Control of Learning in Anticoordination Network Games

Ceyhun Eksin, *Member, IEEE*, and Keith Paarporn, *Member, IEEE*

***Abstract*—Many games have undesirable Nash equilibria. In such games, a designer's goal is to avoid "bad" equilibria. In this article, we focus on which players to control and how to control them so that the emerging outcome of learning dynamics is desirable. In particular, we consider best-response-type learning dynamics for an anticoordination network game. The designer's goal is to achieve maximum anticoordination with the fewest number of players to control at each round. Our analysis shows that despite the incentive to anticoordinate with neighbors, selfish agents may fail to do so. Accordingly, we relate optimal policies for obtaining maximum anticoordination to the set of Nash equilibria. Noting the combinatorial problem of optimally selecting players in benchmark networks, we develop suboptimal solutions via solving a minimum vertex-cover problem, and by greedily selecting players based on their potential to induce cascading effects. Numerical experiments on random networks show that the cascade-based greedy algorithm can lower the control effort significantly compared to random public advertising policies. Moreover, its control effort is no more than twice the optimal control effort in the worst case.**

***Index Terms*—Game theory, networked control systems, network security.**

I. Introduction

DEVELOPING methods that achieve globally optimal behavior while conforming with the computational and informational limitations of the players is of interest, given the ubiquity of noncooperative interactions that arise among actors in networked systems, e.g., epidemics [1]; energy [2], [3]; security [4]; communication [5]; or autonomous systems [6]. Game theoretic learning algorithms are tractable decentralized models for noncooperative decision-making in networked systems. These algorithms accounting for local information access [7] and coupled action spaces [8] guarantee convergence to individually rational behavior, i.e., a Nash equilibrium (NE) action, in certain classes of games, e.g., aggregative [9], [10]; potential [11]; and convex [12]. However, a NE, while being optimal from the perspective of selfish individuals, can be inefficient and undesired at the system level. A canonical example of this is the tragedy of the commons which describes the phenomenon of selfish learning behavior leading to the worst possible outcome for the entire population [13]. Given the possibility of emergence of undesired outcomes, there is a need to develop incentive mechanisms in order to achieve system-wide desired outcomes.

The major challenge in attaining globally desired outcomes in networked systems is that individuals are selfish, heterogeneous, and their actions are coupled while the centralized incentive resources are costly. In this article, we formulate this challenge as the control of decentralized learning dynamics. That is, players selfishly follow some game-theoretic learning dynamics while a centralized authority aims to direct the emergent behavior toward a desired outcome. The two main issues we address with this formulation are i) the selection of which players to control, and ii) what control policy to implement given the selected players.

This challenge is addressed in the literature by characterizing the inefficiency of Nash equilibria [14], by designing payoffs prior to the start of the game to induce efficient Nash equilibria [15], or by developing control mechanisms [16]–[23]. Our approach falls into the last category of controlling players to guide the learning dynamics toward desirable outcomes. In this category, [16] and [17] show a public advertising scheme which improves the efficiency of the emergent outcome for players that act according to best-response dynamics and occasionally listen to the advertised behavior in party affiliation or cut games. An alternative model designs dynamic control incentives that affect every players' payoffs to which players best respond [19]. Uniform and targeted reward policies that induce coordination among players acting according to best-response dynamics in a network coordination game are developed in [20]. When the goal is to minimize efficiency, [23] studies malicious attacks that strategically perturb player learning dynamics in a network coordination game. We depart from these studies in two ways: 1) our goal is to maximize anticoordination instead of social welfare, and 2) we consider which players to control and how to control them.

In particular, we focus on controlling a subset of the players in an anticoordination network game. The aim is to promote maximum anticoordination (MAC)—when all players differentiate their actions from their neighbors. Players belong to one of two possible types. We assume there is a preferred selfish

Manuscript received February 21, 2020; revised February 24, 2020 and February 25, 2020; accepted May 19, 2020. Date of publication June 15, 2020; date of current version December 16, 2020. The work of Ceyhun Eksin was supported in part by a grant from the NSF under Grant NSF CCF-2001529. Recommended by Associate Editor R. Jain. *(Corresponding author: Ceyhun Eksin.)*

Ceyhun Eksin is with the Industrial and Systems Engineering Department, Texas A & M University, College Station, TX 77843 USA (e-mail: eksinc@tamu.edu).

Keith Paarporn is with the Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106 USA (e-mail: kpaarporn@ucsb.edu).

Digital Object Identifier 10.1109/TCNS.2020.3002426

action for each player in the absence of any neighboring players in the network. That is, a player's payoff decreases as more of its neighbors, belonging to the opposite type, take the preferred action (see Section II). Such payoff dependencies can be used to model individual behavior during the spread of an epidemic in a population where the individual types are healthy and sick, the actions represent the level of precautionary measures taken, and payoffs capture the risk of disease transmission to healthy from sick individuals with the preferred action as, not taking any measures [24]. Other examples include modeling individual opinions in a politically polarized environment where players would like to differentiate their actions from players in opposing views [25], or modeling two competing species in an environment [26]. In these games, two neighboring players anticoordinate when at least one of the players do not take the selfish action, e.g., a player in each link takes a precautionary measure during an epidemic.

We formulate the minimum player control for anticoordination (MPCAC) problem as a mixed integer program where the decision variables include which players to control, and how to control them (see Section III). We assume players that are not controlled, follow best-response-type learning dynamics [27] (see Section III-A). We consider static and dynamic control policies. In static MPCAC, a selected subset of players' decisions are fixed for the entire learning horizon. In dynamic MPCAC, the control policy can temporally influence player decisions. We find a feasible policy, based on a minimum vertex covering of a reduced bipartite network, that upper bounds the optimal dynamic MPCAC control policy (see Theor. 1) and is computationally tractable—it can be solved by a linear program. In general, this policy is suboptimal if there exist ways to cause cascades of anticoordination among multihop neighbors via the learning dynamics by only controlling a few players. We present optimal policies on simple networks of arbitrary size (see Section IV). Some of the optimal policies use cascades to achieve anticoordination which exemplify the suboptimality of the vertex cover-based control policy. For general networks, we propose a greedy algorithm that finds a feasible solution by sequentially selecting the player to control. We propose various selection criteria including random selection, and selection with respect to the potential of the player to induce cascades of anticoordination (see Section V). We note that the random selection is akin to the public advertising model considered in [16], [17], and [36]. Cascade potential-based selection outperforms other selection criteria in numerical experiments (see Section VI). We show the greedy algorithm with cascade potential-based selection criterion is likely to match the optimal static control effort in the majority of the cases.[1]

II. Anticoordination Network Games

A game consists of a population of players $\mathcal{N}:=\{1,...,n\}$. Each player $i \in \mathcal{N}$ selects an action $a_i \in \mathcal{A}_i$ in order to maximize its utility function $u_i(\cdot) : \Pi_{j\in\mathcal{N}}\mathcal{A}_j \to \mathbb{R}$. We assume that each player is in one of two possible types $s_i \in \{0, 1\}$. Accordingly, the population is divided into two disjoint sets $\mathcal{S}_0$ and $\mathcal{S}_1$. Only the actions of neighbors that have the opposite type can affect a player's utility function. For instance, if a player i belongs to $\mathcal{S}_0$, then its utility depends on actions of its neighbors in $\mathcal{S}_1$. We can capture the payoff dependence of players using a bipartite graph $\mathcal{G}_B = (\mathcal{S}_0, \mathcal{S}_1, \mathcal{E}_B)$ (see Fig. 1). We define the neighborhood of player i as $\mathcal{N}_i := \{j \in \mathcal{N} : (i, j) \in \mathcal{E}_B\}$.

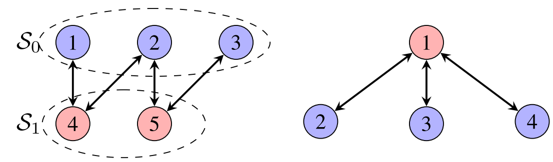


Fig. 1. Network game with binary types on line and star networks.

Given the bipartite network $\mathcal{G}_B$, the network game with binary types can be represented by the tuple $\Gamma = \{\mathcal{N}, \mathcal{A}, \mathcal{G}_B, \{u_i\}_{i\in\mathcal{N}}\}$.

The premise of an anticoordination game is that a player benefits if its opponents yield. Similarly, in an anticoordination network game, a player benefits if its neighbors in the opposing type yield. We assume player i can take actions between zero and one, i.e., $\mathcal{A}_i = [0, 1]$ for all $i \in \mathcal{N}$. The utility function

$$u_i(a_i, a_{\mathcal{N}_i}) = a_i \left(1 - (c_0(1 - s_i) + c_1 s_i) \sum_{j\in\mathcal{N}_i} a_j\right) \quad (1)$$

with $c_0 > 0$, and $c_1 > 0$ as constants, captures the preferences of players to differentiate their actions from their neighbors. Here, $a_{\mathcal{N}_i}$ are the actions of player i's neighbors. Action $a_i = 1$ maximizes the utility if the term inside the parentheses is positive. Otherwise, action $a_i = 0$ maximizes the utility. The constant 1 inside the parentheses means that the preferred action is 1 regardless of a player's type. The term that is subtracted from one captures the decrease in the preference of the player to choose action 1. That is, as player i's neighbors increase their action, the benefit of i from choosing the preferred action decreases. This decrease depends on the type of the player. If the player's type is 0 (1), i.e., $s_i = 0\ (= 1)$, then the decrease is proportional to $c_0\ (c_1)$.

Below we provide examples for the anticoordination network game Γ with payoffs as in (1).

Example 1 (Disease spread on networks): Players want to avoid disease transmission [24]. Each player is either healthy ($s_i = 0$) or sick ($s_i = 1$). The network $\mathcal{G}_B$ is a contact network with each edge representing a chance of disease transmission between a healthy and a sick player. The action space captures the social distancing level of a player with action $a_i = 0$ representing self-isolation and action $a_i = 1$ representing resuming normal activity. Actions between 0 and 1 represent different levels of disease prevention measures, e.g., covering cough, or washing hands often. Resuming normal activity is the preferred action. However, if both players at the two ends of an edge take action 1, then there is a chance of disease transmission. Accordingly, the constant c_0 captures a healthy player's sensitivity for avoiding a risky interaction. The constant c_1 captures a sick

[1]Our prior work [28] considers the static MPCAC formulation for the same anticoordination network game and individual learning rules. This article generalizes the static formulation to dynamic MPCAC providing analytical and numerical results that compare the optimal static and dynamic policies with the proposed greedy algorithms.

player's sensitivity to avoid disease transmission to one of its healthy neighbors.

Example 2 (Political polarization): Players want to differentiate their actions from those with opposing beliefs [25]. The network represents the social interactions among players in opposing beliefs ($\mathcal{S}_0$ and $\mathcal{S}_1$). Action 1 represents a monetary choice or support for a cause that is individually desirable in the absence of partisanship. A player's tendency to take the preferred action (action 1) reduces as it has more neighbors that take action 1. That is, a player can opt-out from individual benefits or societal impact to express partisan preferences. Constants c_0 and c_1 capture the inclination of players in beliefs 0 and 1 to differentiate themselves from those with opposing beliefs.

