consists of v_i and all its neighbors) at time $t-1$ is at least τ_{v_i}; the state of v_i at time t is 0 otherwise. Furthermore, all nodes update their states synchronously. When $G_{\mathcal{S}}$ is a directed graph, a node v transitions to state 1 at a time $t \geq 1$ iff the number of state-1 *in-neighbors* of v (i.e., node v itself and those from which v has incoming edges) is at least τ_v. Note that undirected networks are the primary focuses of this work; we assume that $G_{\mathcal{S}}$ is undirected unless specified otherwise.

Configurations and fixed points. A **configuration** of $\mathcal{S}$ gives the states of all nodes during a time-step. Specifically, a configuration C is an n-vector $C = (C(v_1), C(v_2), ..., C(v_n))$ where $C(v_i) \in \mathbb{B}$ is the state of node $v_i \in V$ under C. There are a total of 2^n possible configurations a system $\mathcal{S}$. During the evolution of the $\mathcal{S}$, the system configuration C changes over time. If $\mathcal{S}$ transitions from C to C' in one time step, then C' is the *successor* of C. Due to the deterministic nature of $(\text{BOOL}, \text{THRESH})$-SyDSs, if $C = C'$, that is, the states of all nodes remain unchanged, then C is a **fixed point** of the system. An example of a $(\text{BOOL}, \text{THRESH})$-SyDS $\mathcal{S} = (G_{\mathcal{S}}, \mathcal{F})$ is shown in Figure 1. Note that given any configuration C, its successor C' can be computed in time that is polynomial in the number of nodes n. As shown in (Goles and Martínez 2013), starting from any initial configuration, $\mathcal{S}$ converges either to a fixed point or a cycle consisting of two configurations within a number of transitions that is a polynomial in n.

Constant-state nodes. A node $v \in V$ is a *constant-1* node if $\tau_v = 0$, that is, given any configuration C, the state of v is 1 in the successor C' of C. Similarly, v is a *constant-0* node if $\tau_v = deg(v) + 2$.

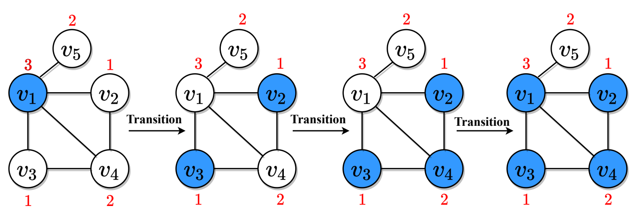

Figure 1: The evolution of a $(\text{BOOL}, \text{THRESH})$-SyDS $\mathcal{S} = (G_{\mathcal{S}}, \mathcal{F})$ where $V(G_{\mathcal{S}}) = \{v_i : i = 1, ..., 5\}$. The threshold values are shown in red: $\tau_1 = 3, \tau_2 = 1, \tau_3 = 1, \tau_4 = 2$, and $\tau_5 = 2$. State-1 nodes are highlighted in blue. The system undergoes the following evolution: $(1, 0, 0, 0, 0) \to (0, 1, 1, 0, 0) \to (0, 1, 1, 1, 0) \to (1, 1, 1, 1, 0)$, with the last configuration being a fixed point.

3.2 Problem definition

Let $\mathcal{S} = (G_{\mathcal{S}}, \mathcal{F})$ be a SyDS and let C be a configuration of $\mathcal{S}$. The *Hamming weight* of C, denoted by $H(C)$, is the number of 1's in C. A *minimum fixed point* of $\mathcal{S}$ is a fixed point with the smallest possible Hamming weight. Note that when a $(\text{BOOL}, \text{THRESH})$-SyDS $\mathcal{S}$ has no constant-1 nodes, the minimum fixed point of $\mathcal{S}$ is trivially $\mathbf{0}^n$. A *nontrivial fixed point* is a fixed point that is different from $\mathbf{0}^n$. Our work focuses on finding *nontrivial minimum fixed points*.

Definition 3.1. A **nontrivial minimum fixed point** of $\mathcal{S}$ is a nontrivial fixed point of minimum Hamming weight.

We now provide a formal definition of the problem.

Nontrivial Minimum Fixed Point Existence (NMIN-FPE)
Instance: A SyDS $\mathcal{S} = (G_{\mathcal{S}}, \mathcal{F})$ and a positive integer q.
Question: Is there a fixed point C of $\mathcal{S}$ with Hamming weight at least 1 and at most q?

We focus on the **NP**-optimization version of NMIN-FPE which is to find a nontrivial minimum fixed point.

4 Computational Hardness of NMIN-FPE

In this section, we present an inapproximability result for NMIN-FPE. Specifically, we show that NMIN-FPE cannot be poly-time approximated within a factor $n^{1-\epsilon}$ for any constant $\epsilon > 0$, unless $\mathbf{P} = \mathbf{NP}$. We also establish that NMIN-FPE is **W[1]**-hard, with the parameter being the Hamming weight of a fixed point. Under standard hypotheses in computational complexity, our results rule out the possibility of obtaining efficient approximation algorithms with provable performance guarantees and fixed parameter tractable algorithms w.r.t. the Hamming weight for NMIN-FPE.

