Dynamics at the Boundary of Game Theory and Distributed Computing

AARON D. JAGGARD, U.S. Naval Research Laboratory
NEIL LUTZ, Rutgers University
MICHAEL SCHAPIRA, Hebrew University of Jerusalem
REBECCA N. WRIGHT, Rutgers University

We use ideas from distributed computing and game theory to study dynamic and decentralized environments in which computational nodes, or decision makers, interact strategically and with limited information. In such environments, which arise in many real-world settings, the participants act as both economic and computational entities. We exhibit a general non-convergence result for a broad class of dynamics in asynchronous settings. We consider implications of our result across a wide variety of interesting and timely applications: game dynamics, circuit design, social networks, Internet routing, and congestion control. We also study the computational and communication complexity of testing the convergence of asynchronous dynamics. Our work opens a new avenue for research at the intersection of distributed computing and game theory.

CCS Concepts: • **Theory of computation → Distributed computing models**; **Convergence and learning in games**; • **Networks →** *Routing protocols*;

Additional Key Words and Phrases: Adaptive heuristics, self stabilization, game dynamics

ACM Reference format:
Aaron D. Jaggard, Neil Lutz, Michael Schapira, and Rebecca N. Wright. 2017. Dynamics at the Boundary of Game Theory and Distributed Computing. *ACM Trans. Econ. Comput.* 5, 3, Article 15 (August 2017), 20 pages.
https://doi.org/10.1145/3107182

A preliminary version of some of this work appeared in *Proceedings of the Second Symposium on Innovations in Computer Science (ICS 2011)* [25]. This work was partially supported by NSF grant CCF-1101690, ISF grant 420/12, the Israeli Center for Research Excellence in Algorithms (I-CORE), and the Office of Naval Research.
Authors' addresses: A. D. Jaggard, Formal Methods Section (Code 5543), U.S. Naval Research Laboratory, Washington, DC 20375, USA; email: aaron.jaggard@nrl.navy.mil (part of this work was carried out while Jaggard was at DIMACS, Rutgers University and visiting Colgate University); N. Lutz (current address), Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104, USA; email: njlutz@rutgers.edu (this work was carried out while Lutz was at Rutgers University and visiting the Hebrew University of Jerusalem); M. Schapira, School of Computer Science and Engineering, Hebrew University of Jerusalem, Edmond J. Safra Campus, Givat Ram, Jerusalem 91904, Israel; email: schapiram@huji.ac.il; R. N. Wright, Department of Computer Science and DIMACS, Rutgers University, Piscataway, NJ 08854, USA; email: rebecca.wright@rutgers.edu.

https://doi.org/10.1145/3107182

1 INTRODUCTION

Dynamic environments where decision makers repeatedly interact arise in a variety of settings, such as Internet protocols, large-scale markets, social networks, and multi-processor computer architectures. Study of these environments lies at the boundary of game theory and distributed computing. The decision makers are both strategic entities with individual economic preferences and computational entities with limited resources, working in a decentralized and uncertain environment. To understand the global behaviors that result from these interactions—the *dynamics* of these systems—we draw on ideas from both disciplines.

The notion of *self-stabilization* to a "legitimate" state in a distributed system parallels that of convergence to an equilibrium in a game. The foci, however, differ considerably. In game theory, there is extensive research on dynamics that result from what is perceived as natural strategic decision making (e.g., best- or better-response dynamics, fictitious play, or regret minimization). Even simple heuristics that require little information or computational resources can yield sophisticated behavior, such as the convergence of best-response dynamics to equilibrium points (see [20] and references therein). These positive results for simple game dynamics are, with few exceptions (see Section 2), based on the sometimes implicit and often unrealistic premise of a controlled environment in which actions are synchronous and coordinated. Distributed computing research emphasizes the environmental uncertainty that results from decentralization but has no notion of "natural" rules of behavior. It has long been known that environmental uncertainty—in the form of both asynchrony [15, 30] and arbitrary initialization [10]—introduces substantial difficulties for protocol termination in distributed systems. Our work bridges the gap between these two approaches by initiating the study of game dynamics in distributed computing settings. We take the first steps of this research agenda, focusing primarily on systems in which the decision makers, or *computational nodes*, are deterministic and have *bounded recall*, meaning that their behavior is based only on the "recent history" of system interaction. Our model is asynchronous in the sense of allowing, at every timestep, an adversarially chosen subset of nodes to be activated.

Our main contribution is a general impossibility result (Theorem 4.1) for asynchronous environments, showing that a large and natural class of bounded-recall dynamics can fail to converge whenever there are at least two "equilibrium points" of the dynamics. We prove this result using a *valency* argument (a now-standard technique in distributed computing theory [14, 30]). We discuss the implications of this result for game dynamics and describe its applications to asynchronous circuit design, social networks, interdomain routing protocols such as BGP, and congestion control in networks. We also explore the impact on convergence guarantees of asynchrony that is bounded, and we present complexity hardness results for checking whether an asynchronous system will always converge: We show that it is PSPACE-hard and requires exponential communication.

2 RELATED WORK

Our work relates to many ideas in game theory and in distributed computing. Here, we discuss game-theoretic work on the dynamics of simple strategies and on asynchrony, distributed computing work on fault tolerance and self-stabilization, and other connections between game theory and computer science (for more, see [18]). We also highlight the application areas we consider.

Algorithmic game theory. Since our work draws on both game theory and computer science, it may be considered part of the broader research program of algorithmic game theory (AGT), which merges concepts and ideas from those two fields [35]. Three main areas of study in AGT have been algorithmic mechanism design, which applies concepts from computer science to economic mechanism design [34]; the "price of anarchy," which describes the efficiency of equilibria and draws on approximability research [27]; and algorithmic and complexity research

on the computation of equilibria [36]. Analyzing the computational power of learning dynamics in games has been of particular interest (see, e.g., [2, 6, 26, 37]). Our work creates another link between game theory and computer science by drawing on two previously disjoint areas, self-stabilization in distributed computing theory and game dynamics, to explore broader classes of dynamics operating in adversarial distributed environments.

Adaptive heuristics. Much work in game theory and economics deals with *adaptive heuristics* (see [20] and references therein). Generally speaking, this long line of research explores the "convergence" of simple and myopic rules of behavior (e.g., best-response/fictitious-play/no-regret dynamics) to an "equilibrium." However, with few exceptions (see below), such analysis has so far primarily concentrated on synchronous environments in which steps take place simultaneously or in some other predetermined order. In this work, we explore dynamics of this type in asynchronous environments, which are more realistic for many applications.

Game-theoretic work on asynchronous environments. Some game-theoretic work on repeated games considers "asynchronous moves." Often, as in Marden and Shamma [31], this asynchrony merely indicates that players are not all activated at each time step and thus is used to describe environments where only one player is activated at a time ("alternating moves") or where there is a probability distribution that determines which player(s) are activated at each timestep. Other work does not explore the behavior of dynamics but has other research goals (e.g., characterizing equilibria, establishing folk theorems); see [29, 42], among others, and references therein. To the best of our knowledge, we are the first to study the effects of asynchrony (in the broad distributed computing sense) on the convergence of *game dynamics* to equilibria.

Fault-tolerant computation. We use ideas and techniques from work in distributed computing on protocol termination in asynchronous computational environments where nodes and communication channels are possibly faulty. Protocol termination in such environments, initially motivated by multi-processor computer architectures, has been extensively studied in the past three decades [3, 4, 9, 15, 23, 39], as nicely surveyed in [14, 30]. Fischer, Lynch, and Paterson [15] showed, in a landmark article, that a broad class of failure-resilient consensus protocols cannot provably terminate. Intuitively, the risk of protocol non-termination in that work stems from the possibility of failures; a computational node cannot tell whether another node is silent due to a failure or is simply taking a long time to react. Our non-convergence result, by contrast, applies to failure-free environments. In game-theoretic work that incorporated fault tolerance concepts, Abraham et al. [1] studied equilibria that are robust to defection and collusion.

Self-stabilization. The concept of self-stabilization is fundamental to distributed computing and dates back to Dijkstra 1974 [8] (see [10] and references therein). Convergence of dynamics to an "equilibrium" in our model can be viewed as the self-stabilization of such dynamics (where the "equilibrium points" are the legitimate configurations). Our formulation draws ideas from work in distributed computing (e.g., Burns' distributed daemon model [5]) and in networking research [17] on self-stabilization.

Applications. We discuss the implications of our non-convergence result across a wide variety of applications, that have previously been studied: convergence of game dynamics (see, e.g., [21, 22]); asynchronous circuits (see, e.g., [7]); diffusion of innovations, behaviors, and so on, in social networks (see [24, 33]); interdomain routing [17, 40]; and congestion control [16].