Example 3 (Hawk–Dove network game): Two species ($\mathcal{S}_0$ and $\mathcal{S}_1$) face off in an ecological environment. At each interaction players decide to be hawkish ($a_i = 1$) or dovish ($a_i = 0$). A hawk move gets the highest reward if its neighboring competitors play dove. If both interacting players play dove, they miss the opportunity to overcome their competitor. If both interacting players are hawkish, they challenge each other and face costs. The constants c_0 and c_1 represent the costs species 0 and 1 incur, respectively, when they act hawkish against a hawkish competitor.

Remark 1: The anticoordination network game with payoffs in (1) has similar incentives as network games known as party affiliation or cut games [16], [29], [30]. In cut games, a player incurs a cost of 1 for each neighbor it agrees with. Besides the above examples, the motivation for studying cut games stem from resource allocation in wireless communication settings where players that transmit over the same channel incur collisions [31]. Unlike the payoff in (1), players are indifferent between two actions if they have no neighbors in cut games. The payoff in (1) primes players to take action 1 unless enough neighbors take action 1. Moreover, in (1), two neighboring players do not incur a penalty if both players choose action 0, while in cut games no such distinction is made between actions. That is, players incur a cost if they agree regardless of the action. Having a symmetric edge weight of 1 implies the cut games are potential games [32], while the game in (1) is not necessarily a potential game unless $c_0 = c_1$. An important feature of potential games is that best response dynamics converge to a pure NE, which in our case is not guaranteed but can be shown to be also true—see Remarks 3 and 4.

III. Maximum Anticoordination Problem

Selfish behavior may result in failure of anticoordination, i.e., both players that form a link take action 1. Our goal is to find the fewest number of players to control so that anticoordination happens at every link.

We define the edge between players $(i, j) \in \mathcal{E}_B$ as *inactive* if $a_i \times a_j = 0$, with $a_i \in \{0, 1\}$ and $a_j \in \{0, 1\}$. The edge is *active* if it is not inactive, i.e., $a_i \times a_j > 0$. We obtain MAC when all edges are inactive. We denote the set of action profiles that induce MAC on a graph $\mathcal{G}$ as

$$M(\mathcal{G}) := \{a \in \mathcal{A} : (i, j) \text{ is inactive for all } (i, j) \in \mathcal{E}\}. \quad (2)$$

An equilibrium action profile a^* of the game need not be in the set $M(\mathcal{G})$. Hence, to ensure MAC, it is necessary to externally control players' actions.

We consider the control of players that follow local learning dynamics. A local learning algorithm $\Phi(a^0)$ assumes agents repeatedly take actions and observe others' actions yielding a sequence of action profiles $(a^0, a^1, \ldots)$ starting from an initial action profile a^0. We will make use of the notation

$$\Phi_k(a^0) := a^k \quad (3)$$

to denote the resulting action profile after k iterations.

A *control profile* is an infinite sequence of subsets of players: $\mathcal{X} = \{\mathcal{X}^0, \mathcal{X}^1, \ldots\}$ with $\mathcal{X}^t \subseteq \mathcal{N}$ for all $t \geq 0$. We let $x_i^t = 1$ if player $i \in \mathcal{X}^t$, and $x_i^t = 0$ otherwise. Furthermore, we denote $\delta_i^t \in \{0, 1\}$ as the *forced action* of player i at time $t \geq 0$ if $i \in \mathcal{X}^t$. For convention, and without loss of generality, we say $\delta_i^t = 0$ if $i \notin \mathcal{X}^t$. We write $\Delta \equiv (\delta^0, \delta^1, \ldots)$ for the sequence of forced action profiles. With the control profile $\mathcal{X}$, the action profile trajectory, with initial action profile y^0 is written

$$\Phi(y^0, \mathcal{X}, \Delta) := (a^0, a^1, \ldots) \quad (4)$$

where a^t obey the following dynamics for $t = 0, 1, \ldots$:

$$\begin{aligned} a_i^t &= (1 - x_i^t)y_i^t + x_i^t \delta_i^t, \text{for all } i \in \mathcal{N} \\ a^t &= (a_1^t, \ldots, a_n^t) \\ y^{t+1} &= \Phi_1(a^t). \end{aligned} \quad (5)$$

Note $\Phi_1(a^t)$ is the uncontrolled action at time $t + 1$ by (3) given the controlled action profile a^t at time t. We will refer to the pair $(\mathcal{X}, \Delta)$ as a *control policy*.

After the following remark, we describe the players' best-response-type learning dynamics.

Remark 2: An underlying goal is to achieve efficiency measured by the social welfare, e.g., sum of players' payoffs. The premise for using efficiency as a metric is that an action profile that maximizes social welfare would be desired by the society as a whole. Note that the social welfare maximizing action profile here need not eliminate all active links, e.g., cases with very small c_0 and c_1 would have all players taking action 1 as the NE and as the efficient action profile. In settings, e.g., epidemics, or wireless communications, a social welfare maximizing action profile that does not eliminate any active links may be undesired. In such scenarios, counting the number of (active) links that fail anticoordination may be a better measure of efficiency. Along these lines, MAC represents the goal to eliminate all active links.[2]

A. Progressive Decentralized Learning Dynamics

We assume players repeatedly play the anticoordination network game $\{\mathcal{N}, \mathcal{A}, \mathcal{G}_B, \{u_i\}_{i\in\mathcal{N}}\}$ taking actions $a_i^k \in \mathcal{A}_i$ at

[2] In [33], we show the effectiveness of inducing good behavior in an epidemics scenario. Specifically, a disease propagates over a network changing the state of nodes at each step. The policy-maker wants to eliminate as many active (risky) links as possible by selecting a given number of players (budget) at each step. The results show the effectiveness of the approach compared to isolation of players based on standard centrality-based metrics.

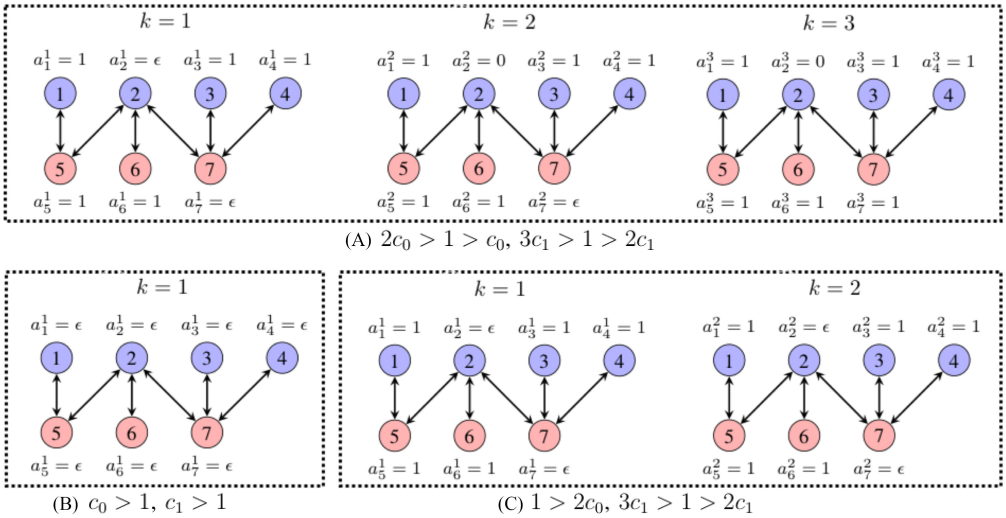


Fig. 2. Algorithm 1's convergence with respect to different payoff constants (A, B, C) on a 7 player network. Initially, all players are undecided $a_i^0 = \epsilon$. (A) At step $k = 1$, players 1, 3, and 4 update their actions to 1 by (8) given $1 > c_0$, players 5 and 6 also update their action to 1 by (8) given $1 > 2c_1$, and players 2 and 7 remain undecided. At step $k = 2$, player 2 observes previous actions of its neighbors—$\{a_5^1, a_6^1\}$—and updates its action to 0 by (9) given $2c_0 > 1$. At step $k = 3$, player 7 observes player 2's update—$a_2^2 = 0$—and updates its action to 1 by (8) given $1 > 2c_1$. All players are decided, and the action profile is the unique NE of the game. (B) All players remain undecided for all times because no player can eliminate any action by (8). (C) At step $k = 1$, players 1, 3, 4, 5, and 6 update to action 1 similar to case (A). At step $k = 2$, neither player 2 nor player 1 can eliminate any actions, and both remain undecided. A subset of players are decided while the others remain undecided.

each stage k. Given neighbor action profile $a_{\mathcal{N}_i}$, the best-response action set for player i is defined as

$$BR_i(a_{\mathcal{N}_i}) := \underset{a_i \in [0,1]}{\operatorname{argmax}}\, u_i(a_i, a_{\mathcal{N}_i}) \tag{6}$$

$$= \mathbb{I}\left(1 > (c_0(1 - s_i) + c_1 s_i) \sum_{j \in \mathcal{N}_i} a_j\right) \tag{7}$$

where $\mathbb{I}(\cdot)$ is the indicator function. The actions that maximize the payoff in (1) are in the extremes: $a_i = 1$ or $a_i = 0$, as per (7). During repeated play, we consider learning dynamics, detailed in Algorithm 1, in which an agent selects an extreme action if it is sure that it is the best action under any rational behavior of its neighbors.

Specifically, each player starts by selecting an arbitrary action $a_i^0 \in \{0, \epsilon, 1\}$, where $\epsilon \in (0, 1)$ represents an *undecided* player's action (see Algorithm 1). Player i decides on action 1 ($a_i^k = 1$) as per (8) if it is the best-response action even when its undecided neighbors ($\{j \in \mathcal{N}_i : a_{j,k-1} = \epsilon\}$) end up taking action 1. That is, the ceiling operator makes a worst case scenario assumption and evaluates the utility from action 1. Player i decides on action 0 ($a_i^k = 0$) as per (9) if it is the best-response action even when its undecided neighbors ($\{j \in \mathcal{N}_i : a_{j,k-1} = \epsilon\}$) end up taking action 0. That is, the floor operator makes a best case scenario assumption and evaluates the utility from action 1. Note that the ceil and the floor operators do not alter the actions of neighboring players whose previous action are 0 or 1. If (8) and (9) do not hold at step k, player i remains undecided, i.e., $a_i^k = \epsilon$. We note that our results continue to hold when undecided actions are heterogeneous and time-varying.

Algorithm 1: Local Learning Behavior for Agent i.

Input: $a_i^0 \in \{0, \epsilon, 1\}$;
for $k = 1, 2, \ldots$ **do**
  Observe $a_{\mathcal{N}_i}^{k-1}$;
  Update action using

$$a_i^k = 1 \quad \text{if} \quad \{1\} = BR_i(\lceil a_j^{k-1} \rceil) \tag{8}$$

$$a_i^k = 0 \quad \text{if} \quad \{0\} = BR_i(\lfloor a_j^{k-1} \rfloor)$$

$$a_i^k = \epsilon, \quad \text{otherwise} \tag{9}$$

end for

Fig. 2 shows the iterations of Algorithm 1 on a 7 player network for different payoff constants. We observe that depending on the payoff constants (c_0, c_1), the algorithm gives an action profile where all players are decided (A), all remain undecided (B), or some are decided and some remain undecided (C). In all cases, the updates converge after at most $k = 3$ steps. Indeed, the local learning rule is *progressive* in the sense that once decided, a player will never change its action or become undecided. This property leads to convergence of the action profile in at most n steps where we recall n is the number of players (see Theorem 2 in Appendix B). The finite time convergence guarantee allows efficient evaluation of the objective in the MAC problem presented in the following section.