Theorem 4.1. The problem NMIN-FPE **cannot** be approximated to within a factor $n^{1-\epsilon}$ for any constant $\epsilon > 0$, unless $\mathbf{P} = \mathbf{NP}$. This inapproximability holds even when the underlying graph is bipartite.

Proof (sketch). The overall scheme is a reduction from MINIMUM VERTEX COVER (MVC) such that if there exists a polynomial time factor $n^{1-\epsilon}$ approximation algorithm $\mathcal{A}$ for NMIN-FPE, we then can use $\mathcal{A}$ to solve MVC in polynomial time, implying $\mathbf{P} = \mathbf{NP}$. Let $\mathcal{M} = \langle G_{\mathcal{M}}, k \rangle$ be an arbitrary instance of MVC, where $n_{\mathcal{M}} = |V(G_{\mathcal{M}})|$, $m_{\mathcal{M}} = |E(G_{\mathcal{M}})|$, and the target vertex cover size is k.

We build an instance $\mathcal{S} = (G_{\mathcal{S}}, \mathcal{F})$ of NMIN-FPE for which $n_{\mathcal{S}} = |V(G_{\mathcal{S}})|$ and $m_{\mathcal{S}} = |E(G_{\mathcal{S}})|$. Let $\alpha = m_{\mathcal{M}} + n_{\mathcal{M}} + 1$, and $\beta = \alpha^{\lceil 2/\epsilon \rceil}$. The construction is as follows.

The vertex set $V(G_{\mathcal{S}})$: Let $X = \{x_u : u \in V(G_{\mathcal{M}})\}$ and $Y = \{y_e : e \in E(G_{\mathcal{M}})\}$ be two disjoint sets of nodes in $G_{\mathcal{S}}$ that correspond to nodes and edges in $G_{\mathcal{M}}$, respectively. Let w, z be two additional nodes. Lastly, we introduce a set of β nodes $R = \{r_1, ..., r_\beta\}$.

The edge set $E(G_{\mathcal{S}})$: Let $E_1 = \{(x_u, y_e) : e \in E(G_{\mathcal{M}})$ is incident on $u \in V(G_{\mathcal{M}})\}$; thus, if an edge e is incident on a node u in $G_{\mathcal{M}}$, their corresponding nodes y_e and x_u are adjacent in $G_{\mathcal{S}}$. Let $E_2 = \{(y_e, z) : y_e \in Y\}$; that is, node z is adjacent to all $y_e \in Y$. Let $E_3 = \{(x_u, w) : x_u \in X\}$; E_3 ensures that w is adjacent to all $x_u \in X$. Let $E_4 = \{(r_i, r_{i+1}) : r_i \in R, 1 \leq i \leq \beta - 1\}$; that is, nodes in R form a simple path. Lastly, we introduce an additional edge (w, r_1) to connect w with an endpoint of the above path. Thus, the edge set $E(G_{\mathcal{S}}) = E_1 \cup E_2 \cup E_3 \cup E_4 \cup \{(w, r_1)\}$.

Thresholds: Let $\tau_{x_u} = deg(u) + 1, \forall x_u \in X$; $\tau_{y_e} = 3, \forall y_e \in Y$; $\tau_w = k + 1$, $\tau_z = m_{\mathcal{M}} + 1$; $\tau_{r_i} = 1, \forall \tau_{r_i} \in R$.

We establish that $G_{\mathcal{M}}$ has a vertex cover of size at most k if and only if Algorithm $\mathcal{A}$ returns a nontrivial fixed point for $\mathcal{S}$ with Hamming weight at most $n_{\mathcal{S}}^{1-\epsilon}\alpha$. A detailed proof appears in the full version. □

Theorem 4.1 points out that for NMIN-FPE, even obtaining an approximation guarantee that is slightly better than a linear factor is hard.

Parameterized Complexity of NMIN-FPE. Next, we examine whether NMIN-FPE is fixed-parameter tractable (FPT) w.r.t. a *natural* structural parameter of the problem, namely the Hamming weight of a fixed point.

Theorem 4.2. The problem NMIN-FPE is **W[1]-hard** w.r.t. to the natural parameter (i.e., the Hamming weight of a fixed point) for (BOOL, THRESH)-SyDSs.

Theorem 4.2, whose proof is in the full version, implies that NMIN-FPE is not FPT w.r.t. to the natural parameter. Note that this does not exclude FPT results for other parameters, as we identify one such parameter in the next section.

5 Approaches for Solving NMIN-FPE

In this section, we consider several approaches for tackling the hardness of NMIN-FPE. We start by identifying special cases where the problem can be solved efficiently. We also present an integer linear programming (ILP) formulation that can be used to obtain optimal solutions for networks of reasonable sizes. Then we introduce a heuristic framework for NMIN-FPE that is useful in obtaining good (but not necessarily optimal) solutions in larger networks.

5.1 Efficient algorithms for special classes

Restricted classes. We identify four special classes of problem instances where NMIN-FPE can be solved in polynomial time. Motivated by real-world scenarios, we consider the classic *progressive threshold model* (Kempe, Kleinberg, and Tardos 2003), where once a vertex changes to state 1, it retains the state 1 for all subsequent time-steps. Further, we investigate the problem on special graph classes.

