Towards Optimal Control of Evolutionary Games on Networks

Riehl, James R.; Cao, Ming

Published in:
IEEE Transactions on Automatic Control

DOI:
10.1109/TAC.2016.2558290

Towards Optimal Control of Evolutionary Games on Networks

James R. Riehl† and Ming Cao†

***Abstract*—We investigate the control of evolutionary games on networks, in which each edge represents a two-player repeating game between neighboring agents. After each round of games, agents can update their strategies based on local payoff and strategy information, while a subset of agents can be assigned strategies and thus serve as control inputs. We seek here the smallest set of control agents needed to drive the network to a desired uniform strategy state. After presenting exact solutions for complete and star networks and describing a general solution approach that is computationally practical only for small networks, we design a fast algorithm for approximating the solution on arbitrary networks using a weighted minimum spanning tree and strategy propagation algorithm. We show that the resulting approximation is exact for certain classes of games on complete and star networks. Finally, simulations demonstrate that the algorithm yields near-optimal solutions in more general cases, although it performs best for coordination games.**

I. Introduction

One of the most remarkable trends of the modern era is the rapid growth in connectivity of society and technology. Accompanying this is an equally fast growth in complexity of the systems we rely upon, resulting in a whole new set of challenges that engineers must overcome in the analysis of large dynamic networks and the design of systems to interact with and regulate them. This helps explain the recent boom in research on networked systems and control theory in applications ranging from distributed sensing and robotics to epidemic control and human decision making. Control design for networked multi-agent systems traditionally involves solving optimization problems, sometimes centralized and other times distributed, but very often with a shared global objective. However, when the agents have different or even competing objectives, as is often the case, each agent must take into account the actions of its competitors and single-objective optimization methods fail. Game theory has long been used to tackle these kinds of problems on a small scale and where the agents are assumed to be perfectly rational [1], but for larger scale and more complex systems, these assumptions often no longer hold and the strategy dynamics are better modeled by *evolutionary game theory* [2].

A key innovation of evolutionary game theory is that rather than assuming perfect rationality in a large group of players, we allow strategies to propagate dynamically through the population based on the payoffs acquired by the agents using these strategies [3]. In biological terms, the mechanism for this propagation is an evolutionary *survival of the fittest* process in which reproductive potential is directly related to the payoffs of the game. The agents need not be simple organisms however; more complex systems such as human social networks and robotic networks can also fit well into an evolutionary game framework, but here the strategy propagation mechanism can be better thought of as a learning process or update rule. Indeed there have been numerous studies on the best strategies and expected outcomes of evolutionary games for various types of population structures, payoff functions, and update rules [2][4][5]. One of the well-established findings in the field is that evolutionary games often lead to complex and undesired outcomes with respect to the population as a whole, such as in *prisoner's dilemma* games or *tragedy of the commons*, where selfishness tends to prevail over cooperation, potentially resulting in congestion, inefficiency, and depletion of resources [6][7]. There is consequently a strong incentive to devise methods for influencing evolutionary games in order to catalyze better collective outcomes in populations.

In this work we seek solutions or approximations to a minimal agent control problem on a broad class of networks and payoff matrices under imitative evolutionary game dynamics. The complex structures of networked games, which is a challenging obstacle in the analysis of population dynamics, also offers a unique opportunity. Depending on the topology of the network, there are very likely to be certain agents that have more influence on the network than others, and may thus be able to achieve large shifts in strategy distribution at relatively low cost. Hence one of the primary challenges we address in this paper is to identify which agents are the most important to control such that the network can most efficiently be driven towards a desired strategy state.

*The work was supported in part by the European Research Council (ERCStG-307207) and the EU INTERREG program under the auspices of the SMARTBOT project.

†Faculty of Mathematics and Natural Sciences, ENTEG, University of Groningen, The Netherlands, {j.r.riehl, m.cao}@rug.nl

II. Evolutionary Game Framework

The framework used in this paper consists of a network, payoff matrix, and strategy update dynamics. We define each of these concepts and related notation here.