Remark 3: In addition to being progressive, the local behavior rule eliminates all strictly dominated actions, i.e., actions that can not be rational—see Definition 3. Thus it guarantees convergence to the unique NE when the game is dominance solvable [27][Ch. 2], e.g., see Fig. 2(A). When the game is not dominance solvable, the local behavior rule converges with a subset of the players remaining undecided, e.g., see Fig. 2(B) and (C). This implies that there are multiple (at least two) Nash equilibria in this game. For instance, in Fig. 2(B) and (C), if all undecided players in $\mathcal{S}_0$ play action 1 (0) and all undecided players in $\mathcal{S}_1$ play action 0 (1), then we obtain a NE action profile.

Remark 4: Following the above remark's intuition, we can guarantee convergence to a NE profile for any payoff constant values (c_0 and c_1), and any network $\mathcal{G}_B$ by adding a random decision-making step to the local learning rule as follows. In this new step, we can let a randomly chosen undecided player select an action $\{0, 1\}$ after every n steps. Because an undecided player takes an action, this would potentially bring a cascade of new decisions via steps (8) and (9) in the following steps. This additional step would guarantee convergence to a NE action profile in at most n^2 steps by the same reasoning in Theorem 2. Here, our focus is not to find an NE but to achieve MAC via external control. Thus, we keep the local learning rule as in Algorithm 1 due to the fact that it allows efficient evaluation of the MPCAC objective.

B. Static and Dynamic MPCAC

In the first formulation of the MPCAC problem, we seek to achieve MAC using as few fixed control players as possible. In this static setting, a *static control policy* $(\mathcal{X}_s, \delta)$ is specified by one control set $\mathcal{X}_s \subseteq \mathcal{N}$, and one forced action profile δ.

Definition 1 (Static MPCAC):

$$\begin{aligned} \min_{\substack{\mathcal{X}_s \subset \mathcal{N} \\ \delta_i \in \{0,1\}, i \in \mathcal{X}_s}} \quad & J_s(\mathcal{X}_s, \delta) := |\mathcal{X}_s| \\ \text{s.t. } & a^n \in M(\mathcal{G}_B) \\ & \mathcal{X} = \{\mathcal{X}_s, \mathcal{X}_s, \ldots\} \\ & \delta_i^t = \delta_i \in \{0, 1\} \quad \text{for all } \ i \in \mathcal{X}_s, \text{ for all } t \geq 0 \\ & (a^0, a^1, \ldots) = \Phi(\vec{\epsilon}, \mathcal{X}, \Delta), \text{ where } \vec{\epsilon} = \epsilon 1_n. \end{aligned} \tag{10}$$

The first constraint ensures MAC by time n where all players must be decided (2). The second and third constraints define the controlled players and their forced actions for all times, respectively. The last constraint specifies the controlled learning dynamics (4). A static control policy $(\mathcal{X}_s, \delta)$ is called *feasible* if it satisfies the above constraints.

The optimization problem (10) runs for n time steps to allow for convergence of Algorithm 1. The following result shows that any static policy that satisfies the MAC (first) constraint above at time n will continue to satisfy it for $t > n$.

Lemma 1: If $\Pi_s = (\mathcal{X}_s, \delta)$ is a feasible static policy, then $a^t = a^n$ for all $t > n$.

Proof: Define the game Γ' among players $\mathcal{N} \setminus \mathcal{X}_s$ where players connected to $\mathcal{X}_s$ have a set of decided neighbors according to forced action profile δ. The game Γ' must be dominance solvable so that Algorithm 1 converges by time n. Thus, if we continue to apply the forced actions δ, no player would change its decision after time n. ■

In static MPCAC, we decide on players to control at the beginning and set their actions for the entire horizon. The objective only accounts for the number of players controlled but not the number of times we control a player. In many situations, it may be enough to control a player for a finite time to achieve MAC. For instance, in case (B) in Fig 2, if we set the actions of players 5–7 to 0 for one time step, the remaining players (1–4) will take action 1 by (8). If we stop controlling the players 5–7 in the next time step, they will continue to take action 0 by (9). Hence, the resultant action profile will achieve maximal anticoordination. We formulate the dynamic MPCAC problem that allows for dynamic selection of players to control.

Definition 2 (Dynamic MPCAC):

$$\begin{aligned} \min_{\mathcal{X}, \Delta} J_d(\mathcal{X}, \Delta) := \frac{1}{n} \sum_{t=1}^{n} |\mathcal{X}^t| + \lim_{T' \to \infty} \frac{1}{T'} \sum_{t=n+1}^{T'} |\mathcal{X}^t| \\ \text{s.t. } a^t \in M(\mathcal{G}_B) \quad \text{for all } t \geq n \\ (a^0, a^1, \ldots) = \Phi(\vec{\epsilon}, \mathcal{X}, \Delta). \end{aligned} \tag{11}$$

The two terms in the penalty function in (11) equally weigh the control effort per player before convergence and after convergence to MAC. In dynamic MPCAC, we allow for the set of controlled players to change at each step. Lemma 1 does not necessarily apply in the dynamic setting. Hence, we explicitly require that the MAC constraint is maintained for all times after n in the first constraint. The last constraint specifies the controlled learning dynamics (4).

As is evident from the formulations of static and dynamic MPCAC and Lemma 1, an optimal static MPCAC policy is a feasible solution to the dynamic MPCAC. Thus, the average number of players controlled per time step in the optimal dynamic policy is less than the number of players in the optimal static control policy, i.e., $\frac{1}{n} \sum_{t=1}^{n} |\mathcal{X}_*^t| \leq |\mathcal{X}_s^*|$ where $\mathcal{X}_*^t$ is the optimal set of players to control in a dynamic policy, and $\mathcal{X}_s^*$ is the optimal static policy. Hence, we can always use the optimal solution for static MPCAC to upper bound the penalty in the dynamic MPCAC. In fact, if the static MPCAC solution reaches an action profile a^n that is an equilibrium of the game, then the optimal solution to dynamic MPCAC is upper bounded by $|\mathcal{X}_s^*|$. That is, we do not need to make any control efforts to remain in $M(\mathcal{G}_B)$ because the action profile is also an equilibrium of the game.

The first constraint of dynamic MPCAC requires that MAC is satisfied for all $t \geq n$, whether or not control actions are used to maintain it. This together with the penalization of control efforts for all times after n gives preference to control policy solutions, when feasible, that leverage the controlled dynamics Φ in (4) in order to achieve MAC. The following result supports this intuition for dominance solvable games.

Lemma 2: If the game is dominance solvable, the optimal policy for dynamic MPCAC ($\Pi_d = (\mathcal{X}_*, \Delta_*)$) is either $\mathcal{X}_*^t = \emptyset$ for all $t = 1, 2, \ldots$ or $\mathcal{X}_*^t \neq \emptyset$ for $t \geq n$.

The proof (given in the Appendix) relies on whether the unique NE is in $M(\mathcal{G}_B)$ or not, which then corresponds to an empty control profile or a nonempty control profile for $t > n$, respectively. Note that Γ always has a pure NE—see Remark 3. However, a pure NE that achieves MAC may or may not exist. In general, if the game Γ has (pure) Nash equilibria in $M(\mathcal{G}_B)$, we

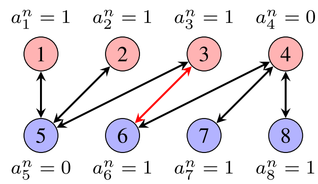


Fig. 3. Assume $c_0 = 0.4$ and $c_1 = 0.4$. The game is dominance solvable and Algorithm 1 converges to the above action profile with the only active link between 3 and 6. That is, $\mathcal{N}^n = \{1, 2, 3, 6, 7, 8\}$ and $\mathcal{E}^n = \{(3, 6)\}$. The solution to (12) if we disregard the last constraint is either $x_3^* = 1$ or $x_6^* = 1$. These solutions violate the last constraint for players 4 or 5. If we fix $\rho_i = 0$ for $i \in \{4, 5\}$, then (12) is infeasible.

would expect control policies that induce convergence to such Nash equilibria over policies that control players after time n in order to avoid the second term cost of (11). We formalize this intuition in the following lemma (proof given in Appendix).

Lemma 3: If there exists a set of Nash equilibria that achieves MAC, then the optimal solution to dynamic MPCAC will reach an action profile in this set.

C. Dynamic Policy Based on Minimum Vertex Cover

We relate the dynamic MPCAC problem to a vertex covering problem on a bipartite network. We define the graph with active links after Algorithm 1 converges at time n as follows. Let $\mathcal{G}^n = (\mathcal{N}^n, \mathcal{E}^n)$ be the graph with vertices composed of players that remain undecided or take action 1 after Algorithm 1 converges at step n, i.e., $\mathcal{N}^n := \{j \in \mathcal{N} : a_j^n = \{\epsilon, 1\}\}$, and with edges that connect players in $\mathcal{N}^n$ in $\mathcal{E}_B$, that is, $\mathcal{E}^n = \{(i, j) \in \mathcal{E}_B : i, j \in \mathcal{N}^n\}$. For example, in Fig. 2 case (A), the network $\mathcal{G}^n$ has the vertex set $\mathcal{N}^n = \{1, 3, 4, 5, 6, 7\}$.

A cardinality vertex cover for the graph $\mathcal{G}^n = (\mathcal{N}^n, \mathcal{E}^n)$ looks for a minimum cardinality subset of vertices $\mathcal{X}^n \subseteq \mathcal{N}^n$ such that each edge has at least one endpoint incident at $\mathcal{X}^n$ [34]. Given the action profile a^n, we consider the following modified cardinality vertex cover problem with tuning parameter $\lambda \geq 0$:

$$\min_{\mathbf{x}, \rho \geq 0} \sum_{i \in \mathcal{N}^n} x_i + \lambda \sum_{i \in \mathcal{N} \setminus \mathcal{N}^n} |\rho_i|$$
$$\text{s.t. } \mathbf{x} \in M(\mathcal{G}^n)$$
$$c_i \sum_{j \in \mathcal{N}_k} (\lceil a_j^n \rceil - x_j) \geq 1 - \rho_i \quad \text{for all } k \notin \mathcal{N}^n \tag{12}$$

where $c_i := c_0(1 - s_i) + c_1 s_i$ is player i's payoff constant. If the second term in the objective and the last constraint are excluded, the problem formulation would be the minimum vertex covering problem in the bipartite network $\mathcal{G}^n$. If we set $\rho_i = 0$ for players $i \notin \mathcal{N}^n$, the last constraint makes sure that we select the players in the vertex covering such that the players $i \notin \mathcal{N}^n$ who are decided on action 0, do not change their actions as a result of the control efforts. In general, the last constraint with $\rho_i = 0$ can make (12) infeasible—see Fig. 3 for an example. However, we note that we can always achieve MAC by solving the minimum vertex cover problem on the original network $\mathcal{G}_B$. One of the solutions of this problem is to control all the players belonging to the type with fewest number of players.