Theorem 5.1. For (BOOL, THRESH)-SyDSs, NMIN-FPE admits a polynomial time algorithm for any of the following restricted cases:

1. The SyDS uses the progressive threshold model.
2. The underlying graph is a directed acyclic graph.
3. The underlying graph is a complete graph.
4. There exists at least one constant-1 node.

Proof (sketch). We provide proof sketches for the first two cases. Detailed proofs for all the cases appear in the full version. Let $\mathcal{S} = (G_{\mathcal{S}}, \mathcal{F})$ be a (BOOL, THRESH)-SyDS.

Algorithm 1: NMIN-FPE_DAG($\mathcal{S}$)

Input : A (BOOL, THRESH)-SyDS $\mathcal{S} = (G_{\mathcal{S}}, \mathcal{F})$; $G_{\mathcal{S}}$ is a DAG
Output: A non-trivial minimum fixed point C^*

```
V' = {v : v ∈ V(G_S), τ_v = 1}
min_h = |V(G_S)| + 1
for each v ∈ V' do
    C_v ← the configuration where only v is in state 1
    C*_v ← the fixed point of S reached from C_v
    if H(C*_v) < min_h then
        min_h = H(C*_v)
        C* = C*_v
    end
end
return C*
```

Part 1: *Under the progressive threshold model,* NMIN-FPE *can be solved in $O(mn + n^2)$ time.*

The algorithm consists of n iterations where $n = |V(G_{\mathcal{S}})|$. At the ith iteration, $1 \le i \le n$, we construct an initial configuration C_i by setting node v_i to state 1, and all other nodes to state 0. We then evolve the system from C_i. Due to the progressive model, $\mathcal{S}$ always reaches some fixed point C_i^*. By repeating the above process for n iterations, we get a collection of n fixed points $\mathcal{C} = \{C_1^*, C_2^*, ..., C_n^*\}$. Again because of the progressive model, all n fixed points are non-trivial. It can be seen that a fixed point with minimum Hamming weight in $\mathcal{C}$ is an optimal solution. With a simple data structure, the fixed point reached from an initial configuration can be found in $O(m + n)$ time. Thus, the overall running time of the algorithm is $O(mn + n^2)$.

Part 2: NMIN-FPE *can be solved in $O(mn+n^2)$ time when $G_{\mathcal{S}}$ is a directed acyclic graph.*

We first establish that if all nodes have thresholds greater than 1, then there are **no** nontrivial fixed points for $\mathcal{S}$. Thus, suppose that there is at least one node whose threshold is 1. Our algorithm for finding a nontrivial minimum fixed point is shown in Algorithm 1. It can be shown that for each initial configuration, the fixed point reached by $\mathcal{S}$ can be computed in $O(m+n)$ time. So, the running time is $O(mn+n^2)$. ☐

Fixed parameter tractability. We further extend the solvability of NMIN-FPE and establish that NMIN-FPE is fixed parameter tractable w.r.t. the number of nodes with thresholds greater than 1.

Theorem 5.2. For (BOOL, THRESH)-SyDSs, NMIN-FPE is fixed parameter tractable w.r.t. the parameter k which is the number of nodes with thresholds greater than 1.

Specifically, we develop a quadratic kernelization algorithm that finds a nontrivial minimum fixed point in time $O(2^k + n + m)$. A detailed proof appears in the full version.

5.2 Solving NMIN-FPE in general networks

An ILP formulation. Given a (BOOL, THRESH)-SyDS $\mathcal{S} = (G_{\mathcal{S}}, \mathcal{F})$, our ILP solves the NMIN-FPE problem by constructing a nonempty minimum-cardinality subset $V^* \subseteq V(G_{\mathcal{S}})$ of nodes to set to state 1, and all nodes not in V^* to state 0, such that the resulting configuration is a fixed point of $\mathcal{S}$. Let $x_v,\ v \in V(G_{\mathcal{S}})$, be a binary variable where $x_v = 1$ if and only if node $v \in V^*$. Let $\Delta = max\_deg(G) + 2$ denote a constant that is greater than the maximum degree of G. Let $N(v),\ v \in V(G_{\mathcal{S}})$ denote the closed neighborhood of v. Then an ILP formulation for NMIN-FPE is defined as follows in (1). An optimal solution to the ILP yields a set of state-1 nodes in a nontrivial minimum fixed point. See full version for proof of correctness.

$$\min \quad \sum_{v \in V(G_{\mathcal{S}})} x_v \tag{1a}$$

$$\text{s.t.} \quad \tau_v x_v \le \sum_{u \in N(v)} x_u, \quad \forall v \in V(G_{\mathcal{S}}) \tag{1b}$$

$$\sum_{v \in V(G_{\mathcal{S}})} x_v \ge 1 \tag{1c}$$

$$\Delta \cdot x_v + \tau_v \ge \sum_{u \in N(v)} x_u + 1, \quad \forall v \in V(G_{\mathcal{S}}) \tag{1d}$$

$$x_v \in \{0, 1\}, \quad \forall v \in V(G_{\mathcal{S}}) \tag{1e}$$

The greedy framework for heuristics. As we have shown that NMIN-FPE is hard to approximate, one can only rely on heuristics to quickly find solutions for large networks. Given a (BOOL, THRESH)-SyDS $\mathcal{S} = (G_{\mathcal{S}}, \mathcal{F})$, we propose a general framework that iteratively constructs a fixed point by greedily setting nodes to state 1. Specifically, the framework iterates over each node $u \in V(G_{\mathcal{S}})$ and finds a fixed point seeded at u, which is denoted as C_u. In C_u, we have (1) u is in state 1, and (2) the Hamming weight of C_u is heuristically small. The algorithm then returns a fixed point C_u with the smallest Hamming weight over all $u \in V(G_{\mathcal{S}})$ as shown in Algorithm 2.