Network and single-round payoffs: Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote an undirected network whose node set $\mathcal{V} = \{1, \ldots, n\}$ corresponds to agents engaged in multiple 2-player symmetric games defined by the edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. At each time step, the agents play a single strategy from the finite set $\mathcal{S} := \{A, B, \ldots\}$ against all neighbors and receive payoffs upon completion of each game according to the matrix M of size $|\mathcal{S}| \times |\mathcal{S}|$. We denote the strategy state by $x(t) = [x_1(t), \ldots, x_n(t)]^T$, where $x_i(t) \in \mathcal{S}$ is the strategy of agent i at time t. Total payoffs are given by

$$y_i(t) = w_i \sum_{j \in \mathcal{N}_i} M_{x_i(t), x_j(t)}, \tag{1}$$

where $\mathcal{N}_i := \{j \in \mathcal{V} : \{i, j\} \in \mathcal{E}\}$ is the neighbor set of agent i and the most common values for the weights w_i are 1 for cumulative and $\frac{1}{|\mathcal{N}_i|}$ for averaged payoffs.

Strategy update dynamics: A fundamental concept of evolutionary games is that better performing strategies are adopted more often. This means rather than rationally choosing best-response strategies, agents imitate strategies in their neighborhood that result in higher payoffs. We capture this dynamic with a strategy update rule that is a function of neighboring strategies and payoffs:

$$x_i(t+1) = f\left(\{x_j(t), y_j(t) : j \in \mathcal{N}_i \cup \{i\}\}\right).$$

The only restrictions we make on the update rule is that it is *payoff monotone*, *i.e.* players only switch to strategies with which at least one agent in the neighborhood achieves a greater payoff [2], and *persistent*, *i.e.* if there exists a better performing strategy in an agent's neighborhood, then the agent will switch to that strategy with a probability that is lower bounded by $\epsilon > 0$.

One example of such dynamics is the *proportional imitation* rule, where each agent chooses a neighbor randomly and if this neighbor received a higher payoff in the previous round by using a different strategy, then the agent will switch with a probability proportional to the payoff difference. This widely studied model has some nice properties, in particular that the strategy distributions in well-mixed populations using proportional imitation can be approximated by the standard replicator dynamics [8]. The *proportional imitation* rule can be expressed as follows:

$$p\big(x_i(t+1) = x_j(t)\big) := \left[\frac{\lambda}{|\mathcal{N}_i|}\big(y_j(t) - y_i(t)\big)\right]_0^1 \tag{2}$$

for each agent $i \in \mathcal{V}$ where $j \in \mathcal{N}_i$ is a randomly chosen neighbor, $\lambda > 0$ is an arbitrary rate constant, and the notation $[z]_0^1$ indicates $\max(0, \min(1, z))$.

Strategy update with control agents: We add to this conventional framework a set of *control agents* $\mathcal{L} \subseteq \mathcal{V}$ whose strategy can be externally manipulated, either through direct control or by adding neighbors with artificially high payoffs, resulting in the following controlled evolutionary game dynamics:

$$x_i(t+1) = \begin{cases} x_i^*, & i \in \mathcal{L} \\ f(\cdot), & \text{otherwise} \end{cases}, \tag{3}$$

where x_i^* is the desired strategy for agent i and $f(\cdot)$ denotes the unforced strategy update rule. The combination of a network, payoff matrix, and update rule forms what we call a *network game* $\Gamma := (\mathcal{G}, M, f)$.

III. Problem Formulation

Now that we have a general dynamic evolutionary game model with control inputs, we are interested in how one can influence the network through efficient use of these inputs in order to achieve some desired outcome of strategies. Before stating the problem, we define the following notion of probabilistic convergence.

Definition 1. *We say that $x_i(t)$ converges* almost surely *to the strategy X if $p[\lim_{t\to\infty} x_i(t) = X] = 1$, and indicate this with the shorthand notation $x_i(t) \to X$.*

In this work, we focus on achieving consensus in strategy A and pose what we call the *Minimum Agent Consensus Control* (MACC) problem.