3 ASYNCHRONOUS DYNAMIC INTERACTION

In this section, we present our model of asynchronous dynamic interaction. Intuitively, an *interaction system* consists of a collection of computational nodes, each capable of selecting *actions* that

are visible to the other nodes. The *state* of the system at any time consists of each node's current action. Each node has a deterministic *reaction function* that maps system histories to actions. At every discrete timestep, each node activated by a schedule simultaneously applies its deterministic reaction function to select a new action, which is immediately visible to all other nodes.

Definition 3.1. An *interaction system* is characterized by a tuple $(n, A, \mathbf{f})$:

- The system has $n \in \mathbb{Z}_+$ *computational nodes*, labeled $1, \ldots, n$.
- $A = A_1 \times \cdots \times A_n$, where each A_i is a finite set called the *action space* of node i. A is called the *state space* of the system, and a *state* is an n-tuple $\mathbf{a} = (a_1, \ldots, a_n) \in A$. A *history* of the system is a nonempty finite sequence of states, $H \in A^\ell$, for some $\ell \in \mathbb{Z}_+$. The set of all histories is $A^+ = \bigcup_{\ell \in \mathbb{Z}_+} A^\ell$.
- $\mathbf{f} : A^+ \to A$ is a function given by $\mathbf{f}(H) = (f_1(H), \ldots, f_n(H))$, where $f_i : A^+ \to A_i$ is called node i's *reaction function*.

We now describe the asynchronous dynamics of our model, that is, the ways that a system's state can evolve due to interactions between nodes. Informally, there is some initial state, and, in each discrete time step $1, 2, 3, \ldots$, a subset of the nodes are *activated* according to a *schedule*. The nodes that are activated in a given timestep react simultaneously; each applies its reaction function to the current state to choose a new action. This updated action is *immediately observable* to all other nodes.[1]

Definition 3.2. Let $S \subseteq [n]$ be a set of nodes. Define the function $\mathbf{f}_S : A^+ \to A$ by $\mathbf{f}_S(H) = (\hat{f}_1(H), \ldots, \hat{f}_n(H))$, where each function $\hat{f}_i : A^+ \to A_i$ is given by

$$\hat{f}_i(\mathbf{a}^0, \ldots, \mathbf{a}^\ell) = \begin{cases} f_i(\mathbf{a}^0, \ldots, \mathbf{a}^\ell) & \text{if } i \in S \\ a_i^\ell & \text{otherwise.} \end{cases}$$

A *schedule* is a function $\sigma : \mathbb{Z}_+ \to 2^{[n]}$ that maps each t to a (possibly empty) subset of the computational nodes.[2] If $i \in \sigma(t)$, then we say that node i is *activated* at time t.

Since the reaction functions are deterministic, an initial history and a schedule completely determine the resulting infinite state sequence; we call these state sequences *trajectories*.

Definition 3.3. Let $H = (\mathbf{a}^0, \ldots, \mathbf{a}^\ell) \in A^+$ be a history, and let σ be a schedule. The (H, σ)-*trajectory* of the system is the infinite sequence $\mathbf{a}^0, \mathbf{a}^1, \mathbf{a}^2, \ldots$ extending H such that for every $t > \ell$,

$$\mathbf{a}^t = \mathbf{f}_{\sigma(t)}(\mathbf{a}^0, \ldots, \mathbf{a}^{t-1})$$

The history $(\mathbf{a}^0, \ldots, \mathbf{a}^{t-1})$ is the length-t *prefix* of the (H, σ)-trajectory.

3.1 Fairness and Convergence

Our main theorem is an impossibility result, in which we show that an adversarially chosen initial history and schedule can prevent desirable system behavior. Notice that an arbitrary schedule might never allow some or all nodes to react or might stop activating them after some time. Hence, we limit this adversarial power (thereby strengthening our impossibility result) by restricting our attention to *fair* schedules, which never permanently stop activating any node.

Definition 3.4. A *fair schedule* is a schedule σ that activates each node infinitely many times, that is, for each $i \in [n]$, the set $\{t \in \mathbb{Z}_+ : i \in \sigma(t)\}$ is infinite. A *fair trajectory* is one that is the (H, σ)-trajectory for some history H and some fair schedule σ.

[1] This model has "perfect monitoring." While this is clearly unrealistic in some important real-life contexts (e.g., some of the environments considered in Section 5), this restriction only strengthens our main results, which are impossibility results.
[2] $[n]$ denotes $\{1, \ldots, n\}$, and for any set S, 2^S is the set of all subsets of S.

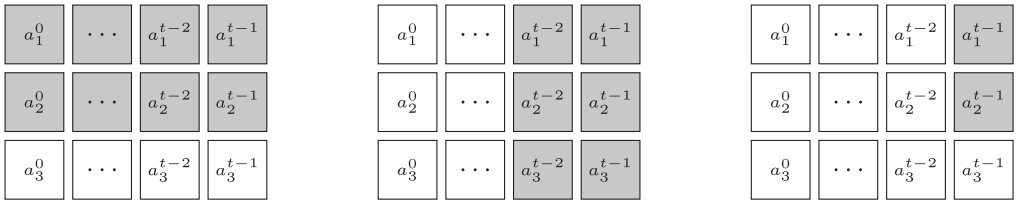

Fig. 1. Shading shows the information about past and current actions available to node 3 at time t given different reaction function restrictions. Left: self-independent. Node 3 can see the entire record of other nodes' past actions, but not its own. The length of this record gives the current timestamp t. Center: 2-recall. Node 3 can see only the two most recent states. Unless the reaction function is stationary, it may also use the value of the current timestamp. Right: self-independent and historyless. Node 3 can only see other nodes' most recent actions and cannot even see the value of the current timestamp.

We are especially interested in whether a system's fair trajectories *converge*, eventually remaining at a single state forever.

Definition 3.5. A trajectory $\mathbf{a}^0, \mathbf{a}^1, \mathbf{a}^2, \ldots$ *converges* to a state $\mathbf{b}$ if there exists some $T \in \mathbb{Z}_+$ such that, for all $t > T$, $\mathbf{a}^t = \mathbf{b}$. The system is *convergent* if every fair trajectory converges. A state $\mathbf{b}$ is a *limit state* of the system if some fair trajectory converges to $\mathbf{b}$.

Note that it is possible for a trajectory to visit a limit state without converging to that state, meaning that limit states are not necessarily "stable" or "absorbing." They may, however, have basins of attraction, in the sense that reaching certain histories might guarantee convergence to a given limit state.

Definition 3.6. A history H is *committed* to a limit state $\mathbf{b}$ if, for every fair schedule σ, the (H, σ)-trajectory converges to $\mathbf{b}$. An *uncommitted* history is one that is not committed to any state.

3.2 Informational Restrictions on Reaction Functions

This framework allows for very powerful reaction functions. We now present several possible restrictions on the information they may use. These are illustrated in Figure 1.

Our main theorem concerns systems in which the reaction functions are *self-independent*, meaning that each node ignores its own past and present actions when reacting to the system's state. In discussing self-independence, we use the notation

$$A_{-i} = A_1 \times \cdots \times A_{i-1} \times A_{i+1} \times \cdots \times A_n,$$

the state space of the system when i is ignored. Similarly, for a state $\mathbf{a}$, $\mathbf{a}_{-i} \in A_{-i}$ denotes $(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)$, and given a history $H = (\mathbf{a}^0, \ldots, \mathbf{a}^{\ell-1})$, we write H_{-i} for $(\mathbf{a}^0_{-i}, \ldots, \mathbf{a}^{\ell-1}_{-i})$. Using this notation, we formally define self-independence.

Definition 3.7. A reaction function f_i is *self-independent* if there exists a function $g_i : A^+_{-i} \to A_i$ such that $f_i(H) = g_i(H_{-i})$ for every history $H \in A^+$.

A reaction function has *bounded recall* if it only depends on recent states.

Definition 3.8. Given $k \in \mathbb{Z}_+$ and a history $H = (\mathbf{a}^0, \ldots, \mathbf{a}^{t-1}) \in A^t$ with $t \geq k$, the *k-history* at H is $H_{|k} := (\mathbf{a}^{t-k}, \ldots, \mathbf{a}^{t-1})$, the k-tuple of most recent states. A reaction function f_i has *k-recall* if it only depends on the k-history and the time counter, that is, there exists a function $g_i : A^k \times \mathbb{Z}_+ \to A_i$ such that $f_i(H) = g_i(H_{|k}, t)$ for every time $t \geq k$ and history $H \in A^t$.