Algorithm 2: `NM1FPE`$(\mathcal{S})$

Input : SyDS $\mathcal{S} = (G_{\mathcal{S}}, \mathcal{F})$
Output: A fixed point $C^* \in \{0,1\}^n$
$best\_obj = |V(G_{\mathcal{S}})|$
for $u \in V(G)$ **do**
  C_u = `Greedy_Seeded_NM1FPE`$(\mathcal{S},\ u)$
  if $H(C_u) < best\_obj$ **then**
    $best\_obj = H(C_u)$
    $C^* = C_u$
  end
end
return C^*

We now introduce the greedy scheme to compute $C_u, u \in V(G_{\mathcal{S}})$. Let $A_u \subseteq V(G_{\mathcal{S}})$ be a subset of state 1 nodes under C_u. The scheme constructs C_u by progressively adding nodes to A_u, thus effectively setting them to state 1. Overall, the scheme has the following two key steps: (1) the construction of A_u terminates when the *fixed point condition* is met. Specifically, (i) for each $v \in A_u$, the threshold τ_v is satisfied, that is, the number of v's neighbors (including v itself) in A_u is at least τ_v, and (ii) for each $v \in V(G_{\mathcal{S}}) \setminus A_u$, the number of v's neighbors in A_u is less than τ_v; (2) nodes are added to A_u based on a *greedy selection method* specified by a heuristic. Under this framework, different heuristics can be obtained by using different greedy selection strategies.

The fixed point condition. We use a superscript to denote the iteration number. Initially, $A_u^{(0)}$ contains only u, and we actively add a new node to $A_u^{(k)}$ in each iteration $k \geq 1$. To determine if the fixed point condition is met at the kth iteration, we maintain a value $\delta^{(k)}$ that is the number of *additional nodes* that need to be selected to satisfy the thresholds of nodes in $A_u^{(k)}$ at the kth iteration. Heuristically, $\delta^{(k)} = \sum_{v \in A_u^{(k)}} \tilde{\tau}_v^{(k)}$ where $\tilde{\tau}_v^{(k)} = \max\{0, \tau_v - |A_u^{(k)} \cap N(v)|\}$ is the *residual threshold* of v at the kth iteration. We can view $\tilde{\tau}_v^{(k)}$ as the additional number of v's neighbors that need to be selected to satisfy τ_v.

Given a node $w \in A_u^{(k)}$, we call w *unsatisfied* if $\tilde{\tau}_w \neq 0$. Note that after adding a node v to $A_u^{(k+1)}$, v decreases the residual thresholds of its unsatisfied neighbors in $A_u^{(k)}$; yet, v may remain unsatisfied. Furthermore, we might "passively" satisfy the thresholds of some other nodes that are not in $A_u^{(k)}$. Let A' denote such a set of "passive" nodes. We define $\epsilon_v^{(k)}$ to be the decrease of $\delta^{(k)}$ after selecting v and A'. Thus, $\delta^{(k+1)}$ for the next iteration is computed by $\delta^{(k+1)} = \delta^{(k)} + \tilde{\tau}_v - \epsilon_v^{(k)}$. Lastly, $A_u^{(k+1)} = A_u^{(k)} \cup \{v\} \cup A'$. We have the following proposition.

Proposition 5.3. The fixed point condition is met at the kth iteration of the algorithm if and only if $\delta^{(k)} = 0$.

The subroutine returns the fixed point C_u where a node v is in state 1 iff $v \in A_u$. The pseudocode of the entire framework is presented under Algorithms 2 and 3.

Algorithm 3: `Greedy_Seeded_NM1FPE`$(\mathcal{S}, u)$

Input : SyDS $\mathcal{S} = (G_{\mathcal{S}}, \mathcal{F})$; node $u \in V(G_{\mathcal{S}})$
Output: A fixed point C_u seeded at node u
$A' \leftarrow$ The set of nodes being passively set to state 1 if $s(u) = 1$
$A_u \leftarrow \{u\} \cup A'$
$B \leftarrow \{N(u) \setminus A_u\}$ ▷ `Candidate nodes`
$D = \{v \in A_u : \tilde{\tau}_v \neq 0\}$ ▷ `Unsatisfied nodes`
$\delta = \max\{0, \tau_u - |A_u \cap N(u)|\}$
while $\delta \neq 0$ **do**
  $v^* = \arg\min_{v \in B}\{obj(v)\}$ ▷ `Greedy selection`
  $\delta = \delta + \tilde{\tau}_{v^*} - \epsilon_{v^*}$
  $A' \leftarrow$ The set of nodes being passively set to state 1 after selecting v^*
  $A_u = A_u \cup \{v^*\} \cup A'$
  Update sets B and D
end
$C_u \leftarrow$ the configuration with $v \in V(G_{\mathcal{S}})$ in state 1 iff $v \in A_u$
return C_v