Problem 1 (**MACC**). *Given a network game Γ and initial strategy state $x(0)$, find the smallest set of control agents $\mathcal{L}$ such that $x_i(t) \to A$ for each agent $i \in \mathcal{V}$.*

IV. Related Work

Although the literature on evolutionary games on networks in the absence of control is quite extensive [2][9], we focus here on attempts to manipulate the outcome of evolutionary games with some external control, which is only beginning to see increased attention.

A natural starting point is the case of infinite and well-mixed populations, where one can model the dynamics of strategy proportion as a system of first order ODEs. In the context of replicator dynamics, Kanazawa et al. proposed using taxes and subsidies imposed by a government or central administrator to alter the game payoffs such that the players will converge to a desired strategy [10]. They derive criteria for stabilization of prescribed equilibria and simulate the approach on a congestion control problem. Along similar lines, but using logit

choice dynamics, [11] introduced pricing schemes to promote efficient choices on roadway networks, also extended to more general economic contexts in [12].

Finite populations with complex structure are better able to capture the complex interactions in real multi-agent networks, but this increases the difficulty of analysis. In a continuous-time setting with best response dynamics, [13] characterized the number of control nodes required to stabilize desired equilibrium states on path and star networks. Also, a recent simulation study showed that controlling agents with higher degree induced a higher frequency of cooperation than randomly controlling agents in a prisoner's dilemma game with imitative dynamics on scale-free networks [14].

Finally, Cheng *et al.* presented a framework for the control of networked evolutionary games using large-scale logical dynamic networks to model transitions between all possible strategy states for various update rules and derived equivalent conditions for reachability and consensus of strategies on a network given a particular set of control nodes [15]. However, the optimal control of such systems remains a challenging open problem, and is the primary focus of this paper, which builds upon earlier work on deterministic [16] and two-strategy stochastic [17] network games with new analytical results as well an extension to the multiple-strategy stochastic setting for arbitrary networks.

V. Exact Solutions

One approach to solving the MACC problem exactly involves building very large state transition probability matrices similar to those used in [15], but for a sequence of candidate control sets $\mathcal{L}$. Since the goal is to find the smallest control set for which the network converges to the desired strategy state, one could start by checking $\mathcal{L} := \emptyset$ and proceed to all one-agent sets, two-agent sets, etc. until the desired convergence occurs. In a network game with n agents and s strategies, the total number of possible strategy states is s^n, and therefore the size of the state transition matrix $\Phi_{\mathcal{L}}$ is $s^n \times s^n$. Once the transition probability matrix is constructed, the set of all stationary states can be computed by finding the eigenvectors corresponding to eigenvalues of $\Phi_{\mathcal{L}}$ equal to one. If the desired strategy state is the *only* stationary state that is accessible from $x(0)$, then a solution has been reached. Otherwise, the incremental search must proceed until this condition is satisfied. Since Φ_L becomes prohibitively large even for medium sized networks, the computational complexity of this approach is clearly too high to be practical in general. Whether or not this problem belongs to a well-known class of difficult problems such as NP-hard remains an open research problem. If we restrict to some highly structured classes of networks however, exact analytical solutions are obtainable.

A. Two-strategy games on complete networks

Consider a completely connected network and a two-strategy payoff matrix:

$$M = \begin{array}{c c} & \begin{array}{cc} A & B \end{array} \\ \begin{array}{c} A \\ B \end{array} & \begin{pmatrix} a & b \\ c & d \end{pmatrix} \end{array}.$$

A necessary condition for all agents initially playing B to eventually switch to A is that the initial payoff of agents playing A must be strictly greater than that of agents playing B. Denoting these payoffs by y_A and y_B and taking $w_i = 1$, we can write this condition as follows:

$$y_A = a(n_A - 1) + bn_B > cn_A + d(n_B - 1) = y_B,$$

where n_A and n_B denote the number of agents initially playing A and B. Rearranging the terms yields

$$n_A(a + d - b - c) > n(d - b) + a - d, \tag{4}$$

where we used the fact that $n_A + n_B = n$. We immediately see that the characteristics of the game change significantly depending on the sign of the term $\delta = a + d - b - c$. If $\delta > 0$, an agent switching from B to A increases the payoff to agents already playing A, and we say that the game is *coordinating* in strategy A. On the other hand if $\delta < 0$, an agent switching from B to A decreases the payoff to agents playing A and we say the game is *anti-coordinating* in A. Finally, if $\delta = 0$ then the number of agents playing A has no net effect on the payoff to agents playing A and we say the game is *neutrally-coordinating* in A. Let $\gamma = n(d-b)+a-d$. Using condition (4), we can solve for n_A on complete networks for each of these cases.