We sometimes slightly abuse notation by referring to the restricted-domain function g_i, rather than f_i, as the node's reaction function.

A bounded-recall reaction function is *stationary* if it also ignores the time counter.

Definition 3.9. We say that a k-recall reaction function is *stationary* if the time counter t is of no importance. That is, if there exists a function $g_i : A^k \to A_i$ such that $f_i(H) = g_i(H_{|k})$ for every time $t \geq k$ and history $H \in A^t$. A reaction function f_i is *historyless* if f_i is both 1-recall and stationary. That is, if f_i only depends on the nodes' most recent actions.

While seemingly very restricted, historyless dynamics capture the prominent and extensively studied best-response dynamics from game theory (as we discuss in Section 5.1). We show in Section 5 that historyless dynamics also encompass a host of other applications of interest, ranging from Internet protocols to the adoption of technologies in social network.

4 GENERAL NON-CONVERGENCE RESULT

We now present our main theorem, a general impossibility result for convergence of nodes' actions under bounded-recall dynamics in asynchronous, distributed computational environments.

Theorem 4.1 (Main Theorem). *In an interaction system where every reaction function is self-independent and has bounded recall, the existence of multiple limit states implies that the system is not convergent.*

We prove this theorem in Section 4.1 by using a valency argument. In Section 4.2, we show that the hypotheses of Theorem 4.1 are necessary. We then discuss in Section 4.3 the connections of this work to the famous result of Fischer et al. [15] on the impossibility of resilient consensus.

Note that system convergence is closely related to *self-stabilization*, which is a guarantee that the system will reach and remaining within a set of *legitimate states*. For a set $L \subseteq A$, we say that a system *self-stabilizes to* L if, for every fair trajectory $\mathbf{a}^0, \mathbf{a}^1, \mathbf{a}^2, \ldots$, there is some $T \in \mathbb{Z}_+$ such that, for every $t > T$, $\mathbf{a}^t \in L$. In systems satisfying its hypotheses, Theorem 4.1 precludes self-stabilization to any set containing only committed states.

4.1 Proof of Main Theorem

In proving this theorem, we use the following sequence of lemmas. We first show in Lemma 4.2 that it is sufficient to consider systems with 1-recall reaction functions. Then, in Lemma 4.3, we argue that such a system can be convergent only if every fair trajectory has no committed prefix. To show the existence of a fair trajectory with no committed prefix, we show that in any such system, uncommitted histories exist (Lemma 4.5) and can be extended to a longer uncommitted histories in a way that activates any given node (Lemma 4.6). This means that committed prefixes can be avoided forever on a trajectory that activates every node infinitely many times, that is, a fair trajectory.

Lemma 4.2. *If there exists a convergent interaction system with bounded-recall, self-independent reaction functions and multiple limit states, then there is also a convergent interaction system with 1-recall, self-independent reaction functions, and multiple limit states.*

Proof. Assume that $\Gamma = (n, A, \mathbf{f})$ is a convergent system with self-independent, k-recall reaction functions and multiple limit states, for some $k \in \mathbb{Z}_+$. Consider a 1-recall system $\Gamma' = (n, A', \mathbf{f}')$, where $A' = A_1^k \times \cdots \times A_n^k$ and $\mathbf{f}' : A' \times \mathbb{Z}_+ \to A'$ is given by

$$f_i'\left(\left(a_1^1, \ldots, a_1^k\right), \ldots, \left(a_n^1, \ldots, a_n^k\right), t\right) = \left[f_i\left(\left(a_1^1, \ldots, a_n^1\right), \ldots, \left(a_1^k, \ldots, a_n^k\right), kt\right)\right]^k.$$

Informally, a state in Γ' is the transpose of a k-history for Γ. The reaction function f_i' applies f_i to this transpose and repeats the output k times. Notice that Γ' has self-independent reaction functions. Furthermore, if $(a_1, \ldots, a_n)$ is a limit state of Γ, then $((a_1, \ldots, a_1), \ldots, (a_n, \ldots, a_n))$ is a limit state of Γ', so Γ' also has multiple limit states.

Let $\sigma : \mathbb{Z}_+ \to 2^{[n]}$ be a fair schedule, and let $H \in (A^k)^\ell$ be a history of Γ' for some $\ell \in \mathbb{Z}_+$. Define the schedule $\sigma' : \mathbb{Z}_+ \to 2^{[n]}$ by

$$\sigma'(t) = \begin{cases} \sigma(t/k) & t/k \in \mathbb{Z}_+ \\ \emptyset & \text{otherwise.} \end{cases}$$

Notice that σ' is also fair. Let $H' \in A^{k\ell}$ be the history for Γ formed by concatenating the k-tuples in H. It is easy to see that the (H, σ)-trajectory of Γ' converges if and only if the (H', σ')-trajectory of Γ converges. Since we assumed that Γ is convergent, it follows that Γ' is also. □

Lemma 4.3. *Let Γ be a convergent system with self-independent, 1-recall reaction functions. Then every fair trajectory in Γ has a committed finite prefix.*

Proof. Assume there exist some history H and fair schedule σ for Γ such that the (H, σ)-trajectory converges to a state $\mathbf{a} = (a_1, \ldots, a_n)$ but has no committed finite prefix. We will construct a fair schedule σ' such that the (H, σ')-trajectory does not converge, giving a contradiction.

Let $\mathbf{u}^0, \mathbf{u}^1, \mathbf{u}^2, \ldots$ be the (H, σ)-trajectory. Then there is some $t_0 \in \mathbb{Z}_+$ such that $\mathbf{u}^t = \mathbf{a}$ for all $t \geq t_0$. The fairness of σ implies that there is some $t_1 > t_0$ such that every node is activated by σ between t_0 and t_1, that is, $\bigcup_{t_0 < t < t_1} \sigma(t) = [n]$. By assumption, $(\mathbf{u}^0, \ldots, \mathbf{u}^{t_1})$ is not committed to $\mathbf{a}$, which means there is some time $t_2 \geq t_1$ and node $i \in [n]$ such that $f_i(\mathbf{a}, t_2) \neq a_i$. The fairness of σ also implies that there is some $t_3 > t_2$ such that $i \in \sigma(t_3)$. Since $t_3 \geq t_0$, we must have $f_i(\mathbf{a}, t_3) = a_i$. By self-independence, then, $f_i(\mathbf{a}', t_3) = a_i$ for all $\mathbf{a}'$ such that $\mathbf{a}'_{-i} = \mathbf{a}_{-i}$.

We use these facts to iteratively build our fair schedule σ'. In the (H, σ')-trajectory $\mathbf{v}^0, \mathbf{v}^1, \mathbf{v}^2, \ldots$, the system will repeatedly enter and exit the state $\mathbf{a}$. First, let $\sigma'(t) = \sigma(t)$ for all $1 \leq t \leq t_0$, so $\mathbf{v}^{t_0} = \mathbf{a}$. Define $\sigma'(t)$ on values $t_0 < t \leq t_2$ as follows.

$$\sigma'(t) = \begin{cases} \sigma(t) & t_0 < t < t_2 \\ \{i\} & t_2 \leq t \leq t_3. \end{cases}$$

By our choices of t_0, t_2, and t_3, this partial schedule activates every node and induces a segment $(\mathbf{v}^{t_0+1}, \ldots, \mathbf{v}^{t_3})$ of the (H, σ')-trajectory such that $\mathbf{v}^t = \mathbf{a}$ whenever $t_0 < t < t_2$ or $t = t_3$, but $\mathbf{v}^{t_2} \neq \mathbf{a}$.

Now set $t_0 = t_3$, select new t_1, t_2, i, and t_3 relative to this t_0, and iterate this process to define $\sigma'(t)$ for all values $t \in \mathbb{Z}_+$. Notice that σ' is fair and that the (H, σ')-trajectory $\mathbf{v}^0, \mathbf{v}^1, \mathbf{v}^2, \ldots$ does not converge, which contradicts the assumption that the system is convergent. Therefore every fair trajectory in the system must have a committed finite prefix. □

We will use the following consequence of self-independence in the course of proving Lemmas 4.5 and 4.6.