We construct a feasible control policy $\Pi_v = \{\mathcal{X}, \Delta\}$ as follows. Let $\mathcal{X}^t = \emptyset$ if $t < n$. Let $\mathbf{x}^* = (x_1^*, \ldots, x_n^*)$, and $\rho = (\rho_1^*, \ldots, \rho_n^*)$ be an optimal solution to (12). Define the controlled player set $\mathcal{X}^n$ and their actions Δ^n at time n based on the optimal solution of the above problem as follows:

$$\mathcal{X}^n = \{i \in \mathcal{N}^n : x_i^* = 1\} \cup \{i \in \mathcal{N}^n : x_i^* = 0, a_i^n = \epsilon\} \tag{13}$$

and

$$\delta_i^n = \begin{cases} 0 & \text{for all } i \in \{i \in \mathcal{N}^n : x_i^* = 1\} \\ 1 & \text{for all } i \in \{i \in \mathcal{N}^n : x_i^* = 0, a_i^n = \epsilon\}. \end{cases} \tag{14}$$

For $t = n + 1$, we define the controlled player set $\mathcal{X}^{n+1}$ and their actions Δ^{n+1} as follows:

$$\mathcal{X}^{n+1} = \{i \in \mathcal{N}^n : x_i^* = 1\} \cup \{i \notin \mathcal{N}^n : \rho_i^* > 0\} \tag{15}$$

and

$$\delta_i^{n+1} = 0 \quad \text{for all } i \in \mathcal{X}^{n+1}. \tag{16}$$

For $t \geq n + 1$, we let $\mathcal{X}^t = \mathcal{X}^{n+1}$ if the action profile $\{a_i^{n+1}, \delta_i^{n+1}\}_{i \in \mathcal{N}}$ is not a NE of the game. Otherwise, $\mathcal{X}^t = \emptyset$ for $t > n$.

Theorem 1: The dynamic control policy $\Pi_v = \{\mathcal{X}, \Delta\}$ with control and action sets at time n given by (13)–(14) and for time $t \geq n + 1$ given by (15)–(16) is a feasible dynamic policy for the dynamic MPCAC problem in (11).

Proof: Given (13) and (14), (5) gives the controlled action profile

$$a_i^n = \begin{cases} 1 - x_i^* & \text{if } i \in \mathcal{N}^n \\ 0 & \text{o.w.} \end{cases} \tag{17}$$

Given the definition of $\mathcal{G}^n$ and Δ^n, we have the controlled actions a^n satisfy $a_i^n + a_j^n \leq 1$ for any $(i, j) \in \mathcal{E}^n$, since we force at least one player in every link of $\mathcal{G}^n$ to play action 0. This means that $a^n \in M(\mathcal{G}_B)$. Further, all players are decided satisfying the first constraint in (11).

Next, we show $\mathcal{X}^{n+1}, \Delta^{n+1}$ maintains anticoordination with the controlled action profile a^{n+1}. Consider the set partition

$$\begin{aligned} \mathcal{N}^n &= \{i \in \mathcal{N} : y_i^n = 1\} \\ &\cup \{i \in \mathcal{N} : y_i^n = \epsilon, x_i^* = 1\} \\ &\cup \{i \in \mathcal{N} : y_i^n = \epsilon, x_i^* = 0\} \end{aligned} \tag{18}$$

where y^n is the uncontrolled action profile at time n. For the first two sets in (18), we have $a_i^{n+1} = 1 - x_i^*$ by (15). The last set of players were controlled to play action 1. At time n, they have no neighbors that take action 1 in (14). So they continue to take action 1 using the update (8) even when they are not controlled any longer. Hence, $y_i^{n+1} = 1 - x_i^*$ for $\{i \in \mathcal{N} : a_i^n = \epsilon, x_i^* = 0\}$. This implies that $a_i^{n+1} = 1 - x_i^*$ for all $i \in \mathcal{N}^n$.

Next, consider the following partition of the set of players not belonging to $\mathcal{N}^n$:

$$\mathcal{N} \setminus \mathcal{N}^n = \left\{ i \notin \mathcal{N}^n : a_i^n = 0, c_i \sum_{j \in \mathcal{N}_i} (\lceil a_j^n \rceil - x_j^*) < 1 \right\}$$
$$\cup \left\{ i \notin \mathcal{N}^n : a_i^n = 0, c_i \sum_{j \in \mathcal{N}_i} (\lceil a_j^n \rceil - x_j^*) > 1 \right\}. \tag{19}$$

Note that all the players in the first set in (19) are controlled to play action 0 in (15), that is, $a_i^{n+1} = 0$ because $\{i \notin \mathcal{N}^n : a_i^n = 0, c_i \sum_{j \in \mathcal{N}_i} (\lceil a_j^n \rceil - x_j^*) < 1\} = \{i \notin \mathcal{N}^n : \rho_i^* > 0\}$ by the last constraint in (12). For the second set in (19), all players continue to select action 0 by (9). This implies that $a_i^{n+1} = 0$ for $i \notin \mathcal{N}^n$.

Combining the two arguments, we have $a_i^{n+1} = a_i^n$ where a_i^n is given in (17). Hence, the controlled action profile $a^{n+1} \in M(\mathcal{G}_B)$. Suppose, a^{n+1} is a NE action profile, then $a^{n+1} = \Phi_1(a^{n+1})$ and no further control effort is necessary. Otherwise, we have $y^{n+2} = \Phi_1(a^{n+1}) = \Phi_1(a^n) = y^{n+1}$ where y^{n+1} and y^{n+2} are the uncontrolled action profiles at time $n+1$ and $n+2$, respectively. Since, $\mathcal{X}^t = \mathcal{X}^{n+1}$, $\Delta^t = \Delta^{n+1}$ for $t \geq n+1$, we have that $a^{n+2} = a^{n+1}$. By induction, the control policy Π_v is feasible. ■

Theorem 1 shows that the MPCAC problems can be upper bounded by solving the minimum vertex cover on the reduced bipartite graph $\mathcal{G}^n$. Further, if we set the tuning parameter $\lambda = 0$, then the integer program in (12) has an exact linear programming relaxation due to total unimodularity of bipartite networks [35, Ch. 3]. Hence, we can obtain a feasible policy efficiently by solving (12) with penalty term $\lambda = 0$. Then, including all players that have positive ρ_i in the control set ($\{i \notin \mathcal{N}^n : \rho_i^* > 0\}$). The following corollary presents a scenario in which Π_v optimal.

Corollary 1: The policy Π_v defined in Theorem 1 is an optimal policy if all players eliminate their actions in one time step in Algorithm 1.

The proof given in the appendix relies on showing that when all players eliminate their actions, the optimization problem in (12) reduces to solving the minimum cardinality vertex cover for the entire network $\mathcal{G}$. This case happens only when c_i is smaller than the inverse of the maximum degree of the network ($c_i|\mathcal{N}_i| < 1$ for all $i \in \mathcal{N}$). The control policy Π_v is an upper bound in general because it does not make use of cascades, i.e., use the learning dynamics to make multiple hop links inactive, by controlling a subset of players. Instead, it directly eliminates all remaining active links. As we show in numerical examples in Section VI, the policy Π_v tends to perform well when c_0 is small and c_1 is large, or when c_0 is large and c_1 is small. In the following section, we present optimal policies for benchmark networks that exemplify the optimal use of cascades to eliminate active links.

IV. Optimal Solutions for Benchmark Networks

A. Star Network

Consider a star network of n players with player 1 in the center. Without loss of generality we assume $s_1 = 1$ and $j \in \mathcal{S}_0$ for $j \notin \{1\}$—see Fig. 1.

Proposition 1: The optimal dynamic MPCAC solutions for the star network are as follows.

a) If $1 > c_0$ and $1 > c_1(n-1)$, then

$$\begin{cases} \mathcal{X}^t = \varnothing, & \text{if } t < n \\ \mathcal{X}^t = \{1\}, \delta_1^t = 0 & \text{if } t \geq n. \end{cases}$$

The resulting optimal cost is $C^* = 1$.

b) If $1 < c_0$ and $1 > c_1(n-1)$, or $1 > c_0$ and $1 < c_1(n-1)$, then $\mathcal{X}^t = \varnothing$ for all t. The resulting optimal cost is $C^* = 0$.

c) If $1 < c_0$ and $1 < c_1(n-1)$, then $\mathcal{X}^1 = \{1\}$.

$$\begin{cases} \mathcal{X}^t = \{1\}, \delta_1^t = 0 & \text{if } t = 0, 1 \\ \mathcal{X}^t = \varnothing & \text{if } t > 1. \end{cases}$$

The resulting optimal cost is $C^* = 2/n$.

Proof: In cases (a) and (b), the game is dominance solvable. Further, the algorithm converges in $t \leq 2$ steps. Lemma 2 together with Theorem 1 gives the optimal policies (a) and (b), respectively. In case (c), the game is not dominance solvable and there exists two Nash equilibria belonging to $M(\mathcal{G}_B)$: ($a_1 = 0, \{a_i = 1\}_{i \neq 1}$) and ($a_1 = 1, \{a_i = 0\}_{i \neq 1}$). By Lemma 3, the optimal policy will induce the dynamics to converge to one of these configurations. Either of these equilibria can be achieved through a dynamic policy that targets only the center node for two time steps, which gives the optimal cost $2/n$. For instance, we set $\delta_1^0 = 0$ so that fringe players select $a_i^1 = 1$. We continue with $\delta_1^1 = 0$, so that the center node's unforced action is 0 by (9). We lift the control for all $t \geq 2$, by which time the players are in equilibrium and MAC is achieved. This policy is optimal because there cannot be a policy that achieves a smaller objective value than $2/n$ and achieve MAC. ■

The above analysis exemplifies the three uncontrolled action profiles (3) that can arise in a star network: (a) all players decide but MAC is not satisfied; (b) MAC is satisfied; (c) all players remain undecided. In case (a), we control the center player to induce MAC. In case (b), no control is necessary. In case (c), we control the center player to trigger decisions for the fringe players.

B. Line Network

We consider a line network in which the neighborhood of player $i \in \mathcal{N} \setminus \{1, n\}$ is given by $\mathcal{N}_i = \{i-1, i+1\}$. The players at the endpoints have neighbor sets $\mathcal{N}_1 = \{2\}, \mathcal{N}_n = \{n-1\}$. The type configuration alternates between types 0 and 1: $i \in \mathcal{S}_0$ for i odd, and $i \in \mathcal{S}_1$ for i even. We will also refer to the subsets $\mathcal{S}_m^{\text{odd}}, \mathcal{S}_m^{\text{even}}$ for $m \in \{0, 1\}$ to denote every odd (even) player from the set $\mathcal{S}_m$ along the line. In the following, we consider all possible payoff cases for the line network.