Proposition 1. *The solution to the MACC problem on complete networks with $x_i(0) = B$ for all agents is given in the table below.*

$\delta > 0$	$\lvert\mathcal{L}^*\rvert = \begin{cases} \min\left(\lfloor 1 + \frac{\gamma}{\delta} \rfloor, n\right), & \gamma \geq 0 \\ 1, & \gamma < 0 \end{cases}$
$\delta = 0$	$\lvert\mathcal{L}^*\rvert = \begin{cases} n, & \gamma \geq 0 \\ 1, & \gamma < 0 \end{cases}$
$\delta < 0$	$\lvert\mathcal{L}^*\rvert = \begin{cases} n, & \gamma \geq -\delta \text{ or mixed eq exists,} \\ 1, & \text{otherwise} \end{cases}$

Proof: For $\delta > 0$, consider the energy-like function $V(x) := \lvert\{i \in \mathcal{V} : x_i = B\}\rvert$. Let $\mathcal{S}^* := \{x \in \mathcal{S}^n : V(x) < \frac{\gamma}{\delta}\}$. Due to (4) and the payoff monotonicity

assumption, $V(x(t+1)) \leq V(x(t))$ for all $x(t) \in \mathcal{S}^*$, which also implies that $x(t+1) \in S^*$. Due to the persistence assumption, $p\left[V(x(t+1)) = V(x(t))\right] < (1-\epsilon)^{V(x(t))}$. Let $t^0 = 0$. It follows that $p\left[\forall t > t^0, V(x(t)) = V(x(t^0))\right] = 0$, which implies that $p\left[\exists t^1 > t_0 : V(x(t^1)) < V(x(t^0))\right] = 1$. By repeating this procedure, since $V(x)$ is finite and takes discrete values, there must exist a t^* such that $V(x(t^*)) = 0$, which implies that $x_i(t^*) = A$ for all $i \in \mathcal{V}$. Hence we have shown that $x(0) \in \mathcal{S}^* \implies x_i(t) \to A$. Due to payoff monotonicity, if $x(0) \notin \mathcal{S}^*$ then $x(t) \notin \mathcal{S}^*$ for all $t > 0$ and there will be no convergence. Therefore the minimum number of control agents needed to ensure that $x(0) \in \mathcal{S}^*$ is 1 for $\gamma < 0$ and otherwise $\min\left(\lfloor 1 + \frac{\gamma}{\delta} \rfloor, n\right)$.

For $\delta = 0$, (4) is either never satisfied or always satisfied depending on whether $\gamma < 0$ or $\gamma \geq 0$ and thus the size of $\mathcal{S}^*$ equals $\emptyset$ or $\mathcal{S}^n$, respectively. Therefore the minimum number of control agents needed to ensure that $x(0) \in \mathcal{S}^*$ is n or 1.

Finally, for $\delta < 0$, if $\gamma \geq -\delta$ then there is clearly no value $n_A > 0$ which will satisfy (4) and n control agents are needed. If $\gamma < -\delta$, then $n_A = 1$ satisfies (4). However, if some but not all of the remaining agents switch to A, then (4) is no longer satisfied and the payoffs to the agents playing B will be greater. The state will thus converge either to the desired equilibria of all A, or a mixed equilibria, which exists if and only if $\frac{\gamma}{\delta} < n$ and $\delta \mod \gamma = 0$. ∎