Observation 4.4. *Let $H' = (\mathbf{a}^0, \ldots, \mathbf{a}^\ell)$ and $H' = (\mathbf{b}^0, \ldots, \mathbf{b}^\ell)$ be committed histories in an system with self-independent, 1-recall reaction functions. If $\mathbf{a}^\ell_{-i} = \mathbf{b}^\ell_{-i}$ for some $i \in [n]$, then H and H' are committed to the same limit state.*

Proof. Let $H = (\mathbf{a}^0, \ldots, \mathbf{a}^\ell)$ and $H' = (\mathbf{b}^0, \ldots, \mathbf{b}^\ell)$ be committed histories such that, for some $i \in [n]$, $\mathbf{a}^\ell_{-i} = \mathbf{b}^\ell_{-i}$, as in Figure 2. Let σ be any fair schedule such that $\sigma(\ell + 1) = \{i\}$, and consider the (H, σ)- and (H', σ)-trajectories. When node i is activated, it will choose the same action regardless of whether the history is H or H', by self-independence. As the reaction functions have 1-recall, this means that both these trajectories are identical after time $\ell + 1$. Thus, since H and H' are both committed, they must be committed to the same limit state. □

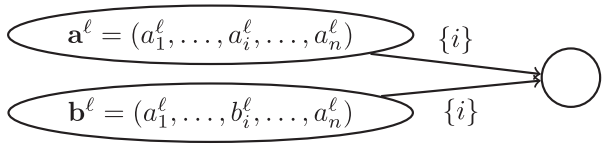

Fig. 2. Activating $\{i\}$ from $H = (\mathbf{a}^0, \ldots, \mathbf{a}^\ell)$ or $H' = (\mathbf{b}^0, \ldots, \mathbf{b}^\ell)$ will have the same outcome.

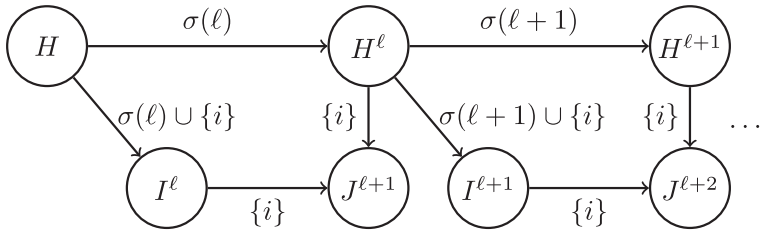

Fig. 3. All histories in the bottom row are committed to the same limit state $\mathbf{b}$.

Lemma 4.5. *Every interaction system with 1-recall, self-independent reaction functions and more than one limit state has at least one uncommitted history.*

Proof. Suppose that every history of length one is committed, and consider two such histories $(\mathbf{a}) = ((a_1, \ldots, a_n))$ and $(\mathbf{b}) = ((b_1, \ldots, b_n))$. Observation 4.4 implies that, for all $1 \leq i <$, the histories $((a_1, \ldots, a_{i-1}, b_i, \ldots, b_n))$ and $((a_1, \ldots, a_i, b_{i+1}, \ldots, b_n))$ are committed to the same limit state and therefore that $\mathbf{a}$ and $\mathbf{b}$ are committed to the same limit state. Thus, all histories of length one must be committed to the same limit state, and it follows that all histories must be committed to the same limit state. This contradicts the system having more than one limit state. □

Lemma 4.6. *Let $(n, A, \mathbf{f})$ be an interaction system with self-independent, 1-recall reaction functions and more than one limit state, let $H = (\mathbf{a}^0, \ldots, \mathbf{a}^{\ell-1}) \in A^\ell$ be an uncommitted history, for some $\ell \in \mathbb{Z}_+$, and let $i \in [n]$ be a node. Then there exist some $t \geq \ell$ and schedule σ such that $i \in \sigma(t)$ and the length-$(t+1)$ prefix of the (H, σ)-trajectory is uncommitted.*

Proof. Assume for contradiction that no such t and σ exist. Consider all histories that result from activating a set containing i at history H. By assumption, each of these histories is committed. Notice that for all $S \subseteq [n]$ and $j \in [n]$, the states $\mathbf{f}_S(H)$ and $\mathbf{f}_{S \cup \{j\}}(H)$ can only differ at coordinate j. Hence, we can iteratively apply Observation 4.4, much as in the proof of Lemma 4.5, to see that all these histories must be committed to the same limit state, which we call $\mathbf{b}$.

Let σ be any fair schedule, and let $\mathbf{a}^0, \mathbf{a}^1, \ldots$ be the (H, σ)-trajectory. For each $t \in \mathbb{Z}_+$, let $H^t = (\mathbf{a}^0, \ldots, \mathbf{a}^t)$, and notice that $H^{\ell-1} = H$. For each $t \geq \ell$, let $\mathbf{v}^t = \mathbf{f}_{\sigma(t) \cup \{i\}}(H^{t-1})$, and let $I^t = (H^{t-1}, \mathbf{v}^t)$. Since $i \in \sigma(t) \cup \{i\}$, our assumption implies that each history I^t is committed. Let $\mathbf{w}^t = \mathbf{f}_{\{i\}}(H^{t-1})$, and note that by self-independence, $\mathbf{f}_{\{i\}}(I^{t-1}) = \mathbf{w}^t$ also, as illustrated in Figure 3. Let $J^t = (I^{t-1}, \mathbf{w}^t)$.

We now show by induction on t that, for every $t \geq \ell$, the history I^t is committed to $\mathbf{b}$. This holds for $t = \ell$ by our definition of $\mathbf{b}$. Fix $t > \ell$, and suppose that I^{t-1} is committed to $\mathbf{b}$. Then J^t is also committed to $\mathbf{b}$. Consider all histories that result from activating a set containing i at history H^{t-1}. As before, our assumption implies that all these histories are committed, and iterative application of Observation 4.4 shows that they are all committed to the same limit state. In particular, I^t must be committed to the same limit state as J^t, namely $\mathbf{b}$.

Since σ is a fair schedule, there is some time t for which $i \in \sigma(t)$. For this t, we have $H^t = I^t$, so H^t is committed to $\mathbf{b}$. Thus for every fair schedule σ, the (H, σ)-trajectory converges to $\mathbf{b}$, contradicting the assumption that H is uncommitted. We conclude that our assumption was false and that the lemma holds. □

Proof of Theorem 4.1. It follows from Lemmas 4.5 and 4.6 that, in every system with 1-recall, self-independent reaction functions and multiple limit states, it is be possible to activate each node infinitely many times without ever reaching a committed history. This means that every such system has a fair trajectory with no committed prefix. By Lemma 4.3, this implies that no such system may be convergent. The theorem follows immediately by Lemma 4.2.[3] □

4.2 Tightness of Main Theorem

The following two examples demonstrate that the statement of Theorem 4.1 does not hold if either the self-independence restriction or the bounded-recall restriction is removed.

Example 4.7. The self-independence restriction cannot be removed.

Consider a system with one node, with action space $\{\alpha, \beta\}$. When activated, the node always re-selects its own current action. Observe that the system is convergent despite having two limit states.

Example 4.8. The bounded-recall restriction cannot be removed.

Consider a system with two nodes, 1 and 2, each with the action space $\{\alpha, \beta\}$. The self-independent reaction functions of the nodes are as follows: Node 2 always chooses node 1's action, node 1 will choose β if node 2's action changed from α to β in the past, and α otherwise. Observe that node 1's reaction function has unbounded recall: it depends on the entire history of interaction. We make the observations that the system is convergent and has two limit states. Observe that if node 1 chooses β at some point in time due to the fact that node 2's action changed from α to β, then it will continue to do so thereafter; if, on the other hand, 1 never does so, then from some point in time onwards, node 1's action is constantly α. In both cases, node 2 will have the same action as node 1 eventually, and thus convergence to one of the two limit states, (α, α) and (β, β), is guaranteed. Hence, two limit states exist and the system is convergent nonetheless. Notice also that node 1's reaction functions requires only two states, so the bounded-recall restriction cannot be replaced by a memory restriction.

4.3 Connection to Consensus Protocols

We now discuss the relationship of our main result (Theorem 4.1) to the seminal result of Fischer et al. on the impossibility of *fault-resilient consensus protocols*. The consensus problem is fundamental to distributed computing research. We give a brief description of it here, and we refer the reader to Fischer et al. [15] for a detailed explanation of the model.

Fischer et al. studied an environment in which a group of *processes*, each with an initial value in $\{0, 1\}$, communicate with each other via *messages*. The objective is for all *non-faulty* processes to eventually agree on some *consensus* value $x \in \{0, 1\}$, where x must match the initial value of some process. Fischer et al. established that no consensus protocol is resilient to even a single failure. Their proof of this breakthrough non-termination result introduced the idea of a *valency* argument.

[3] Although our primary focus is on discrete state spaces, we note that in continuous metric spaces, the standard notion of convergence only requires indefinite approach; the limit point might never be reached. Accordingly, given a metric d on an infinite state space A, one could modify Definition 3.5 to say that a trajectory $\mathbf{a}^0, \mathbf{a}^1, \mathbf{a}^2, \ldots$ converges to a state $\mathbf{b}$ if for every $\varepsilon > 0$ there exists some $T \in \mathbb{Z}_+$ such that, for all $t > T$, $d(\mathbf{a}^t, \mathbf{b}) < \varepsilon$. If we require every limit state to have a committed neighborhood, then our proof of Theorem 4.1 still holds in this setting.