Proposition 2: Depending on the utility constants c_0 and c_1, the optimal control policy ($\{\mathcal{X}^t\} \geq 0, \Delta$) for the dynamic MPCAC problem on a line network with n odd is given as

a) If $1 > 2c_0$ and $1 > 2c_1$, then

$$\begin{cases} \mathcal{X}^t = \varnothing & \text{if } t < n \\ \mathcal{X}^t = \mathcal{S}_1, \{\delta_i^t\}_{i \in \mathcal{S}} = 0 & \text{if } t \geq n \end{cases}$$

which gives the optimal cost $C^* = \lfloor n/2 \rfloor$.

b) If $1 > 2c_0$ and $2c_1 > 1 > c_1$, then $\mathcal{X}^t = \varnothing$ for all $t \geq 0$.

c) If $1 > 2c_0$ and $c_1 > 1$, then $\mathcal{X}^t = \varnothing$ for all $t \geq 0$.

d) If $2c_0 > 1 > c_0$ and $2c_1 > 1 > c_1$, then

$$\begin{cases} \mathcal{X}^t = \mathcal{S}_1^{\text{even}}, \delta_{\mathcal{X}^t}^t = 0 & \text{if } t = 0, 1 \\ \mathcal{X}^t = \varnothing & \text{if } t > 1 \end{cases}$$

which gives the optimal cost $C^* = 2\lfloor n/4 \rfloor / n$.

e) If $c_0 > 1$ and $2c_1 > 1 > c_1$, then

$$\begin{cases} \mathcal{X}^t = \{i\}, \delta_i^t = s_i & \text{if } t = 0, 1 \\ \mathcal{X}^t = \varnothing & \text{if } t > 1 \end{cases}$$

for any $i \in \mathcal{N}$. This gives the optimal cost $C^* = 2/n$.

f) Suppose $c_0 > 1$ and $c_1 > 1$. Then

$$\begin{cases} \mathcal{X}^t = \mathcal{S}_0^{\text{even}}, \delta_{\mathcal{X}^t}^t = 1 & \text{if } t = 0, 1 \\ \mathcal{X}^t = \varnothing & \text{if } t > 1 \end{cases}$$

gives the optimal cost $C^* = 2\lceil (n-1)/4 \rceil / n$.

The optimal control policies vary depending on the payoff constants. If all players decide to play action 1 as a result of the learning process, then we solve a minimum vertex covering problem along the lines of Theorem 1, and the players selected have to be controlled for all times $t \geq n$—case (a) in all of the propositions in this section. If the game is dominance solvable and all links are inactive in the equilibrium action profile, the optimal policy is the empty set—see 1(b), 2(b,c). If a subset of players remain undecided as a result of the learning algorithm and all links between decided players are inactive, an optimal policy can be found by leveraging Lemma 3. In the benchmark examples, these policies required controlling a subset of the players for two time steps, causing the undecided players in the other type to decide in the first step and the controlled players to consolidate their decisions in the second time step. Consequently, we can control a smaller subset of players and leverage the learning dynamics to create cascades of decision-making—see cases (d–f) in Proposition 2. In the latter scenarios, the policy suggested by Theorem 1 is an upper bound of the optimal policy. The optimal static policies are similar to the dynamic MPCAC policies presented here.

V. Suboptimal Algorithms for General Networks

For general bipartite network topologies, it is challenging to solve exactly for the MPCAC (both static and dynamic) solution. We devise an algorithm that selects at each iteration one player to control according to a greedy approach. The algorithm results in a subset of control agents $\hat{\mathcal{X}}$ that ensures MAC, but may not be the optimal MPCAC solution.

Given a network game $\{\mathcal{N}, \mathcal{A}, \mathcal{G}_B, \{u_i\}_{i \in \mathcal{N}}\}$, we define player i's active neighbor set given action profile $a \in \mathcal{A}$ as

$$\mathcal{N}_i^*(a) := \{j \in \mathcal{N}_i : a_j \in \{\epsilon, 1\}\}.$$

The *active edge set* of the network in action profile a is

$$\mathcal{E}^*(a) := \{(i,j) \in \mathcal{E}_B : a_i, a_j \in \{\epsilon, 1\}\}.$$

At each iteration k, the *greedy algorithm* selects the player i_k^* that, upon holding its action fixed at $a_{i_k^*} = \delta_i \in \{0, 1\}$, results in the most number of active edges eliminated by the time the system dynamics converge. We call the number of such links eliminated from choosing any player i in the action profile a the *cascade potential* of player i

$$\text{CP}_i(a, \delta_i) := |\mathcal{E}^*(a)| - |\mathcal{E}^*(\Phi_n(a, \{i\}_{t\geq 0}, \{\delta_i\}_{t \geq 0}))|.$$

The process repeats, incrementally building up the player control set $\hat{\mathcal{X}}$, until all active edges are eliminated from the network.

Algorithm 2: A Greedy Algorithm for MPCAC.

1: **Input:** $a_i^0 = \epsilon$ for all $i \in \mathcal{N}$; $\hat{\mathcal{X}}(0) = \varnothing$; $\hat{\delta} = 0_n$; $k \leftarrow 0$;
2: **while** $a^k \notin M(\mathcal{G}_B)$ **do**
3: Allow dynamics to run until convergence

$$a^{k+1} \leftarrow \Phi_n(a^k, \{\hat{\mathcal{X}}(k)\}_{t\geq 0}, \{\hat{\delta}\}_{t \geq 0})$$

4: Store convergence time

$$t_k \leftarrow \min\{t \leq n : \Phi_t(a^k, \hat{\mathcal{X}}(k), \hat{\delta}) = \Phi_{t-1}(a^k, \hat{\mathcal{X}}(k), \hat{\delta})\}$$

5: Selection criterion

$$(i_k^*, \delta_{i_k^*}) \leftarrow \text{rand}\left(i \in \underset{\substack{i: a_i^{k+1} \in \{\epsilon, 1\} \\ \delta_i \in \{0,1\}}}{\text{argmax}} \text{CP}_i(a^{k+1}, \delta_i) \right)$$

6: Update control set
$\hat{\mathcal{X}}(k+1) \leftarrow \hat{\mathcal{X}}(k) \cup i_k^*$
$\hat{\delta}_{i_k^*} \leftarrow \delta_{i_k^*}$
7: $k \leftarrow k + 1$
8: **end while**
9: $\hat{\mathcal{X}} \leftarrow \hat{\mathcal{X}}(k)$

If multiple players satisfy the maximum cascade potential criterion in step 5, ties are broken by random selection. The looping condition makes sure that we select players to control until no links remain active. We also consider a few variants of the above algorithm by replacing the selection criterion in step 5) by

a) $i_k^* \leftarrow \text{rand}(i \in \text{argmax}_{i: a_i^k \in \{\epsilon, 1\}} |\mathcal{N}_i^*(a^k)|)$
$\hat{\delta}_{i_k^*} \leftarrow 0$ (max degree)

b) $i_k^* \leftarrow \text{rand}(\{i : (i,j) \in \mathcal{E}^*(a^{k+1})$ for some $j\})$
$\hat{\delta}_{i_k^*} \leftarrow 0$ (rand)

c) $i_k^* \leftarrow \text{rand}(i \in \text{argmax}_{i, \delta_i} \text{CP}_i(a^{k+1}, \delta_i) + \sum_{j=1}^n 1(a_j^{k+1} \neq a_j^k)$ (CP 2)

d) Here, replace Algorithm 2 after **Initialize** with

$$\bar{a} \leftarrow \Phi_n(a^0)$$

$$\hat{\mathcal{X}} = \text{VC}(\mathcal{E}^*(\bar{a}))$$

where VC is the minimum vertex cover scheme detailed in (15) of the resulting active network after the first convergence of the dynamics.

In a), a player with the maximum number of active neighboring links is selected to play action 0. We denote this the "max degree" variant. In b), one player connected to the active network is selected uniformly at random with forced action 0. This algorithm is called "rand." We note that the "rand" algorithm can be thought of as an advertising model, where a random player with active links becomes receptive to the advice from the designer. A similar advertising model, proposed in [16] and [17] for party affiliation and cut games, is a "minimally invasive" designer policy that does not require information about the learning dynamics and network structure. A variant of cascade potential, "CP2," is described in c), which selects the node that has the highest cascade potential in addition to inducing the most amount of players to change their action (see Remark 5 for motivation). In d), we select the control set as the minimum vertex cover as specified in (15) of Theorem 1. Note that in variants "max degree," "rand," and "VC," the forced actions are always $\hat{\delta}_i = 0$, since it guarantees active edges to be eliminated (immediate neighbors). In the CP-based algorithms, $\hat{\delta}_i$ can either be 0 or 1, depending on which results in the most "cascaded" active links broken.

Proposition 3: For all variants of Algorithm 2, a resulting control set $(\hat{\mathcal{X}}, \hat{\delta})$ is a feasible static control policy of MPCAC. A feasible dynamic policy $(\mathcal{X}, \Delta)$ is also produced by implementing control on the players i_k in the order they were selected, and holding $a_{i_k} = 0$ for all times

$$\mathcal{X}^t = \begin{cases} \hat{\mathcal{X}}(k), \text{ for } \sum_{m=0}^{k-1} t_m \le t < \sum_{m=0}^{k} t_m \\ \hat{\mathcal{X}}, \text{ for } t \ge \sum_{m=0}^{\bar{k}} t_m \end{cases}$$

$$\Delta^t = \hat{\delta}, \quad \text{for all } t \ge 0$$

with $\sum_{m=0}^{-1} t_m \equiv 0$.

Proof: Consider the static policy $(\hat{\mathcal{X}}, \hat{\delta})$. By the definition of Algorithm 2, step 1 at the last iteration k ensures MAC, where the control set used is $\hat{\mathcal{X}}(k) = \hat{\mathcal{X}}$. That is, $\Phi_n(a^k, \hat{\mathcal{X}}(k), \hat{\delta}) \in M(\mathcal{G}_B)$. It is also true that $\Phi_n(\vec{\epsilon}, \hat{\mathcal{X}}(k), \hat{\delta})$ gives the same action profile because the players in $\hat{\mathcal{X}}$ were iteratively selected to eliminate all active links. If active links appeared as a result of a selection, these links are ensured to be eliminated in a subsequent iteration of the algorithm. Hence, $(\hat{\mathcal{X}}, \hat{\delta})$ is a feasible static MPCAC solution.

The policy $(\mathcal{X}^t, \Delta)$ is a feasible dynamic MPCAC solution because it simply mimics the iterations of Algorithm 2. ■

Remark 5: In practice, Algorithm 2 (CP variant, as written) will find the optimal static solution to the star network. For the line network, it finds the optimal static solution in cases (b-e) of Proposition 2, and in cases (a,f) with some probability. In case (a), this is due to the possibility that the algorithm can select two players with different types along the line, which causes redundancies in control because the vertex cover $\mathcal{S}_1$ is the optimal control choice by Corollary 1. In case (f), the cascade potential of selecting any player is two, as long as that player is not an endpoint. Algorithm 2 may sometimes select a player one node from an endpoint. However, selecting a player two nodes from an endpoint would cover more ground – it results in the same cascade potential in addition to forcing the endpoint player to decide. This is the motivation behind introducing the CP2 variant. CP2 variant chooses the player two nodes from an endpoint with certainty because it causes the same number of cascaded links, but induces more players to change their decision.