Greedy selection strategies. We now discuss a methodology for adding a new node v to $A_u^{(k)}$. Observe that under any nontrivial minimum fixed point, the subgraph induced by the state-1 nodes is *connected*, assuming the absence of constant-1 nodes. Thus, only the unselected neighbors of unsatisfied nodes, denoted by $B^{(k)}$, are candidate nodes at the kth iteration. Specifically, we greedily select a node $v = \arg\min_{v \in B}\{obj(v)\}$ into $A_u^{(k)}$ where $obj(v)$ is an objective function specified by some heuristic. We now present objective functions for three heuristics. Let $\rho_v^{(k)}$ denote the number of unselected nodes that will be passively set to state 1 if v is set to state 1 at the kth iteration. The objective for the first heuristic `GreedyFull` is defined as $obj_1(v) = \tilde{\tau}_v^{(k)} + \rho_v^{(k)} - \epsilon_v^{(k)}$ which considers v's residual thresholds, the number of passive nodes, and the decrease of $\delta^{(k)}$. The objective functions of two other methods, namely `GreedyNP` and `GreedyThresh`, are simplifications of the first heuristic with $obj_2(v) = \tilde{\tau}_v^{(k)} - \epsilon_v^{(k)}$ and $obj_3(v) = \tilde{\tau}_v^{(k)}$, respectively. To speed up the execution time, we use *pruning* in the implementation of heuristics. Specifically, for each node u enumerated in the heuristic framework, we keep track of the current best Hamming weight and actively terminate construction of C_u if the accumulated Hamming weight of C_u is larger than the current best value. In addition, we examine nodes in ascending order of their threshold values, thus heuristically attempting to find a small fixed point faster. To further simplify the

GreedyFull algorithm, we propose GreedySub, which only examines the seeded fixed points for nodes v if they have never been set to state 1 under C_u for each node u that is examined in previous iterations.

As shown in the full version, the time complexity of the proposed framework is $O(n^2 \cdot q + n^2 m)$ where q is the runtime of a single greedy selection process. Subsequently, the time complexities of GreedyNP/Thresh and GreedySub/Full are $O(n^2 m)$ and $O(n^3 m)$ respectively.

6 Experimental Results

We conduct extensive experiments to investigate the performance of the heuristics across several scenarios. Overall, our results demonstrate high effectiveness of the heuristics in real-world networks.

6.1 Experimental setup

Datasets. We select the networks based on their sizes, diversity and application areas. Overall, we evaluate the heuristics on 13 real-world networks from various domains (listed in Table 1) and on Erdös Rényi (Gnp) random networks.

Dataset	Type	n	m	*Max deg*
router	Infrastructure	2,113	6,632	109
power	Infrastructure	5,300	8,271	19
twitch	Social	7,126	35,324	720
retweet	Social	7,252	8,060	1,884
lastfm	Social	7,624	27,806	216
arena	Social	10,680	24,316	205
gnutella	Peer-to-Peer	10,876	39,994	103
auto	Infrastructure	11,370	22,002	2,312
astroph	Coauthor	17,903	196,972	504
condmat	Coauthor	21,363	91,286	279
facebook	Social	22,470	170,823	709
google+	Social	23,613	39,182	2,761
Deezer	Social	28,281	92,752	172

Table 1: List of networks

Heuristics and baselines. We evaluate the performance of the proposed greedy heuristics by comparing with the following baselines: (1) DegDis: a minimization version of the selection method proposed in (Chen, Wang, and Yang 2009); (2) Random: select nodes randomly. We also consider other methods that select nodes with a smallest value of the metrics: (3) Pagerank, (4) Distance (closeness centrality) which are also widely used by others as baselines (Yao et al. 2015; Kempe, Kleinberg, and Tardos 2003).

Experimental scenarios. We consider the following three cases in investigating the effectiveness of the above heuristics. (1) *Random thresholds*, (2) *Uniform thresholds*, and (3) *Gnp networks with increasing sizes*. The details of each setting are given in later sections.

Evaluation metric. We use the *approximation ratio* $\gamma = obj/OPT$ as the evaluation metric, where OPT is the optimal objective (i.e., the Hamming weight of a minimum nontrivial fixed point) of a problem instance, computed by solving the proposed ILP using the Gurobi solver (Gurobi Optimization), and *obj* is the objective value returned by a heuristic. We remark that $\gamma \geq 1$ and *an algorithm with a lower* γ *gives a better solution*.

Machine and reproducibility. All experiments were performed on Intel Xeon(R) Linux machines with 64GB of RAM. Our source code (in C++ and Python), documentation, and selected datasets are provided in the GitHub.

6.2 Experimental results

We summarize the results produced by the heuristics under the three experimental scenarios mentioned earlier.

Random thresholds. We first study the scenario where a threshold value is assigned to each node v randomly from the range $[3, deg(v) + 1]$. This assignment guarantees that there are no *constant*-1 nodes as the NMIN-FPE would become efficiently solvable in that case by Theorem 5.1. The random threshold assignment is a way to cope with the incomplete knowledge of the actual threshold values of nodes (Kempe, Kleinberg, and Tardos 2003).