B. Two-strategy games on star networks

Let us now consider networks with a star topology – trees spanning one level with a root node connected to $n-1$ branch nodes. In this case, for the B branch nodes to switch to A, we must have the root node playing A and $a(n_A - 1) + b(n - n_A) > c$. Let $\delta = a - b$ and $\gamma = b(1-n) + c$. Then the inequality that must hold is $n_A\delta > \gamma$. Define the following quantities:

$$q = \left[\lfloor 1 + \frac{\gamma}{\delta} \rfloor\right]_0^{n-1}, \quad y_r^B = cq + d(n - q - 1)$$

Proposition 2. *The solution to the MACC problem on star networks with $x_i(0) = B$ for all agents is given in the Table I.*

We refer the reader to [18] for the proof, which uses very similar ideas to the proof of Proposition 1.

VI. Approximation for the General Case

Although it is much more difficult to solve the problem on networks of arbitrary structure, in this section we show how similar ideas to the solution for star networks can be used to approximate the solutions on trees as well as arbitrary networks. Originally designed for tree networks in [19], we extended the approach to two-strategy games on arbitrary networks in [17]. Here we futher extend the approach to games with an arbitrary number of strategies.

TABLE I

$\delta > 0$	$\lvert\mathcal{L}^*\rvert = \begin{cases} 1, & \gamma < 0 \\ q, & \gamma \geq 0, y_B < b \\ 1+q, & \gamma \geq 0, y_B \geq b \end{cases}$
$\delta = 0$	$\lvert\mathcal{L}^*\rvert = \begin{cases} 1, & \gamma \geq 0 \\ n, & \gamma < 0 \end{cases}$
$\delta < 0$	$\lvert\mathcal{L}^*\rvert = \begin{cases} 1, & n = 2 \text{ or } \frac{\gamma}{\delta} > n-2, \\ n, & \text{otherwise} \end{cases}$

For convenience in describing the algorithm, we use the analogy of a family tree starting from a single common ancestor or *root*, and consisting of successive generations or *levels* of parents and respective children. Algorithm 1 achieves its objective by working from the bottom of the tree towards the top (root), using the following procedure to ensure that the desired strategy can propagate in the reverse direction (down the tree) at each level. For each parent agent, add children to the control set in decreasing order of a switching threshold (to be defined shortly) until all remaining children will eventually switch to A once the agents in higher generations are playing A.

Before introducing the formal algorithm, we need to define a few quantities and agent sets. First we choose an arbitrary root agent r. Since any agent of a tree can serve as the root, we will take $r = 1$ without loss of generality. Denoting the number of levels by n_ℓ, let $g : \mathcal{V} \to \{0, \ldots, n_\ell\}$ be the mapping of agents to their level in the tree rooted at r, which is equivalent to the number of edges in the shortest path from the agent to the root. Let $\mathcal{V}_\ell := \{i \in \mathcal{V} : g(i) = \ell\}$ denote the set of all agents on level ℓ, and let $\mathcal{C}_p := \{c \in \mathcal{V}_{g(p)+1} : \{p, c\} \in \mathcal{E}\}$ denote the set of children of a given agent p. Finally, we define a function $\rho : \mathcal{V} \to V \cup \{0\}$ mapping each agent to its parent except for the root agent which is mapped to zero.

Let $\mathcal{T}$ denote a minimum spanning tree of the original network $\mathcal{G}$ using edge weights equal to the reciprocal of the sum of the degrees of the adjacent nodes. Although the modified algorithm will work with any spanning tree,

we choose these weights in order to retain the edges adjacent to the highest-degree agents because these are likely to be the most influential agents.