They showed that there exists some initial configuration that is *bivalent*, meaning that the resulting consensus could be either 0 and 1 (the outcome depends on the asynchronous schedule of message transmission), and that this bivalence can be maintained. Our proof of Theorem 4.1 also uses a valency argument, where uncommitted histories play the role of bivalent configurations.

Intuitively, the risk of protocol non-termination in the environment studied by Fischer et al. stems from the possibility of failures; a computational node cannot tell whether another node is silent due to a failure or is simply taking a long time to react. Our non-convergence result concerns environments in which nodes/communication channels do not fail. Thus, each node is guaranteed that all other nodes will eventually react. Observe that in such an environment reaching a consensus is easy; one pre-specified node i (the "dictator") waits until it learns all other nodes' inputs (this is guaranteed to happen as failures are impossible) and then selects a value v_i and informs all other nodes; then, all other nodes select v_i. By contrast, the possibility of non-convergence shown in Theorem 4.1 stems from limitations on nodes' behaviors. Hence, there is no immediate translation from the result of Fischer et al. to ours (and vice versa).

5 APPLICATIONS: GAMES, CIRCUITS, SOCIAL NETWORKS, AND ROUTING

We present implications of our impossibility result, Theorem 4.1, for several well-studied environments: game dynamics, circuit design, social networks, and Internet protocols. For most of these applications, the reaction functions are historyless. Recalling Definition 3.9, this means that they react only to the current state of the system.

5.1 Asynchronous Game Dynamics

Traditionally, work in game theory on game dynamics (e.g., best-response dynamics) relies on the explicit or implicit premise that players' behavior is somehow synchronized (in some contexts play is sequential, while in others it is simultaneous). Here, we consider the realistic scenario that there is no computational center than can synchronize players' selection of strategies. We describe these dynamics in the setting of this work and exhibit an impossibility result for best-response, and more general, dynamics.

A *game* is characterized by a triple $(n, S, \mathbf{u})$. There are n *players*, $1, \ldots, n$. Each player i has a *strategy set* S_i. $S = S_1 \times \cdots \times S_n$ is the space of *strategy profiles* $\mathbf{s} = (s_1, \ldots, s_n)$. Each player i has a *utility function* $u_i : S \to \mathbb{R}$, where $\mathbf{u} = (u_1 \ldots u_n)$. Intuitively, player i "prefers" states for which u_i is higher. Informally, a player is *best responding* when it has no incentive to unilaterally change its strategy.

Definition 5.1. In a game $U = (n, S, \mathbf{u})$, player i is *best responding* at $\mathbf{s} \in S$ if $u_i(\mathbf{s}) \geq u_i(\mathbf{s}')$ for every $\mathbf{s}' \in S$ such that $s_{-i} = s'_{-i}$. We write $s_i \in BR_i^U(\mathbf{s})$. A strategy profile $\mathbf{s} \in S$ is a *pure Nash equilibrium* (*PNE*) if every player is best responding at $\mathbf{s}$.

There is a natural relationship between games and the interaction systems described in Section 3. A player with a strategy set corresponds directly to a node with an action space, and a strategy profile may be viewed as a state. These correspondences are so immediate that we often use these terms interchangeably.

Consider the case of *best-response dynamics* for a game in which best responses are unique (a *generic* game): Starting from some arbitrary strategy profile, each player chooses its unique best response to other players' strategies when activated. Convergence to pure Nash equilibria under best-response dynamics is the subject of extensive research in game theory and economics, and both positive [32, 38] and negative [21, 22] results are known. If we view each player i in a game $(n, S, \mathbf{u})$ as a node in an interaction system, then under best-response dynamics its utility function u_i induces a self-independent historyless reaction function $f_i : S_{-i} \to S_i$, as long as best responses

are unique. Formally,

$$f_i(\mathbf{a}_{-i}) = \arg\max_{\alpha \in S_i} u_i(a_1, \ldots, \alpha, \ldots, a_n).$$

Conversely, any system with historyless and self-independent reaction functions can be described as following best-response dynamics for a game with unique best responses. Given reaction functions $f_1, \ldots, f_n$, consider the game where each player i's utility function is given by

$$u_i(\mathbf{a}) = \begin{cases} 1 & \text{if } f_i(\mathbf{a}) = a_i \\ 0 & \text{otherwise.} \end{cases}$$

Best-response dynamics on this game replicate the dynamics induced by those reaction functions. Thus historyless and self-independent dynamics are exactly equivalent to best-response dynamics. Since pure Nash equilibria are fixed points of these dynamics, the historyless case of Theorem 4.1 may be restated in the following form.

Theorem 5.2. *If there are two or more pure Nash equilibria in a game with unique best responses, then asynchronous best-response dynamics can potentially oscillate indefinitely.*

In fact, best-response dynamics are just one way to derive reaction functions from utility functions, that is, to translate preferences into behaviors. In general, a *game dynamics protocol* is a mapping from games to systems that makes this translation. Given a game $(n, S, \mathbf{u})$ as input, the protocol selects reaction functions $\mathbf{f} = (f_1, \ldots, f_n)$ and returns an interaction system $(n, S, \mathbf{f})$. The above non-convergence result holds for a large class of these protocols. In particular, it holds for bounded-recall and self-independent game dynamics, whenever pure Nash equilibria are limit states. When cast into game-theoretic terminology, Theorem 4.1 says that if players' choices of strategies are not synchronized, then the existence of two (or more) pure Nash equilibria implies that this broad class of game dynamics are not guaranteed to reach a pure Nash equilibrium. This result should be contrasted with positive results for such dynamics in the traditional synchronous game-theoretic environments. In particular, this result applies to best-response dynamics with bounded recall and consistent tie-breaking rules (studied by Zapechelnyuk [43]).

Theorem 5.3. *If there are two or more pure Nash equilibria in a game with unique best responses, then all bounded-recall self-independent dynamics for which those equilibria are fixed points can fail to converge in asynchronous environments.*

5.2 Asynchronous Circuits

The implications of asynchrony for circuit design have been extensively studied in computer architecture research [7]. By regarding each logic gate as a node executing an inherently historyless and self-independent reaction function, we show that an impossibility result for stabilization of asynchronous circuits follows from Theorem 4.1.

In this setting there is a Boolean circuit, represented as a directed graph G, in which the vertices represent the circuit's inputs and logic gates, and the edges represent the circuit's connections. The activation of the logic gates is asynchronous. That is, the gates' outputs are initialized in some arbitrary way, and then the update of each gate's output, given its inputs, is uncoordinated and unsynchronized. A *stable Boolean assignment* in this framework is an assignment of Boolean values to the circuit inputs and the logic gates that is consistent with each gate's truth table. We say that a Boolean circuit is *inherently stable* if it is guaranteed to converge to a stable Boolean assignment regardless of the initial Boolean assignment.

To show how Theorem 4.1 applies to this setting, we model an asynchronous circuit with a fixed input as a historyless interaction system with self-independent reaction functions. Every node in the system has action space $\{0, 1\}$. There is a node for each input vertex, which has a

constant (and thus self-independent) reaction function. For each logic gate, the system includes a node whose reaction function implements the logic gate on the actions of the nodes corresponding to its inputs. If any gate takes its own output directly as an input, then we model this using an additional identity gate; this means that all reaction functions are self-independent. Since every stable Boolean assignment corresponds to a limit state of this system, this instability result follows from Theorem 4.1:

Theorem 5.4. *If two or more stable Boolean assignments exist for an asynchronous Boolean circuit with a given input, then that asynchronous circuit is not inherently stable on that input.*

5.3 Diffusion of Technologies in Social Networks

Understanding the ways in which innovations, ideas, technologies, and practices disseminate through social networks is fundamental to the social sciences. We consider the classic economic setting [33] (which has lately also been approached by computer scientists [24]) where each decision maker has two technologies $\{X, Y\}$ to choose from and wishes to have the same technology as the majority of its "friends" (neighboring nodes in the social network). We exhibit a general asynchronous instability result for this environment.

In this setting there is a social network of users, represented by a connected graph in which users are the vertices and edges correspond to friendship relationships. There are two competing technologies, X and Y. Each user will repeatedly reassess his or her choice of technology, at timesteps separated by arbitrary finite intervals. When this happens, the user will select X if at least half of his or her friends are using X and otherwise select Y. A "stable global state" is a fixed point of these choice functions, meaning that no user will ever again switch technologies. Observe that if every user has chosen X or every user has chosen Y, then the system is in a stable global state.