The results, averaged over 10 initializations of threshold assignments, are shown in Fig. 2 **under the** $\log_{10}$ **scale**. Overall, we observe that the proposed Greedy family significantly outperforms other baselines. Specifically, the averaged approximation ratios (over all networks) of GreedyFull/NP/Sub are less than 3, which is over 10 times better than those of baselines on most networks. Within the Greedy family, the simplest heuristic GreedyThresh shows the lowest performance, with an averaged approximation ratio of 4.34. Nevertheless, we remark that this empirically constant ratio is significantly better than that of other baselines. Further, GreedyThresh, which is the most efficient among all its counterparts, finds solutions in minutes for most of the selected networks.

Uniform thresholds. We assign all nodes the same threshold $\tau > 3$. We consider different values of the uniform threshold τ to study how the heuristics perform. Note that as τ increases, more constant-0 nodes emerge in the system because their threshold values are larger than their degrees. Thus, the uniform-threshold setting pushes the performance limits of algorithms by (1) setting the thresholds of nodes to be indistinguishable and (2) introducing a considerable number of constant-0 nodes which makes it harder for the heuristics to even find a feasible solution.

We first present analyses on instances of the same network with different uniform thresholds τ. Due to page limits, we show results for the Google+ network in Fig. 3; the results are similar for all other networks where the Greedy family outperforms the other heuristics in terms of approximation ratios. In particular, the averaged ratios for heuristics in the Greedy family are all less than 3 for Google+ network, with GreedyThresh having the highest averaged ratio (lower is better) of 2.41. We also observe when the uniform threshold τ is large enough, many natural heuristics failed to even find a valid solution due to the presence of a large number of constant-0 nodes. This experiment demonstrates the high effectiveness of the proposed framework when nodes have indistinguishable thresholds.

Next, we fix the uniform threshold τ and analyze the heuristics across all networks. We have investigated different values of τ from 3 to 20, where the Greedy family again

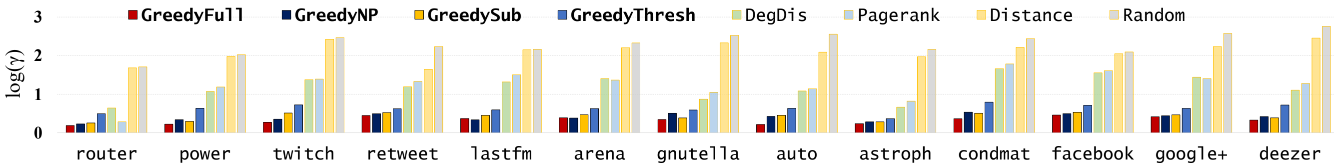

Figure 2: **The approximation ratios (lower is better) under random threshold setting.** The y axis denotes the approximation ratios of heuristics under the $\log_{10}$ scale. The results are averaged over 10 initializations of threshold assignments.

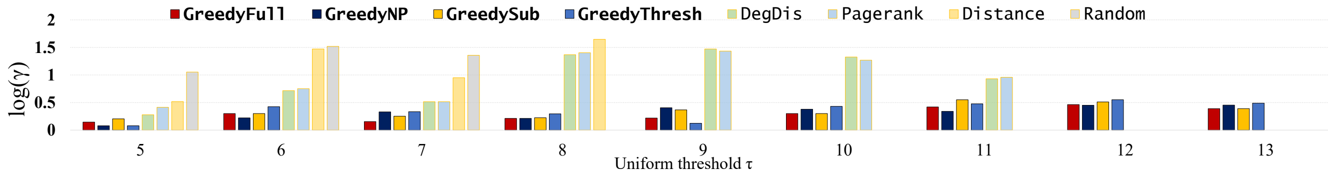

Figure 3: **The approximation ratios (lower is better) of heuristics on `google+` network under uniform threshold setting.** The y axis shows the approximation ratios of heuristics under the $\log_{10}$ scale. Results of several heuristics (`DegDis`, `Pagerank`, `Distance`, `Random`) are absent for some instances because they failed to find valid fixed points.

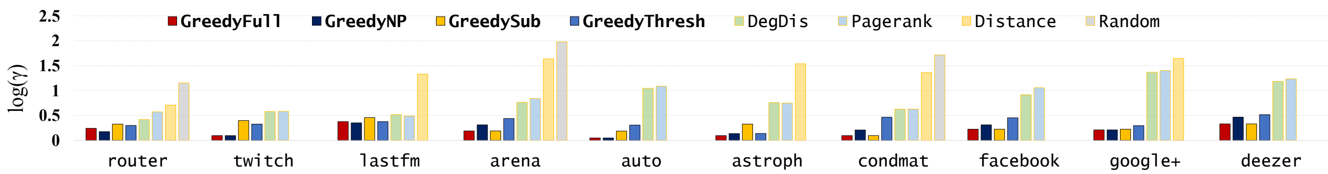

Figure 4: **The approximation ratios (lower is better) of heuristics on networks under uniform threshold setting where $\tau = 8$.** The y shows the approximation ratios of heuristics under the $\log_{10}$ scale. Results of several heuristics (`DegDis`, `Pagerank`, `Distance`, `Random`) are absent for some instances because they failed to find valid fixed points.

outperforms the other heuristics on most instances, producing results that are over 10 times better than baselines. Due to the page limit, we show the results for $\tau = 8$ in Fig. 4. Note we omit the results for networks `power` and `peer` because they have no feasible solutions when $\tau = 8$.