In several steps of Algorithm 1, we check the maximum or minimum achievable payoff for an agent in a given strategy configuration. Let $\mathcal{X}, \mathcal{Y} \subseteq \mathcal{S}$ denote generic strategy subsets assigned to agent i and neighbor $j \in \mathcal{N}_i \cup \{0\}$, respectively. We define the following payoff bounds for agent i:

$$\hat{y}_{i,j}^{\mathcal{X},\mathcal{Y}} := w_i \Big(\sum_{k \in \mathcal{N}_i^{-j}} \max_{\substack{X \in \mathcal{X} \\ Z \in \Omega_k}} M_{X,Z} + \delta_i \max_{\substack{X \in \mathcal{X} \\ Y \in \mathcal{Y}}} M_{X,Y} \Big), \quad (5)$$

$$\check{y}_{i,j}^{\mathcal{X},\mathcal{Y}} := w_i \Big(\sum_{k \in \mathcal{N}_i^{-j}} \min_{\substack{X \in \mathcal{X} \\ Z \in \Omega_k}} M_{X,Z} + \delta_i \min_{\substack{X \in \mathcal{X} \\ Y \in \mathcal{Y}}} M_{X,Y} \Big), \quad (6)$$

where $\mathcal{N}_i^{-j} := \mathcal{N}_i - \{j\}$ denotes the set of all neighbors of agent i excluding j. In the case of singleton sets $\mathcal{X} = \{X\}$ or $\mathcal{Y} = \{Y\}$, we also use the shorthand notation $\hat{y}_{i,j}^{X,Y}$, and we denote the exclusion of a single strategy by ${}^{-}X := \{Y \in \mathcal{S} : Y \neq X\}$. Next we define the reachable strategy sets Ω_i.

Since the spanning tree does not capture all possible interactions in an arbitrary network, we need a method for accounting for additional strategy transitions which may occur. Let Ω_i denote the set of strategies that agent i might play in the course of the game. Let $\tilde{\mathcal{L}} \subseteq \mathcal{V}$ denote a set of candidate control agents that will be used in the process of Algorithm 1. The sets Ω_i are computed through an iterative strategy propagation algorithm, defined in [17].

1 $\hat{\mathcal{L}} := \{\mathrm{r}\}$
2 $\tilde{\mathcal{L}} := \mathcal{V} - \{r\}$
3 $\ell := n_\ell$
4 **while** $\ell > 0$ **do**
5   **foreach** $p \in \mathcal{V}_{\ell-1}$ **do**
6         $\tilde{\mathcal{L}} := \tilde{\mathcal{L}} - \mathcal{C}_p$
7   Update the sets Ω_i using Alg. **??**
8         $c^* := \arg\max_{c \in \mathcal{C}_p \cap \Omega_{-A}} \left(\hat{y}_{c,p}^{-A,A} \right)$
9   **while** $\check{y}_{p,\rho(p)}^{A,A} \leq \hat{y}_{c^*,p}^{-A,A}$ **do**
10             $\hat{\mathcal{L}} := \hat{\mathcal{L}} \cup \{c^*\}$
11             $c^* := \arg\max_{c \in \mathcal{C}_p \cap \Omega_{-A}} \left(\hat{y}_{c,p}^{-A,A} \right)$
12   **end**
13   **end**
14     $\ell := \ell - 1$
15 **end**

Algorithm 1: Computes an approximately minimal set of control agents $\hat{\mathcal{L}}$ needed to drive an arbitrary network to the desired strategy A.

Theorem 1. *Given a network game Γ and initial strategy state $x(0)$, Algorithm 1 computes a sufficient set of control agents $\hat{\mathcal{L}}$ such that $x_i(t) \to A$ for all agents $i \in \mathcal{V}$.*

Proof: It suffices to show that for all $i \in \mathcal{V}$, if there exists $\tau \geq 0$ such that $x_i(\tau) \neq A$ then there exists $\tau' > \tau$ such that $x_i(t) = A$ for all $t \geq \tau'$. Working downwards from the root r of the spanning tree $\mathcal{T}$, consider an agent i such that $i \in \mathcal{C}_r$. Using the condition in step 9 with Lemma 1 in [17] to bound the payoffs, we know that either $i \in \hat{\mathcal{L}}$ and $x_i(t) = A$ for all $t \geq 0$, or $\check{y}_{r,0}^{A,A} > \hat{y}_{i,r}^{-A,A}$, which implies that $y_r(0) > y_i(0)$, and thus by the persistence of the strategy update rule, $x_i(t) \to A$. Similarly, for an agent i on arbitrary level ℓ, we have either $i \in \hat{\mathcal{L}}$ or $\check{y}_{\rho(i),i}^{A,A} > \hat{y}_{i,\rho(i)}^{-A,A}$, which implies that if there exists a time τ'' such that $x_{\rho(i)}(\tau'') = A$ then $y_{\rho(i)}(\tau'') > y_i(\tau'')$ and thus $x_i \to A$. We now have by induction that $x_i \to A$ for all agents $i \in \mathcal{V}$. ∎