The dynamics of this diffusion can be described as asynchronous best-response dynamics for the game in which each player's utility is 1 if his or her choice of technology is consistent with the majority (with ties broken in favor of X) and 0 otherwise. This game has unique best responses, and the strategy profiles $(X, \ldots, X)$ and $(Y, \ldots, Y)$ are both pure Nash equilibria for this game. Thus Theorem 5.2 implies the following result.

Theorem 5.5. *In every social network with at least one edge, the diffusion of technologies can potentially oscillate indefinitely.*

5.4 Interdomain Routing

Interdomain routing is the task of establishing routes between the smaller networks, or *autonomous systems* (ASes), that make up the Internet. It is handled by the *Border Gateway Protocol* (BGP). We abstract a recent result of Sami et al. [40] concerning BGP non-convergence and show that this result extends to several BGP-based *multipath routing* protocols that have been proposed in the past few years.

In the standard model for analyzing BGP dynamics [17], there is a network of *source* ASes that wish to send traffic to a unique *destination* AS d. Each AS i has a *ranking function* $<_i$ that specifies i's strict preferences over all simple (loop-free) routes leading from i to d. Each AS also has an *export policy* that specifies which routes it is willing to make available to each neighboring AS. Under BGP, each AS constantly selects the "best" route that is available to it (see [17] for more details). *BGP safety*, that is, guaranteed convergence to a stable routing outcome, is a fundamental desideratum that has been the subject of extensive work in both the networking and the standards communities. We now cast interdomain routing into the terminology of Section 3 to obtain a non-termination results for BGP as a corollary of Theorem 4.1.

Each AS acts as a computational node. The action space of each node i is the set of all simple routes from i to the destination d, together with the empty route $\emptyset$. For every state $(R_1, \ldots, R_n)$ of the system, i's reaction function f_i considers the set of routes $S = \{(i,j)R_j : j \text{ is } i\text{'s neighbor and } R_j \text{ is exportable to } i\}$. If S is empty, then f_i returns $\emptyset$. Otherwise, f_i selects the route in S that is optimal with respect to $<_i$. Observe that this reaction function is deterministic, self-independent, and historyless and that a stable routing tree is a limit state of this system. Thus Theorem 4.1 implies the following result of Sami et al.

Theorem 5.6 (Sami et al. [40]). *If there are multiple stable routing trees in a network, then BGP is not safe on that network.*

Importantly, the asynchronous model of Section 3 is significantly *more restrictive* than that of Sami et al., so the result implied by Theorem 4.1 is stronger. Theorem 4.1 implies an even more general non-safety result for routing protocols that depend on ASes that act self-independently and with bounded recall. In particular, this includes recent proposals for BGP-based *multi-path routing* protocols that allow each AS to send traffic along multiple routes, for example, R-BGP [28] and Neighbor-Specific BGP (NS-BGP) [41].

5.5 Congestion Control

We now consider the fundamental task of congestion control in communication networks, which is achieved through a combination of mechanisms on *end-hosts* (e.g., TCP) and on *switches/routers* (e.g., RED and WFQ). We briefly describe the model of congestion control studied by Godfrey et al. [16].

There is a network of routers, represented by a directed graph G, in which vertices represent routers, and edges represent communication links. Each edge e has capacity c_e. There are n *source-target pairs* of vertices (s_i, t_i), termed "*connections*," that represent communicating pairs of end-hosts. Each source-target pair (s_i, t_i) is connected via some *fixed* route, R_i. Each source s_i transmits at a *constant* rate $\gamma_i > 0$.[4] For each of a router's outgoing edges, the router has a *queue management*, or *queuing*, policy, that dictates how the edge's capacity should be allocated between the connections whose routes traverse that edge. The network is asynchronous, so routers' queuing decisions can be made simultaneously. An *equilibrium* of the network is a global configuration of edges' capacity allocation such that the incoming and outgoing flows on each edge are consistent with the queuing policy for that edge. Godfrey et al. show that, while one might expect flow to be received at a constant rate whenever it is transmitted at a constant rate, this is not necessarily the case. Indeed, Godfrey et al. present examples in which connections' throughputs can potentially fluctuate ad infinitum, never converging an equilibrium.

We model such a network as a historyless interaction system to show that *every* network with multiple equilibria can oscillate indefinitely. The computational nodes of the system are the edges. The action space of each edge e intuitively consists of all possible ways to divide traffic going through e between the connections whose routes traverse e. More formally, for every edge e, let $N(e)$ be the number connections whose paths go through e. Then e's action space is $A_e = \{(x_1, \ldots, x_{N(e)}) : \text{ each } x_i \geq 0 \text{ and } \Sigma_i x_i \leq c_e\}$. We assume that the x_i have bounded precision, meaning that the state space is finite.

Edge e's reaction function f_e models the queuing policy according to which e's capacity is shared: For every $N(e)$-tuple of nonnegative incoming flows $(w_1, \ldots, w_{N(e)})$, f_e outputs an action $(x_1, \ldots, x_{N(e)}) \in A_e$ such that for every $i \in [N(e)]$, $x_i \leq w_i$—a connection's flow leaving the edge cannot exceed its flow entering the edge. These reaction functions are historyless and

[4]This is modeled via the addition of an edge $e = (u, s_i)$ to G, such that $c_e = \gamma_i$, and u has no incoming edges.

self-independent, and an equilibrium of the network is a limit state of this system. Using Theorem 4.1, then, we can obtain the following impossibility result.

THEOREM 5.7. *If there are multiple capacity-allocation equilibria in the network, then dynamics of congestion control can potentially oscillate indefinitely.*

6 COMPLEXITY OF ASYNCHRONOUS DYNAMICS

We now turn to the communication complexity and computational complexity of determining whether a system is convergent. We present hardness results in both models of computation even for the case of historyless interaction. Our computational complexity result shows that even if nodes' reaction functions can be succinctly represented, determining whether the system is convergent is PSPACE-complete. Alongside its computational implications, this intractability result implies that (unless PSPACE $\subseteq$ NP) we cannot hope to have short, efficiently verifiable certificates that guarantee a system's convergence.

6.1 Communication Complexity

The following result shows that, in general, determining whether a system is convergent cannot be done efficiently.

THEOREM 6.1. *Determining if a system with n nodes, each with 2 actions, is convergent requires communicating $\Omega(2^n)$ bits. This holds even if all nodes have historyless and self-independent reaction functions.*

PROOF. To prove our result, we present a reduction from the *2-party set disjointness problem*, a well-known problem in communication complexity theory: There are two parties, Alice and Bob. Each party holds a subset of $[q]$; Alice holds the subset A and Bob holds the subset B. The objective is to determine whether $A \cap B = \emptyset$. This problem instance is denoted $\textsc{Disj}^q(A, B)$. The following is well known.

THEOREM 6.2. *Determining whether $A, B \subseteq [q]$ are disjoint requires (in the worst case) the communication of $\Omega(q)$ bits. This lower bound applies to randomized protocols with bounded 2-sided error and also to nondeterministic protocols.*

We now present a reduction from 2-party set disjointness to the question of determining whether a system with historyless and self-independent reaction functions is convergent. Given an instance $\textsc{Disj}^q(A, B)$, we construct a system with n nodes, each with two actions, as follows. (The relation between the parameters q and n is to be specified later.) Let the action space of each node be $\{0, 1\}$. We now define the reaction functions of the nodes. Consider the possible action profiles of nodes $3, \ldots, n$, that is, the set $\{0, 1\}^{n-2}$. Observe that this set of actions profiles is the $(n-2)$-hypercube Q_{n-2}, and thus can be visualized as the graph whose vertices are indexed by the binary $(n-2)$-tuples and such that two vertices are adjacent if and only if they differ in exactly one coordinate. The reaction functions are based on following a *snake* in this hypercube.

Definition 6.3. A *snake* in a hypercube Q_n is a simple cycle $S = (v_0, \ldots, v_k)$ that is *chordless*, that is, for each v_i, v_j on S, if v_i and v_j are neighbors in Q_n, then $v_j \in \{v_{i-1}, v_{i+1}\}$.