Gnp networks. We study the heuristics on Erdös Rényi networks with sizes up to $1,000$ where thresholds are assigned randomly. We observe that the ratios of `GreedyFull/NP/Thresh` are all over 40, and the ratios of `GreedySub` are as high as 148.48. We remark that in a Gnp network, the state-1 nodes in a nontrivial minimum fixed point are often surrounded by nodes with similar thresholds and similar degrees. Our results suggest a limitation of the proposed framework; that is, when networks exhibit uniformity at the node level, the heuristics might not correctly choose state-1 nodes to construct a minimum fixed point. Note that such results are expected since NMIN-FPE is hard to approximate. We remark that the `Greedy` family still outperforms baselines on Gnp networks.

Efficiency. Our results also demonstrate that the proposed `Greedy NP/Sub/Thresh` are more efficient than the ILP solver `Gurobi` for the tested scenarios. In Table 2, we show the execution times of `Greedy NP/Sub/Thresh` and `Gurobi` solver on the two largest networks under the random threshold scenario.

We remark that the heuristics achieve high efficiency due to the pruning technique. For some networks, `Gurobi` runs comparably fast, usually within 5 minutes. Nevertheless, `Gurobi` uses parallelization mechanisms such as multithreading (over 30 threads are used on each instance) whereas our heuristics achieve high efficiency while executing in serial mode.

Network	GreedyNP	GreedySub	GreedyThresh	Gurobi
Google+	**26.13s**	263.64s	32.16s	1153.04s
Deezer	104.76s	395.89s	**96.22s**	794.89s

Table 2: Execution times of heuristics in seconds.

7 Conclusions and Future Work

In this paper, we study the NMIN-FPE problem from both the theoretical and empirical points of view. We establish the computational hardness of the problem and propose effective algorithms for solving NMIN-FPE under special cases and for general graphs. Our results point to a new way of quantifying system resilience against the diffusion of negative contagions and a new approach to tackle the influence minimization problem. One limitation of the proposed heuristics is the time complexity. Thus, a future direction is to develop more efficient heuristics for solving NMIN-FPE. Another promising direction is to approximate NMIN-FPE for restricted graph classes such as regular graphs. Lastly, we plan to extend the model to multi-layer networks and investigate problems in this new domain.

Acknowledgements

We thank the anonymous reviewers for reading the paper carefully and providing helpful feedback. This work is partially supported by University of Virginia Strategic Investment Fund award number SIF160 and NSF Grants OAC-1916805 (CINES), CCF-1918656 (Expeditions), DTRA subcontract/ARA S-D00189-15-TO-01-UVA and NSF IIS-1908530.

Ethical Impact

The work reported in this paper addresses some complexity and algorithmic issues associated with discrete dynamical systems that serve as formal models for contagion propagation in social networks. The formal results are in the form of theorems and algorithms that are useful in understanding the effectiveness of dynamical system models. All the experimental results presented in the paper use public domain networks.

References

Adam, E. M.; Dahleh, M. A.; and Ozdaglar, A. 2012. On the behavior of threshold models over finite networks. In *Proc. 51st IEEE Conference on Decision and Control (CDC)*, 2672–2677. IEEE.

Anderson, S. P.; Goeree, J. K.; and Holt, C. A. 2001. Minimum-effort coordination games: Stochastic potential and logit equilibrium. *Games and Economic Behavior*, 34(2): 177–199.

Auletta, V.; Ferraioli, D.; and Greco, G. 2018. Reasoning about consensus when opinions diffuse through majority dynamics. In *Proc. International Joint Conference on Artificial Intelligence (IJCAI)*, 49–55.

Barrett, C.; Hunt III, H. B.; Marathe, M. V.; Ravi, S. S.; Rosenkrantz, D. J.; Stearns, R. E.; and Thakur, M. 2007. Predecessor existence problems for finite discrete dynamical systems. *Theoretical Computer Science*, 386(1): 3–37.

Barrett, C. L.; Hunt III, H. B.; Marathe, M. V.; Ravi, S.; Rosenkrantz, D. J.; and Stearns, R. E. 2006. Complexity of reachability problems for finite discrete dynamical systems. *Journal of Computer and System Sciences*, 72(8): 1317–1345.

Cao, H.; Ertin, E.; and Arora, A. 2008. MiniMax equilibrium of networked differential games. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 3(4): 1–21.

Chen, C.; Tong, H.; Prakash, B. A.; Eliassi-Rad, T.; Faloutsos, M.; and Faloutsos, C. 2016. Eigen-optimization on large graphs by edge manipulation. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 10(4): 1–30.

Chen, C.; Tong, H.; Prakash, B. A.; Tsourakakis, C. E.; Eliassi-Rad, T.; Faloutsos, C.; and Chau, D. H. 2015. Node immunization on large graphs: Theory and algorithms. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 28(1): 113–126.

Chen, W.; Wang, Y.; and Yang, S. 2009. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD International conference on Knowledge Discovery and Data Mining (KDD)*, 199–208.

Cheng, J.; Kleinberg, J.; Leskovec, J.; Liben-Nowell, D.; Subbian, K.; Adamic, L.; et al. 2018. Do diffusion protocols govern cascade growth? In *Proceedings of the International AAAI Conference on Web and Social Media (ICWSM)*, volume 12, 32–41.