Remark 1. *For maximum efficiency, the payoff bounds computed in the strategy propagation algorithm can be reused both upon subsequent calls to Alg-SP and in the remaining steps of Algorithm 1. If they are not, then the worst case computational complexity is dominated by the strategy propagation which can be performed up to n times, yielding the conservative estimate of $O(mn^2)$, where m is the number of edges.*

VII. Results: Complete and Star Networks

An important test of any approximation method is to check the results on cases for which one can derive the exact solutions analytically. Indeed the proposed algorithms compute exact solutions for classes of two-strategy games on star, complete, and ring networks [17].

Corollary 1. *Algorithm 1 computes the exact solution to the MACC problem on star networks for two-strategy simple coordination games.*

Corollary 2. *Algorithm 1 computes the exact solution to the MACC problem on complete networks for two-strategy games that are coordinating in A.*

VIII. Simulations

Although it is encouraging that Algorithm 1 produces the expected results for certain classes of network games, the purpose and strength of the general approach is that it can be applied to arbitrary networks and payoff matrices, and networks of much larger size than could be computed exactly in a reasonable amount of time. In this section, we use simulations to test the accuracy of

the approximation on small geometric random networks using the exact solutions. We start with all agents playing B and use the proportional imitation rule (2) with $\lambda = \frac{1}{4}$. We consider three different types of games: stag hunt (SH – a coordination game), prisoner's dilemma (PD – a game that is neutrally-coordinating in A), and snow drift (SD - an anti-coordination game). We use the following payoff matrices for SH, PD, and SD, respectively.

$$\begin{array}{c} \\ A \\ B \end{array}\!\!\begin{array}{c} \begin{array}{cc} A & B \end{array} \\ \begin{pmatrix} 2 & 0 \\ 1 & 1 \end{pmatrix} \end{array}, \quad \begin{array}{c} \\ A \\ B \end{array}\!\!\begin{array}{c} \begin{array}{cc} A & B \end{array} \\ \begin{pmatrix} 4 & -1 \\ 5 & 0 \end{pmatrix} \end{array}, \quad \begin{array}{c} \\ A \\ B \end{array}\!\!\begin{array}{c} \begin{array}{cc} A & B \end{array} \\ \begin{pmatrix} 3 & 1 \\ 5 & 0 \end{pmatrix} \end{array}.$$

Next, we test the algorithm on small geometric random networks created by placing ten agents in the unit square uniformly at random and connecting any two agents that lie closer than a distance of 0.4 from each other. We compare the approximate solutions of Algorithm 1 to exact solutions computed by the method described in Section V. We use small networks because of the extreme computational complexity involved in computing the exact solutions.

TABLE II
ACCURACY ON SMALL GEOMETRIC RANDOM NETWORKS

Game	mean($\|\hat{\mathcal{L}}\|$)	mean($\|\hat{\mathcal{L}}\|-\|\mathcal{L}^*\|$)	$E[\hat{t}]$	$E[t^*]$
SH	4.27	1.01	1.63	5.52
PD	8.01	1.25	1.27	49.4
SD	8.31	4.34	1.32	289

Table II lists the mean difference between the size of the approximate minimum control set $\hat{\mathcal{L}}$ and the true minimum $\mathcal{L}^*$. Much of the difference in the SD results is attributable to the anti-coordinating effect resulting in a tradeoff between size of the control set and the expected convergence times $E[\hat{t}]$ and $E[t^*]$ listed in columns 3 and 4.