VI. Numerical Experiments

We consider a random bipartite network of size n by assigning half of the players to $\mathcal{S}_0$, and the other half $\mathcal{S}_1$ (assuming n is even). A link (i, j) between $i \in \mathcal{S}_0$ and $j \in \mathcal{S}_1$ is present with independent probability $p_B \in (0, 1)$.

In Fig. 4, the control effort for three variants of Algorithm 2 is mapped over varying values of c_0, c_1. The quantity $\lfloor 1/c_0 \rfloor$ is the largest number of neighbors not playing action 0 a player in S_0 can still have to play action 1 as a dominant action. Similarly, $\lfloor 1/c_1 \rfloor$ is the largest number of neighbors not playing action 0 for an S_1 node. As both c_0^{-1} and c_1^{-1} increase, more control is necessary because more nodes will be playing dominant action 1. For c_0^{-1} low and c_1^{-1} high, no control effort is necessary to achieve MAC because S_1 nodes will play dominant action 1, and S_0 nodes in turn will choose dominant action 0. That is, a pure NE that satisfies MAC exists. With both c_0^{-1}, c_1^{-1} low, no nodes can initially decide. Hence, a large control effort is needed to cause the remaining nodes to play action 1.

In Fig. 5, we plot the performance of Algorithm 2 (CP1) and its variants by measuring the control effort $|\hat{\mathcal{X}}|/n$, the fraction of players selected. In the left panel ($\lfloor 1/c_0 \rfloor = \lfloor 1/c_1 \rfloor = 0$), no players decide without external control, i.e., they all remain undecided without an intervention. The number of active links per person (total active links divided by n) is ≈ 0.7. Here, the CP-based algorithms perform significantly better than other variants because forcing $\delta_i = 1$ will cause all neighbors to play action 0. Max degree and VC perform similar. In the center panel ($\lfloor 1/c_0 \rfloor = \lfloor 1/c_1 \rfloor = 2$), a player has a strictly dominant action 1 if two or less neighbors do not play action 0. The number of active links in $\mathcal{G}^n$ per person (total active links after Algorithm 1 divided by n) is ≈ 0.5. The performance of the CP-based algorithms slightly worsen, while the variants slightly improve. In both left and center, the constants for anticoordination is the same for both types of players.

In the right panel ($\lfloor 1/c_0 \rfloor = 2, \lfloor 1/c_1 \rfloor = 1$), the constants are no longer symmetric, and $\mathcal{S}_1$ players are more sensitive to having active links than $\mathcal{S}_0$ players. In this scheme, the number of active links in $\mathcal{G}^n$ per person (total active links after Algorithm 1 divided by n) is ≈ 0.3. Correspondingly, we observe that the control effort needed is less for all proposed methods. In comparison to left and center, the VC variant performs worse than "rand," suggesting that the vertex covering scheme can in some cases be inefficient as a control policy. Indeed, "rand" achieves a control effort similar to max degree. This is because of the fact that at each selection step (Step 5 in Algorithm 2), the active link degree distribution of players with active links is close to being regular. Thus, randomization among them does not lead to a large inefficiency in control efforts. Overall, the CP-based algorithms outperform all other variants due to their exploitation of "cascades" by using the dynamics Φ to eliminate active links.

Recalling the advertising model interpretation of the "rand" [16], [17], it is worth comparing its performance to a CP-based algorithm. We can interpret the control effort in "rand" as the probability of a person accepting the advice from a communication campaign. This probability needs to be larger than 0.5 in left and center panels, while it is close to 0.3 in the right panel in Fig. 5. In comparison, CP-based algorithms require less than half of the control effort needed in "rand." The reduction in control effort is due to the additional information used by the designer in CP-based algorithms. In CP1, the designer needs to simulate forward the learning algorithm assuming it knows both the learning rules and the network structure. In contrast, the

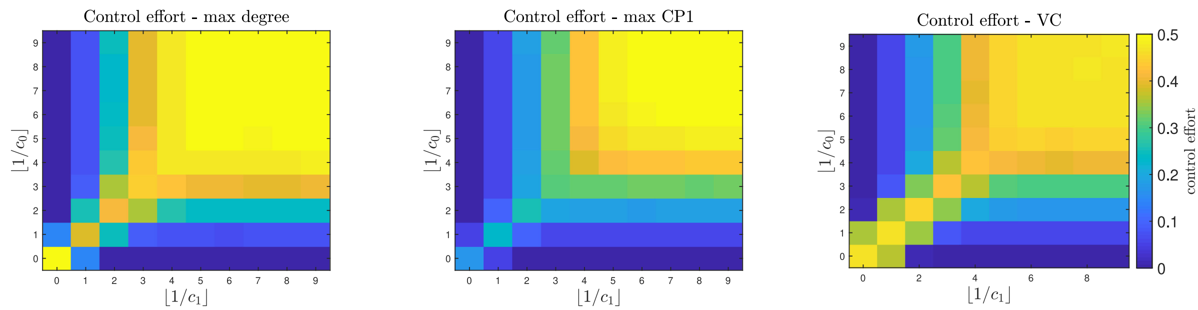


Fig. 4. Control effort $|\hat{\mathcal{X}}|/n$ used by Algorithm 2 (center) and two variants (left and right) on 20-node random bipartite networks, with $p_B = 0.3$ (expected degree of 3). Each payoff constant c_0, c_1 takes ten different values such that $\lfloor 1/c_0 \rfloor$ ranges from 0 to 10. Each value in the grid results from averaging the resulting control effort from 1000 independent realizations of the network.

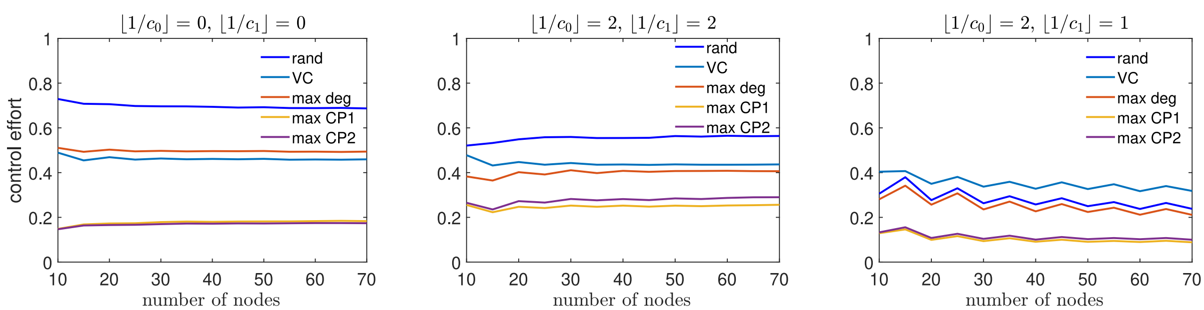


Fig. 5. Fraction of nodes $|\hat{\mathcal{X}}|/n$ (control effort) to achieve MAC versus size of the network among the five variants of Algorithm 2. Three cases of payoff constants are presented. The network is a bipartite network of size n, ranging from $n = 10$ to 70 with $+5$ increments. Each node is independently randomly chosen type 0 or 1 with equal probability, and the link (i, j), $i \in S_0$ $j \in S_1$, is drawn with probability $6/n$, to achieve an expected degree of 3. The CP-based algorithms outperform the other variants.

"rand" policy only advertises good behavior, in this case taking action 0. Thus, we can think of the gap between the control efforts of "rand" and CP1 as the value of knowing the network structure and learning dynamics.

A. Suboptimality of Greedy Algorithm

Given that CP-based algorithms are better than other proposed selection criteria, we numerically analyze CP1's suboptimality with respect to the static optimal control policy in Fig. 6 for the payoff constant values corresponding to left and right panels in Fig. 5. Fig. 6 shows the ratio of the control effort in CP1 with respect to the optimal static control effort [optimal solution to (10)]. We observe that when the payoff constants are symmetric ($c_0 = c_1$), CP1 performs worse compared to the asymmetric constant values case—the optimality ratio is the same when $\lfloor 1/c_0 \rfloor = \lfloor 1/c_1 \rfloor = 2$. The underlying reason for the performance decline is based on the fact that the number of players with the same maximum cascade potential increases with symmetry in constant values. In the case of a tie in the cascade potential of players, the CP1 randomly chooses a player among the ones in a tie possibly starting the cascade from a suboptimal player. We observed a similar case appear in the line network cases (a) and (f)—see Remark 5. When the payoff constants are not the same (blue), the largest optimality ratio is equal to 1.5. Overall, the CP-based algorithms find the optimal static policy in more than 40 of the 50 repetitions for all cases except for $n = 20$, $\lfloor 1/c_0 \rfloor = \lfloor 1/c_1 \rfloor = 0$ where the optimal is found in 36 out of 50.

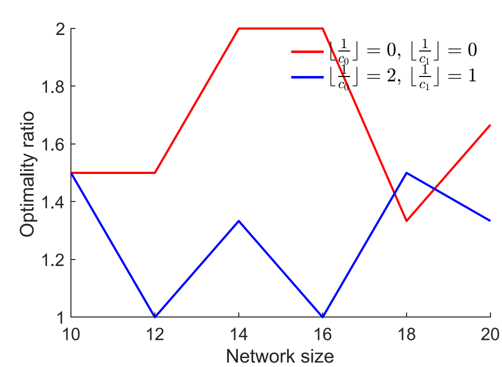


Fig. 6. Optimality gap of greedy CP1 algorithm. The network is a random bipartite network of size n, ranging from $n = 10$ to 20 with $+2$ increments. We consider 50 repetitions given n. Optimality ratio is computed by taking the maximum (worst) ratio between $|\hat{\mathcal{X}}|$ (number of players in the control set from Algorithm 2) and $|\hat{\mathcal{X}}_s^*|$ among all repetitions.

VII. Conclusion

We considered the control of learning processes in a population to influence the emergent outcome in the context of anticoordination network games. With the goal to promote MAC with minimum effort, we developed computationally tractable methods that determine when to control which players, and how to control them. A greedy algorithm that sequentially selected players according to their influence in promoting anticoordination in the future performed well in random networks with arbitrary population sizes.

APPENDIX

A. Game Theoretic Preliminaries

We consider the NE solution concept as a rational action profile. A NE $a^{\mathrm{NE}} \in [0,1]^n$ is an action profile that satisfies

$$u_i(a_i^{\mathrm{NE}}, a_{\mathcal{N}_i}^{\mathrm{NE}}) \geq u_i(a_i, a_{\mathcal{N}_i}^{\mathrm{NE}}) \quad a_i \in [0,1], i \in \mathcal{N}. \qquad (20)$$

In other words, the individuals best-respond to the NE actions of other individuals—see (6) for the definition.

We can equivalently represent the NE definition in (20) by using the best response definition

$$a_i^{\mathrm{NE}} = BR_i(a_{\mathcal{N}_i}^{\mathrm{NE}}) \quad \text{for all } i \in \mathcal{N}. \qquad (21)$$

The notion of strictly dominated is defined in the following.