Chistikov, D.; Lisowski, G.; Paterson, M.; and Turrini, P. 2020. Convergence of opinion diffusion is PSPACE-complete. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, 7103–7110.

Daskalakis, C.; and Papadimitriou, C. H. 2007. Computing equilibria in anonymous games. In *48th Annual IEEE FOCS*, 83–93. IEEE.

Daskalakis, C.; and Papadimitriou, C. H. 2015. Approximate Nash equilibria in anonymous games. *Journal of Economic Theory*, 156: 207–245.

Dong, M.; Zheng, B.; Quoc Viet Hung, N.; Su, H.; and Li, G. 2019. Multiple rumor source detection with graph convolutional networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*, 569–578.

Goles, E.; and Martínez, S. 2013. *Neural and automata networks: dynamical behavior and applications*, volume 58. Springer Science & Business Media.

Granovetter, M. 1978. Threshold models of collective behavior. *American Journal of Sociology*, 83(6): 1420–1443.

Gurobi Optimization. 2021. Reference Manual. https://www.gurobi.com/documentation/9.1/refman/index.html.

Jackson, M. O. 2010. *Social and economic networks*. Princeton, NJ: Princeton University Press.

Kempe, D.; Kleinberg, J.; and Tardos, É. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 137–146.

Khalil, E.; Dilkina, B.; and Song, L. 2013. CuttingEdge: influence minimization in networks. In *Proceedings of Workshop on Frontiers of Network Analysis: Methods, Models, and Applications at NIPS*, 1–13. Citeseer.

Kimura, M.; Saito, K.; and Motoda, H. 2008. Minimizing the spread of contamination by blocking links in a network. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 8, 1175–1180.

Kimura, M.; Saito, K.; and Nakano, R. 2007. Extracting influential nodes for information diffusion on a social network. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 7, 1371–1376.

Kuhlman, C. J.; Kumar, V. S. A.; Marathe, M. V.; Ravi, S. S.; and Rosenkrantz, D. J. 2015. Inhibiting diffusion of complex contagions in social networks: theoretical and experimental results. *Data Mining and Knowledge Discovery.*, 29(2): 423–465.

Kuhlman, C. J.; Tuli, G.; Swarup, S.; Marathe, M. V.; and Ravi, S. S. 2013. Blocking simple and complex contagion by edge removal. In Xiong, H.; Karypis, G.; Thuraisingham, B. M.; Cook, D. J.; and Wu, X., eds., *2013 IEEE 13th International Conference on Data Mining, Dallas, TX, USA, December 7-10, 2013*, 399–408. IEEE Computer Society.

Ramazi, P.; Riehl, J.; and Cao, M. 2016. Networks of conforming or nonconforming individuals tend to reach satisfactory decisions. *Proceedings of the National Academy of Sciences*, 113(46): 12985–12990.

Rosenkrantz, D. J.; Marathe, M. V.; Ravi, S. S.; and Stearns, R. E. 2021. Synchronous dynamical systems on directed acyclic graphs (DAGs): Complexity and algorithms. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 11334–113432.

Salehisadaghiani, F.; and Pavel, L. 2018. Distributed Nash equilibrium seeking in networked graphical games. *Automatica*, 87: 17–24.

Valdez, L. D.; Shekhtman, L.; La Rocca, C. E.; Zhang, X.; Buldyrev, S. V.; Trunfio, P. A.; Braunstein, L. A.; and Havlin, S. 2020. Cascading failures in complex networks. *Journal of Complex Networks*, 8(2): 1–23.

Wang, B.; Chen, G.; Fu, L.; Song, L.; and Wang, X. 2017. DRIMUX: Dynamic rumor influence minimization with user experience in social networks. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 29(10): 2168–2181.

Willis, D. E.; Andersen, J. A.; Bryant-Moore, K.; Selig, J. P.; Long, C. R.; Felix, H. C.; Curran, G. M.; and McElfish, P. A. 2021. COVID-19 vaccine hesitancy: Race/ethnicity, trust, and fear. *Clinical and Translational Science*, 14(6): 2200–2207.

Wolfe, R. M.; and Sharp, L. K. 2002. Anti-vaccinationists past and present. *British Medical Journal (BMJ)*, 325(7361): 430–432.

Yang, L.; Li, Z.; and Giua, A. 2019. Influence minimization in linear threshold networks. *Automatica*, 100: 10–16.

Yao, Q.; Shi, R.; Zhou, C.; Wang, P.; and Guo, L. 2015. Topic-aware social influence minimization. In *Proceedings of the 24th International Conference on World Wide Web*, 139–140.

Ye, M.; Zino, L.; Mlakar, Ž.; Bolderdijk, J. W.; Risselada, H.; Fennis, B. M.; and Cao, M. 2021. Collective patterns of social diffusion are shaped by individual inertia and trend-seeking. *Nature Communications*, 12(1): 1–12.

Yu, S.; Zhou, K.; Brantingham, J.; and Vorobeychik, Y. 2020. Computing equilibria in binary networked public goods games. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, 2310–2317.

Zhu, J.; Ni, P.; and Wang, G. 2020. Activity minimization of misinformation influence in online social networks. *IEEE Transactions on Computational Social Systems*, 7(4): 897–906.