Let S be a maximal snake in Q_{n-2}, and let $q = |S|$. We now show our reduction from $\textsc{Disj}^q$. We identify each element $j \in [q]$ with a unique vertex $v^j \in S$. Without loss of generality, we assume that 0^{n-2} is on S. For ease of exposition we also assume that 1^{n-2} is not on S. (Getting rid of this assumption is easy.) Orient the edges in S to form a cycle. For any edge that has exactly one endpoint in S, orient the edge toward S. An example is given in Figure 4. Orient all other edges arbitrarily. For each $i = 3, \ldots, n$, this orientation induces a function $g_i : Q_{n-2} \to \{0, 1\}$, where $g_i(a_3, \ldots, a_n)$

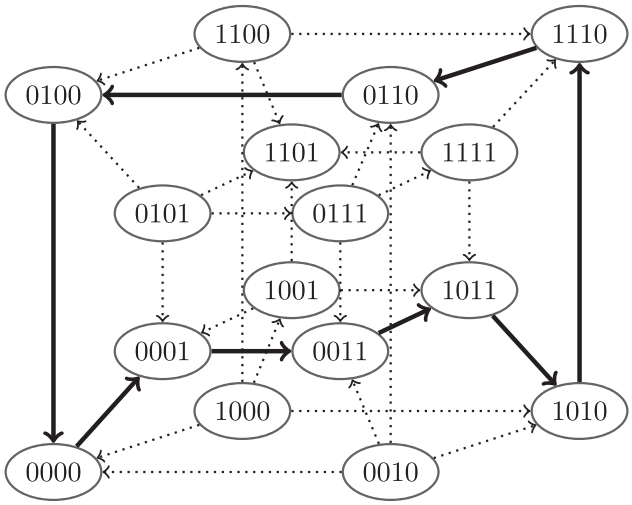

Fig. 4. An acceptable orientation of the edges on Q_4. The solid edges form a cycle on a maximal snake S, and no edge is directed away from S.

is determined by the direction of the edge $\{(a_3, \ldots a_{i-1}, 0, a_{i+1}, \ldots, n), (a_3, \ldots a_{i-1}, 1, a_{i+1}, \ldots, n)\}$. The nodes' self-independent and historyless reaction functions are as follows.

$$f_1(a_1, \ldots, a_n) = \begin{cases} 0 & \text{if } a_2 = 1 \text{ and } (a_3, \ldots, a_n) = v^j \text{ for some } j \in A \\ 1 & \text{otherwise} \end{cases},$$

$$f_2(a_1, \ldots, a_n) = \begin{cases} 0 & \text{if } a_1 = 1 \text{ and } (a_3, \ldots, a_n) = v^j \text{ for some } j \in B \\ 1 & \text{otherwise} \end{cases},$$

$$f_i(a_1, \ldots, a_n) = \begin{cases} g_i(a_3, \ldots, a_n) & \text{if } a_1 = a_2 = 0 \\ 1 & \text{otherwise} \end{cases}.$$

Informally, the aim is for nodes $3, \ldots, n$ to follow the snake S to vertices corresponding to each value $j \in [q]$.

Observation 6.4. *1^n is the unique limit state of the system.*

In our reduction Alice simulates node 1 (whose reaction function is based on A), Bob simulates node 2 (whose reaction function is based on B), and one of the two parties simulates all other nodes (whose reaction functions are based on neither A nor B). The theorem now follows from the combination of the following two claims.

Claim 6.5. *In an oscillation there must be infinitely many time steps in which both node 1 and 2's actions are* 0.

Proof. Suppose that from some moment forth it is never the case that both node 1 and 2's actions are 0. Observe that from that time onwards the nodes $3, \ldots, n$ will always choose the action 1 when activated. Hence, after some time has passed the actions of all nodes in $\{3, \ldots, n\}$ will be 1. Observe that whenever nodes 1 and 2 are activated thereafter they shall choose the action 1, so we have convergence to the limit state 1^n. □

Claim 6.6. *The system is convergent iff $A \cap B = \emptyset$.*

Proof. If $A \cap B \neq \emptyset$, then initialize the system to a state $(0, 0, a_3, \ldots, a_n)$, where $(a_3, \ldots, a_n) = v^1$. Consider a schedule that activates $\{3, \ldots, n\}$ in every timestep until $(a_3, \ldots, a_n) = v^j$ for some $j \in A \cap B$. When that happens, the schedule activates $\{1, 2\}$ for two consecutive timesteps, then resumes activating $\{3, \ldots, n\}$. The functions g_i ensure that, for each $j \in [q]$, the vector $(a_3, \ldots, a_n)$ will be equal to v_j within a finite number of timesteps. Since there is some $j \in A \cap B$, nodes 1 and 2 will eventually be activated, so this schedule is fair. This initial state and schedule clearly produce an oscillation, so the system is not convergent in this case.

Now assume for contradiction that $A \cap B = \emptyset$ and the system is not convergent. We know from Claim 6.5 that if there is an oscillation, then there are infinitely many time steps in which both node 1 and 2's actions are 0. We argue that this implies that there must be infinitely many time steps in which both nodes select action 0 *simultaneously*. Indeed, node 1 only chooses action 0 if node 2's action is 1, and vice versa, and so if both nodes never choose 0 simultaneously, then it is never the case that both nodes' actions are 0 at the same time step, a contradiction. Now, when is it possible for both 1 and 2 to choose 0 at the same time? Observe that this can only be if the actions of nodes $3, \ldots, n$ constitute an element that is in both A and B. Hence, $A \cap B \neq \emptyset$, another contradiction. □

We have reduced $\textsc{Disj}^q(A, B)$, which requires $\Omega(q)$ bits of communication, to the problem of checking convergence of an n-node system with historyless and self-independent reaction functions. As q was defined as the length of a maximal snake in Q_n, a classical combinatorial result due to Evdokimov shows that $q = \Omega(2^n)$.

Theorem 6.7 (Evdokimov [12]). *Let $z \in \mathbb{Z}_+$ be sufficiently large. Then, the size $|S|$ of a maximal snake in the z-hypercube Q_z is at least $\lambda 2^z$ for some $\lambda > 0$.*

This completes the proof of Theorem 6.1. □

6.2 Computational Complexity

The above communication complexity hardness result required the representation of the reaction functions to (potentially) be exponentially long. What if the reaction functions can be succinctly described? We now present a strong computational complexity hardness result for the case that each reaction function f_i is historyless and is given explicitly in the form of a Boolean circuit.

Theorem 6.8. *When the reaction functions are given as Boolean circuits, determining whether a historyless system with n nodes is convergent is PSPACE-complete.*

Proof. Our proof is based on the proof of Fabrikant and Papadimitriou [13] that checking BGP safety is PSPACE-complete. Importantly, that result does not imply Theorem 6.8, since the model of asynchronous interaction considered in that work does not allow for simultaneous activation of nodes. We prove our theorem by reduction from the problem of determining whether a linear space-bounded Turing machine (TM) will halt from every starting configuration.

For $n \in Z_+$, let M be a TM that can only access the first n tape cells. Let Q be M's machine state space, Γ its tape alphabet, and $\delta : Q \times \Gamma \to Q \times \Gamma \times \{-1, 0, 1\}$ its transition function. A configuration of M is a triple $(q, \mathbf{a}, j)$, where $q \in Q$ is a machine state, $\mathbf{a} \in \Gamma^n$ describes the tape contents, and $j \in [n]$ gives the location of the control head.

Then we say that $\langle M, 1^n \rangle$ is in SHC (for space-bounded halting from all configurations) if, for every configuration, M will halt if it begins its computation from that configuration. As Fabrikant and Papadimitriou [13] argue, the problem of checking whether a space-bounded TM will accept a blank input is reducible to SHC, and thus SHC is PSPACE-hard.

We now reduce SHC to the problem of determining whether a historyless interaction system is convergent. Given $n \in \mathbb{Z}_+$ and a TM M that can only access the first n tape cells, we construct a

historyless system that is convergent if and only if $\langle M, 1^n \rangle$ is in SHC. The system has a *cell node* for each of the first n tape cells and a *head node* to represent the control head. The cell nodes $1, \ldots, n$ each have the action space Γ, and the head node $n+1$ has action space $Q \times \Gamma \times [n] \times \{-1, 0, 1\}$. For $\mathbf{a} \in \Gamma^n$, each cell node $i \in [n]$ has the reaction function

$$f_i(\mathbf{a}, (q, \gamma, j, d)) = \begin{cases} \gamma & \text{if } i = j \\ a_i & \text{otherwise.} \end{cases}$$

For the head node, $f_{n+1}(\mathbf{a}, (q, \gamma, j, d))$ is given by the following procedure:

if $a_j = \gamma$ **then**
  if $j + d \in [n]$ **then**
    $j \leftarrow j + d$;
  $(q, \gamma, d) \leftarrow \delta(q, a_j)$;
return (q, γ, j, d);

Observe that $(\mathbf{a}, (q, \gamma, j, d))$ is a limit state of this system if and only if q is a halting machine state for M. Suppose the system is convergent, and let $(q, \mathbf{a}, j)$ be a configuration of M. Consider the system trajectory that begins at state $(\mathbf{a}, (q, a_j, j, 0))$ and follows the schedule that activates every node in every round. This trajectory will reach a limit state, and it corresponds directly to a halting run of M that begins from $(q, \mathbf{a}, j)$, so $\langle M, 1^n \rangle$ is in SHC.