IX. Conclusions and Future Work

In this paper, we posed a minimum-agent control problem for a class of evolutionary games on networks. After first presenting exact solutions to the problem for two-strategy games on complete and star networks as well as a general solution approach that is computationally complex, we introduced a fast algorithm for approximating the solution on arbitrary networks and payoff matrices. The proposed algorithm computes the exact solutions for a class of games on complete and star networks, and we demonstrated via simulations that the algorithm is quite accurate on more small geometric random networks, particularly for games that are coordinating in the desired strategy. Interesting research directions for the future include relaxing the assumption of payoff monotonicity towards development of a more robust result, allowing for dynamic control sequences, and investigating targeted payoff control as an alternative to direct strategy control.

References

[1] R. B. Myerson, *Game theory*. Harvard university press, 2013.

[2] G. Szabó and G. Fáth, "Evolutionary games on graphs," *Physics Reports*, vol. 446, no. 4, pp. 97–9216, 2007.

[3] W. H. Sandholm, *Population games and evolutionary dynamics*. MIT press, 2010.

[4] C. E. Tarnita, T. Antal, H. Ohtsuki, and M. A. Nowak, "Evolutionary dynamics in set structured populations," *Proceedings of the National Academy of Sciences*, vol. 106, no. 21, pp. 8601–8604, 2009.

[5] R. J. Smith, J. T. Okano, J. S. Kahn, E. N. Bodine, and S. Blower, "Evolutionary dynamics of complex networks of hiv drug-resistant strains: the case of san francisco," *Science*, vol. 327, no. 5966, pp. 697–701, 2010.

[6] W. B. Liebrand, H. A. Wilke, R. Vogel, and F. J. Wolters, "Value orientation and conformity a study using three types of social dilemma games," *Journal of Conflict Resolution*, vol. 30, no. 1, pp. 77–97, 1986.

[7] E. Ostrom, "Tragedy of the commons," *The New Palgrave Dictionary of Economics*, pp. 3573–3576, 2008.

[8] K. H. Schlag, "Why imitate, and if so, how?: A boundedly rational approach to multi-armed bandits," *Journal of Economic Theory*, vol. 78, no. 1, pp. 130–156, 1998.

[9] M. A. Nowak, C. E. Tarnita, and T. Antal, "Evolutionary dynamics in structured populations," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 365, no. 1537, pp. 19–30, 2010.

[10] T. Kanazawa, T. Misaka, T. Ushio, and Y. Fukumoto, "A control method of selfish routing based on replicator dynamics with capitation tax and subsidy," in *IEEE CCA & ISIC*, 2009, pp. 249–254.

[11] W. H. Sandholm, "Evolutionary implementation and congestion pricing," *The Review of Economic Studies*, vol. 69, no. 3, pp. 667–689, 2002.

[12] ——, "Pigouvian pricing and stochastic evolutionary implementation," *Journal of Economic Theory*, vol. 132, no. 1, pp. 367–382, 2007.

[13] A. Khanafer, T. Başar, and B. Gharesifard, "Stability properties of infected networks with low curing rates," in *American Control Conference (ACC), 2014*. IEEE, 2014, pp. 3579–3584.

[14] W. Du, H. Zhou, Z. Liu, and X. Cao, "The effect of pinning control on evolutionary prisoner's dilemma game," *Modern Physics Letters B*, vol. 24, no. 25, pp. 2581–2589, 2010.

[15] D. Cheng, F. He, H. Qi, and T. Xu, "Modeling, analysis and control of networked evolutionary games," *IEEE Transactions on Automatic Control*, vol. PP, no. 99, pp. 1–1, 2015.

[16] J. Riehl and M. Cao, "Towards control of evolutionary games on networks," in *The 53rd IEEE Conference on Decision and Control*, 2014.

[17] ——, "Control of stochastic evolutionary games on networks," in *The 5th IFAC Workshop on Distributed Estimation and Control in Networked Systems*, 2015.

[18] ——, "Towards optimal control of evolutionary games on networks," Tech. Rep., 2015.

[19] ——, "Minimal-agent control of evolutionary games on tree networks," in *21st Int. Symp. on Mathematical Theory of Networks and Systems*, 2014.