Definition 3 (Strictly dominated action): An action $a_i \in [0,1]$ is strictly dominated if and only if there exists an action $a_i' \in [0,1]$ such that

$$u_i(a_i', a_{\mathcal{N}_i}) > u_i(a_i, a_{\mathcal{N}_i}) \quad \text{for all } a_{\mathcal{N}_i} \in \mathcal{A}_{\mathcal{N}_i}. \qquad (22)$$

If an action a_i is strictly dominated then there exists a more preferable action a_i' for any circumstance. From (21), a strictly dominated action cannot be rational .

In the game Γ, we can iteratively remove the strictly dominated actions. This process is called the iterated elimination of strictly dominated strategies, defined below.

Definition 4 (Iterated elimination): Set the initial set of actions $A_i^0 = [0,1]$ for all i, and for any $k \in \mathbb{N}$ let

$$A_i^k = \{a_i \in A_i^{k-1} | \; a_i \text{ is not strictly dominated by any } a_{\mathcal{N}_i} \in A_{\mathcal{N}_i}^{k-1}\}.$$

We denote the set of player i's actions that survive the iterated elimination by $A_i^\infty := \bigcap_{k=0}^\infty A_i^k$. If A_i^∞ is a singleton (has a single element) for all $i \in \mathcal{N}$, then the game is dominance solvable, and has a unique NE given by the action profile that survives the iterated elimination process.

B. Convergence of the Local Learning Algorithm

The following theorem states that Algorithm 1 eliminates all strictly dominated actions in finite number of steps.

Theorem 2: Assume all players are undecided, i.e., $a_i^0 = \epsilon$ for $i \in \mathcal{N}$. Algorithm 1 converges in at most n iterations, that is, no player changes its action after the nth update. If all players are decided, i.e., $a_i^n = \{0,1\}$ for all $i \in \mathcal{N}$, the resultant action profile a^n is a NE. Otherwise, if a player $i \in \mathcal{N}$ is undecided, then there does not exist $a_i \in \mathcal{A}_i$ that can be strictly dominated.

Proof: We prove by induction the equivalence to the iterated elimination of strictly dominated strategies (Definition 4).

Given $a^0 = \vec{\epsilon}$, $\lceil a_{\mathcal{N}_i}^0 \rceil = \mathbf{1}_{|\mathcal{N}_i|}$. If $BR_i(\mathbf{1}_{|\mathcal{N}_i|}) = 1$, then it is best to play action 1 against all possible actions of neighboring players by $BR_i(a_{\mathcal{N}_i}) \geq BR_i(\lceil a_{\mathcal{N}_i}^{k-1} \rceil)$ given in (7). Hence, all actions in [0, 1) are strictly dominated by $a_i^1 = 1$ by (22). For $BR_i(\mathbf{1}_{|\mathcal{N}_i|}) = 0$, players remain undecided.

Assume at time k players that decided $a_i^{k-1} \in \{0,1\}$ have eliminated rest of their actions. Define the not strictly dominated action space of each player $i \in \mathcal{N}$ as $A_i^{k-1} = \{1\}$ if $a_i^{k-1} = 1$, $A_i^{k-1} = \{0\}$ if $a_i^{k-1} = 0$, and $A_i^{k-1} = [0,1]$ if $a_i^{k-1} = \epsilon$ [recall the notation in Definition (4)]. Using $BR_i(a_{\mathcal{N}_i}) \geq BR_i(\lceil a_{\mathcal{N}_i}^{k-1} \rceil)$ for $a_{\mathcal{N}_i} \in A_{\mathcal{N}_i}^{k-1}$, if $BR_i(\lceil a_{\mathcal{N}_i}^{k-1} \rceil) = 1$ then $a_i^k = 1$ strictly dominates all $a_i \in A_i^{k-1}$ by (22). Using $BR_i(a_{\mathcal{N}_i}) \leq BR_i(\lfloor a_{\mathcal{N}_i}^{k-1} \rfloor)$ for $a_{\mathcal{N}_i} \in A_{\mathcal{N}_i}^{k-1}$, if $BR_i(\lfloor a_{\mathcal{N}_i}^{k-1} \rfloor) = 0$ then $a_i^k = 0$ strictly dominates all $a_i \in A_i^{k-1}$ by (22). Further, we have $BR_i(\lceil a_{\mathcal{N}_i}^{k-1} \rceil) = 1$ if and only if (8) is true, and $BR_i(\lfloor a_{\mathcal{N}_i}^{k-1} \rfloor) = 0$ if and only if (9) is true. Hence, $a^k = \Phi_1(a^{k-1})$ is an action profile where decided players eliminate all strictly dominated actions given the not strictly dominated action space A^{k-1}. Algorithm 1 is equivalent to Definition 4 by induction.

Given this equivalence, if the game is dominance solvable, the algorithm converges to the unique NE. Otherwise, all decided players eliminate all of the actions except the action they selected, and undecided players cannot eliminate any actions from the their initial action space [0, 1].

Next, we prove convergence in n time steps. Suppose at time step k given a^{k-1}, there does not exist a player that switches from being undecided, i.e., $a_i^{k-1} = \epsilon$, to being decided, i.e., $a_i^k = \{0,1\}$. That is, if $a_i^k = a_i^{k-1}$ then $a_i^{k+1} = a_i^k$. Further, if a player is decided, it cannot change its action because all the other possible actions are dominated, that is, if $a_i^k = \{0,1\}$, then $a_i^{k+1} = a_i^k$. Given these two observations, at least one player has to switch to being decided at time $k-1$, in order for at least one player to become decided at time k given that it was undecided at time $k-1$. There can at most be n instances of switching from being undecided to being decided which is the case when the game is dominance solvable. Further there needs to be at least one switching happening at each time step for the updates to continue. Hence, the algorithm converges in at most n steps. ■

The proof above relies on showing that Algorithm 2 is a decentralized version of the iterated elimination of strictly dominated actions as given by Definition 4. The intuition for n step convergence is that at each step at least one player needs to eliminate its action using (8) or (9). If no player updates at a time step, the players stop updating because no new eliminations are triggered. Therefore, there could at most be n iterations to rule out n players one player at a time.

C. Proof of Lemma 2

Dominance solvability implies Algorithm 1 converges to a unique NE a^{NE} defined in (20). This means that $a_i^n \in \{0,1\}$. Either $a^{\mathrm{NE}} \in M(\mathcal{G}_B)$ or $a^{\mathrm{NE}} \notin M(\mathcal{G}_B)$. If the former holds, then the optimal control profile that minimizes the objective is the empty set ($\mathcal{X}_*^t = \emptyset$). If the latter holds, we need to control a nonempty set of players to achieve $a^t \in M(\mathcal{G}_B)$. Assume now the optimal control profile after time n is empty, i.e., $\mathcal{X}_*^t = \emptyset$ for $t > n$. Then because the game is dominance solvable, we have

$a^{2n} = \Phi_n(a^n) = a^{\text{NE}} \notin M(\mathcal{G}_B)$ where $\Phi_k(\cdot)$ is defined in (3). Hence, a control profile $\mathcal{X}_*^t = \emptyset$ for $t > n$ cannot be optimal because it is not feasible when a^{NE} is not feasible.

D. Proof of Lemma 3

Suppose the optimal policy Π^* to dynamic MPCAC in (11) is such that it does not converge to an equilibrium of the game $a^* \in M(\mathcal{G}_B)$. This implies at least one player is controlled for all $t \geq n$ to maintain MAC. Further, at least one player must be controlled at time n to satisfy the MAC constraint. Combining the cost for control before and after time n, $J_d(\Pi^*) \geq 1/n + 1$.

If there exists an equilibrium $a^* \in M(\mathcal{G}_B)$, then Algorithm 1 cannot converge to an action profile a^n with a link satisfying $(i,j) \in \mathcal{E}_B$ where $a_i^n + a_j^n > 1$, because it would imply the equilibrium action profile a^* is strictly dominated by an action profile, a^n. This means that Algorithm 1 can only eliminate possibly active links. Hence, the worst case scenario in terms of cost of control is when all players remain undecided as a result of Algorithm 1, i.e., $a^n = \vec{\epsilon}$. In this case, a feasible solution is one where we control the players in one type ($\mathcal{X}^t = \mathcal{S}_0$ or $\mathcal{X}^t = \mathcal{S}_1$) for two time steps $t = 0, 1$ with forced actions 0. This causes the players $i \notin \mathcal{X}^0$ to play action 1 by (5) at $t = 1$. By continuing at $t = 1$ with $\mathcal{X}^1 = \mathcal{X}^0$, the players $i \in \mathcal{X}^1$ take action 0 by (5), i.e., $a_i^1 = x_i^1 \delta_i^1 = 0$ for $i \in \mathcal{X}^1$, and $a_i^1 = y_i^1 = \Phi(a^0) = 1$ if $i \in\notin \mathcal{X}^1$. When we set $\mathcal{X}^t = \varnothing$ for $t > 2$, players do not change their actions because the resultant action profile is an equilibrium. By selecting the type satisfying $\min\{|\mathcal{S}_0|, |\mathcal{S}_1|\}$, our cost for this control policy is at most $2\lfloor n/2 \rfloor / n \leq 1$, from (11). This policy is an upper bound on the optimal policy. Consequently, $J_d(\Pi^*) \geq 1/n + 1$ cannot be optimal.

E. Proof of Corollary 1

When all players are decided in a single time step, all players decide on taking action 1 using (8), that is, $a = \mathbf{1}_n$ where $\mathbf{1}_n$ is the $n \times 1$ vector with all elements equal to one.

Claim 1: $\mathcal{X}_*^t = \emptyset$ for $t < n$.

Proof: Suppose there exists an optimal policy Π such that $\mathcal{X}_*^{\bar{t}} \neq \emptyset$ for $\bar{t} < n$. Given the controlled action profile at time $\bar{t}$, $a^{\bar{t}}$, we have the uncontrolled action profile at time $t+1$ as $y^{\bar{t}+1} = \mathbf{1}_n$. Hence, the $y^n = \mathbf{1}_n$. As a result, a policy where $\mathcal{X}^t = \emptyset$ for $t < n$ and $\mathcal{X}^t = \mathcal{X}_*^t$ for $t \geq n$ would be feasible and would incur a smaller cost than the policy Π by an amount $\frac{1}{n}\sum_{t=1}^{n-1} |\mathcal{X}_*^t|$. ■

We continue with the proof of Corollary 1. By Claim 1, no control action is taken until time n, where we have $y^n = \mathbf{1}_n$. The optimal control policy at time n is given by the following single time-step optimization:

$$\begin{aligned} &\min_{\mathcal{X}} |\mathcal{X}| \\ &\text{s.t. } a_i + a_j \leq 1 \quad \text{for all } (i,j) \in \mathcal{E}_B \\ &\qquad a_i = 1 - x_i \quad \text{for all } i \in \mathcal{N} \\ &\qquad a_i \in \{0,1\}. \end{aligned} \tag{23}$$

If we select $x_i = 1$, we must select $\delta_i = 0$ because $a_i^n = 1$ for all i. Hence, we have the second constraint from the controlled dynamics, $(1 - x_i)a_i^n + \delta_i x_i = 1 - x_i$. Define the dynamic control policy $\bar{\mathcal{X}}$ where $\bar{\mathcal{X}}^t = \emptyset$ for $t < n$, and we implement $\bar{\mathcal{X}}^n$, which is the solution to the above optimization problem, for $t \geq n$.

Suppose there exists $\tilde{\mathcal{X}}$ that achieves a lower cost than $\bar{\mathcal{X}}$ in the dynamic MPCAC problem. Given the first constraint above it is guaranteed that the controlled dynamics $\Phi_n(\vec{\epsilon}, \bar{\mathcal{X}}, \Delta) \in M(\mathcal{G}_B)$.

By Claim 1, it must be that $\tilde{\mathcal{X}}^t = \emptyset$ for $t < n$. Note that there cannot exist a policy at time n such that $|\tilde{\mathcal{X}}^n| < |\bar{\mathcal{X}}^n|$ because $\bar{\mathcal{X}}^n$ is an optimal solution of (23). If $|\tilde{\mathcal{X}}^n| > |\bar{\mathcal{X}}^n|$ then we can use $\bar{\mathcal{X}}$ to obtain a smaller cost ($\sum_{t=1}^n |\tilde{\mathcal{X}}^t| > \sum_{t=1}^n |\bar{\mathcal{X}}^t| = |\bar{\mathcal{X}}^n|$) for the first n time steps. Further, the uncontrolled action profile at time $n+1$ is given by $y^{n+1} = \Phi_1(a^n) = \mathbf{1}_n$ where controlled action of player i is given by $a_i^n = 1 - x_i^*$ for $x_i^* \in \bar{\mathcal{X}}^n$. Hence, there cannot exist a control policy $\tilde{\mathcal{X}}^t$ such that $|\tilde{\mathcal{X}}^t| < |\bar{\mathcal{X}}^t|$ for $t > n$ by the same reasoning as above. Combining the above findings, we have the dynamic MPCAC objective for the control policy $\tilde{\mathcal{X}}$ as $\frac{1}{n}|\tilde{\mathcal{X}}^n| + \lim_{T' \to \infty} \frac{1}{T'} \sum_{t=n+1}^{T'} |\tilde{\mathcal{X}}^t| > (1/n + 1)|\bar{\mathcal{X}}^n|$. This is a contradiction.

Substituting $a_i = 1 - x_i$ in the constraint $a_i + a_j \leq 1$, we get $x_i + x_j \geq 1$ when $x_i \in \{0,1\}$ and $x_j \in \{0,1\}$. Note that $\mathcal{E}_B = \mathcal{E}^n$ and $\mathcal{N}^n = \mathcal{N}$ given $a_i^n = 1$. Hence, the last constraint in (12) does not exist when $a^n = \mathbf{1}$. This shows that the optimization in (23) is equivalent to the optimization problem (12) when $a^n = \mathbf{1}$.

References

[1] C. Nowzari, V. M. Preciado, and G. J. Pappas, "Analysis and control of epidemics: A survey of spreading processes on complex networks," *IEEE Control Syst.*, vol. 36, no. 1, pp. 26–46, Feb. 2016.

[2] K. Zhang, W. Shi, H. Zhu, E. Dall'Anese, and T. Basar, "Dynamic power distribution system management with a locally connected communication network," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 4, pp. 673-687, Aug. 2018.

[3] M. Ye and G. Hu, "Game design and analysis for price-based demand response: An aggregate game approach," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 720–730, Mar. 2017.

[4] A. R. Hota and S. Sundaram, "Interdependent security games on networks under behavioral probability weighting," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 1, pp. 262–273, Mar. 2018.

[5] G. Bacci, S. Lasaulce, W. Saad, and L. Sanguinetti, "Game theory for networks: A tutorial on game-theoretic tools for emerging signal processing applications," *IEEE Signal Process. Mag.*, vol. 33, no. 1, pp. 94–119, Jan. 2016.

[6] J. Marden, G. Arslan, and J. Shamma, "Cooperative control and potential games," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 6, pp. 1393–1407, Dec. 2009.

[7] B. Swenson, S. Kar, and J. Xavier, "Empirical centroid fictitious play: An approach for distributed learning in multi-agent games," *IEEE Trans. Signal Process.*, vol. 63, no. 15, pp. 3888–3901, Aug. 2015.

[8] P. Yi and L. Pavel, "A distributed primal-dual algorithm for computation of generalized Nash equilibria via operator splitting methods," in *Proc. IEEE Conf. Decis. Control*, 2017, pp. 3841–3846.

[9] F. Parise, B. Gentile, S. Grammatico, and J. Lygeros, "Network aggregative games: Distributed convergence to Nash equilibria," in *Proc. IEEE Conf. Decis. Control*, 2015, pp. 2295–2300.

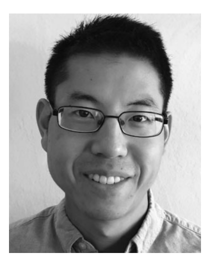
[10] J. Koshal, A. Nedić, and U. V. Shanbhag, "Distributed algorithms for aggregative games on graphs," *Oper. Res.*, vol. 64, no. 3, pp. 680–704, 2016.

[11] B. Swenson, C. Eksin, S. Kar, and A. Ribeiro, "Distributed inertial best-response dynamics," *IEEE Trans. Autom. Control*, vol. 63, no. 12, pp. 4294–4300, Dec. 2018.

[12] G. Scutari, D. P. Palomar, F. Facchinei, and J.-s. Pang, "Convex optimization, game theory, and variational inequality theory," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 35–49, May 2010.
[13] G. Hardin, "The tragedy of the commons," *Science*, vol. 162, no. 3859, pp. 1243–1248, 1968.
[14] J. R. Marden and T. Roughgarden, "Generalized efficiency bounds in distributed resource allocation," *IEEE Trans. Autom. Control*, vol. 59, no. 3, pp. 571–584, Mar. 2014.
[15] N. Li and J. Marden, "Designing games for distributed optimization," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 2, pp. 230–242, Apr. 2013.
[16] M.-F. Balcan, A. Blum, and Y. Mansour, "Improved equilibria via public service advertising," in *Proc. 20th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2009, pp. 728–737.
[17] M.-F. Balcan, A. Blum, and Y. Mansour, "Circumventing the price of anarchy: Leading dynamics to good behavior," *SIAM J. Comput.*, vol. 42, no. 1, pp. 230–264, 2013.
[18] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," *Theory Comput.*, vol. 11, no. 4, pp. 105–147, 2015.
[19] S. Grammatico, "Dynamic control of agents playing aggregative games with coupling constraints," *IEEE Trans. Autom. Control*, vol. 62, no. 9, pp. 4537–4548, Sep. 2017.
[20] J. R. Riehl, P. Ramazi, and M. Cao, "Incentive-based control of asynchronous best-response dynamics on binary decision networks," *IEEE Trans. Control Netw. Syst.*, vol. 6, no. 2, pp. 727–736, Jun. 2018.
[21] F. Fele, E. Debada, J. M. Maestre, and E. F. Camacho, "Coalitional control for self-organizing agents," *IEEE Trans. Autom. Control*, vol. 63, no. 9, pp. 2883–2897, Sep. 2018.
[22] S. Aral and P. S. Dhillon, "Social influence maximization under empirical influence models," *Nature Human Behav.*, vol. 2, pp. 375–382, 2018.
[23] P. N. Brown, H. Borowski, and J. R. Marden, "Security against impersonation attacks in distributed systems," *IEEE Trans. Control Netw. Syst.*, vol. 6, no. 1, pp. 440–450, Mar. 2018.
[24] C. Eksin, J. S. Shamma, and J. S. Weitz, "Disease dynamics in a stochastic network game: A little empathy goes a long way in averting outbreaks," *Scientific Rep.*, vol. 7, 2017, Art. no. 44122.
[25] C. McConnell, Y. Margalit, N. Malhotra, and M. Levendusky, "The economic consequences of partisanship in a polarized era," *Amer. J. Political Sci.*, vol. 62, no. 1, pp. 5–18, 2018.
[26] C. Hauert and M. Doebeli, "Spatial structure often inhibits the evolution of cooperation in the snowdrift game," *Nature*, vol. 428, no. 6983, pp. 643–646, 2004.
[27] D. Fudenberg and D. Levine, *The Theory of Learning in Games*. 1st ed. Cambridge, MA, USA: MIT Press, 1998.
[28] K. Paarporn and C. Eksin, "Incentive control in network anti-coordination games with binary types," in *Proc. IEEE 52nd Asilomar Conf. Signals, Syst. Comput.*, 2018, pp. 316–320.
[29] A. Bhalgat, T. Chakraborty, and S. Khanna, "Approximating pure Nash equilibrium in cut, party affiliation, and satisfiability games," in *Proc. 11th ACM Conf. Electron. Commerce*, 2010, pp. 73–82.
[30] J. Kun, B. Powers, and L. Reyzin, "Anti-coordination games and stable graph colorings," in *Proc. Int. Symp. Algorithmic Game Theory*, 2013, pp. 122–133.
[31] L. Cigler and B. Faltings, "Decentralized anti-coordination through multi-agent learning," *J. Artif. Intell. Res.*, vol. 47, pp. 441–473, 2013.
[32] D. Monderer and L. S. Shapley, "Potential games," *Games Econ. Behav.*, vol. 14, no. 1, pp. 124–143, 1996.
[33] C. Eksin, "Control of stochastic disease network games via influential individuals," in *Proc. IEEE Conf. Decis. Control*, 2019, pp. 6893–6898.
[34] V. V. Vazirani, *Approximation Algorithms*. Berlin, Germany: Springer-Verlag, 2013.
[35] G. L. Nemhauser and L. A. Wolsey, *Integer Programming and Combinatorial Optimization*. Chichester, U.K.: John Wiley, 1988.
[36] M.-F. Balcan, S. Krehbiel, G. Piliouras, and J. Shin, "Minimally invasive mechanism design: Distributed covering with carefully chosen advice," in *Proc. IEEE 51st Conf. Decis. Control*, 2012, pp. 2690–2695.

Ceyhun Eksin (Member, IEEE) received the B.S. degree in control engineering from Istanbul Technical University, Istanbul, Turkey, in 2005, the M.S. degree in industrial engineering from the Boğaziçi University, Istanbul, Turkey, in 2008, the M.A. degree in statistics from Wharton Statistics Department, University of Pennsylvania, Philadelphia, PA, USA, in 2015, and the Ph.D. degree in electrical and systems engineering from the Department of Electrical and Systems Engineering in 2015.

He was a Postdoctoral Researcher jointly affiliated with the Schools of Biological Sciences, and Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. He is currently an Assistant Professor with the Department of Industrial and Systems Engineering, Texas A&M University, College Station, TX, USA. His research interests include the applications of game theory, control theory, and statistical signal processing to networked multiagent systems, game theoretic learning, decentralized optimization, autonomous teams, and epidemics.



Keith Paarporn (Member, IEEE) received the B.S. degree in electrical engineering from the University of Maryland, College Park, MD, USA, in 2013, and the M.S. and Ph.D. degrees in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2016 and 2018, respectively.

He is currently a Postdoctoral Scholar with the Electrical and Computer Engineering Department, University of California, Santa Barbara, Santa Barbara, CA, USA. His research interests include application of game theory and control theory to multiagent systems in engineered, social, and biological domains.