Conversely, suppose that $\langle M, 1^n \rangle$ is in SHC, and let $(\mathbf{a}, (q, \gamma, j, d))$ be a system state. Suppose that $a_j = \gamma$. Then consider the computation by M that begins at configuration $(q, \mathbf{a}, j')$, where $j' = \min\{\max\{j + d, 1\}, n\}$. This computation will reach a halting machine state, and any fair trajectory from $(\mathbf{a}, (q, \gamma, j, d))$ will go through the same sequence of machine states as this computation, which means that the trajectory will converge to a limit state. If $a_j \neq \gamma$, then the system state will remain the same until j is activated and takes action γ, after which the above argument applies. Thus the system is convergent. We conclude that checking whether the system is convergent is PSPACE-hard. Since it is straightforward to check convergence in polynomial space, this problem is PSPACE-complete. □

In a preliminary version of this work [25], we conjectured that the above PSPACE-completeness result also holds for the case of self-independent reaction functions.

CONJECTURE 6.9. *Determining whether a system with n nodes, each with a deterministic self-independent and historyless reaction function, is convergent is PSPACE-complete.*

This conjecture has since been proved by Engelberg et al. [11].

7 CONCLUSIONS AND FUTURE RESEARCH

In this article, we have taken the first steps towards a complete understanding of strategic dynamics in distributed settings. We proved a general non-convergence result and several hardness results within this model. We also discussed some important aspects such as the implications of fairness and randomness, as well as applications to a variety of settings. We believe that we have only scratched the surface in the exploration of the convergence properties of game dynamics in distributed computational environments, and many important questions remain wide open. We now outline several interesting directions for future research.

Other limitations, convergence notions, and equilibria. We have considered particular limitations on reaction functions, modes of convergence, and kinds of equilibria. Understanding the effects of asynchrony on different classes of reaction functions (e.g., uncoupled dynamics, dynamics

with outdated information) and for other types of convergence (e.g., of the empirical distributions of play) and equilibria (e.g., mixed Nash equilibria, correlated equilibria) is a broad and challenging direction for future research.

Other notions of asynchrony. We believe that better understanding the role of degrees of fairness, randomness, and other restrictions on schedules from distributed computing literature, in achieving convergence to equilibrium points is an interesting and important research direction.

Characterizing asynchronous convergence. We still lack characterizations of asynchronous convergence even for simple dynamics (e.g., deterministic and historyless). Our PSPACE -completeness result in Section 6 eliminates the possibility of short witnesses of guaranteed asynchronous convergence unless PSPACE $\subseteq$ NP, but elegant characterizations are still possible.

Topological and knowledge-based approaches. Topological [4, 23, 39] and knowledge-based [19] approaches have been very successful in addressing fundamental questions in distributed computing. Can these approaches shed new light on the implications of asynchrony for strategic dynamics?

Further exploring the environments of Section 5. We have applied our main non-convergence result to the environments described in Section 5. These environments are of independent interest and are indeed the subject of extensive research. Hence, the further exploration of dynamics in these settings is important.

ACKNOWLEDGMENTS

We thank Danny Dolev, Alex Fabrikant, Idit Keidar, Jonathan Laserson, Nati Linial, Yishay Mansour, Yoram Moses, and two anonymous reviewers for helpful discussions and comments. This work was initiated partly as result of the DIMACS Special Focus on Communication Security and Information Privacy.

REFERENCES

[1] Ittai Abraham, Danny Dolev, Rica Gonen, and Joe Halpern. 2006. Distributed computing meets game theory: Robust mechanisms for rational secret sharing and multiparty computation. In *Proceedings of the 25th Annual ACM Symposium on Principles of Distributed Computing (PODC '06)*. ACM, New York, NY, 53–62. DOI: https://dx.doi.org/10.1145/1146381.1146393

[2] Maria-Florina Balcan, Florin Constantin, and Ruta Mehta. 2012. The weighted majority algorithm does not converge in nearly zero-sum games. In *Proceedings of the ICML Workshop on Markets, Mechanisms and Multi-Agent Models*.

[3] Michael Ben-Or. 1986. Randomized agreement protocols. In *Fault-Tolerant Distributed Computing*, Barbara Simonsand Alfred Spector (Eds.). Springer, New York, NY, 72–83. DOI: https://dx.doi.org/10.1007/bfb0042326

[4] Elizabeth Borowsky and Eli Gafni. 1993. Generalized FLP impossibility result for t-resilient asynchronous computations. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing (STOC'93)*. ACM, New York, NY, 91–100. DOI: https://dx.doi.org/10.1145/167088.167119

[5] James E. Burns. 1987. *Self-stabilizing Rings Without Demons*. Technical Report GIT-ICS-87/36. School of Information and Computer Science, Georgia Institute of Technology.

[6] Constantinos Daskalakis, Rafael Frongillo, Christos H. Papadimitriou, George Pierrakos, and Gregory Valiant. 2010. On learning algorithms for Nash equilibria. In *Proceedings of the Third International Symposium on Algorithmic Game Theory (SAGT'10)*, Spyros Kontogiannis, Elias Koutsoupias, and Paul G. Spirakis (Eds.). Springer, Berlin, 114–125. DOI: https://dx.doi.org/10.1007/978-3-642-16170-4_11

[7] Al Davis and Steven M. Nowick. 1997. *An Introduction to Asynchronous Circuit Design*. Technical Report. The Encyclopedia of Computer Science and Technology.

[8] Edsger W. Dijkstra. 1974. Self-stabilizing systems in spite of distributed control. *Commun. ACM* 17, 11 (1974), 643–644. DOI: https://dx.doi.org/10.1145/361179.361202

[9] Danny Dolev, Cynthia Dwork, and Larry Stockmeyer. 1987. On the minimal synchronism needed for distributed consensus. *J. ACM* 34, 1 (1987), 77–97. DOI: https://dx.doi.org/10.1145/7531.7533

[10] Shlomi Dolev. 2000. *Self-Stabilization*. MIT Press, Cambridge, MA.

[11] Roee Engelberg, Alex Fabrikant, Michael Schapira, and David Wajc. 2013. Best-response dynamics out of sync: Complexity and characterization. In *Proceedings of the 13th ACM Conference on Electronic Commerce (EC'13)*. ACM, New York, NY, 379–396. DOI: https://dx.doi.org/10.1145/2492002.2482609

[12] A. A. Evdokimov. 1969. Maximal length of circuit in a unitary n-dimensional cube. *Math. Notes Acad. Sci. USSR* 6, 3 (1969), 642–648. DOI: https://dx.doi.org/10.1007/BF01119684

[13] Alex Fabrikant and Christos H. Papadimitriou. 2008. The complexity of game dynamics: BGP oscillations, sink equilibria, and beyond. In *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'08)*. SIAM, Philadelphia, PA, 844–853.

[14] Faith Fich and Eric Ruppert. 2003. Hundreds of impossibility results for distributed computing. *Distrib. Comput.* 16, 2–3 (2003), 121–163. DOI: https://dx.doi.org/10.1007/s00446-003-0091-y

[15] Michael J. Fischer, Nancy A. Lynch, and Michael S. Paterson. 1985. Impossibility of distributed consensus with one faulty process. *J. ACM* 32, 2 (1985), 374–382. DOI: https://dx.doi.org/10.1145/3149.214121

[16] P. Brighten Godfrey, Michael Schapira, Aviv Zohar, and Scott Shenker. 2010. Incentive compatibility and dynamics of congestion control. In *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'10)*. ACM, New York, NY, 95–106. DOI: https://dx.doi.org/10.1145/1811039.1811051

[17] Timothy G. Griffin, F. Bruce Shepherd, and Gordon Wilfong. 2002. The stable paths problem and interdomain routing. *IEEE/ACM Trans. Netw.* 10, 2 (2002), 232–243. DOI: https://dx.doi.org/10.1109/90.993304

[18] Joseph Y. Halpern. 2003. A computer scientist looks at game theory. *Games Econ. Behav.* 45, 1 (2003), 114–131. DOI: https://dx.doi.org/10.1016/S0899-8256(02)00529-8

[19] Joseph Y. Halpern and Yoram Moses. 1990. Knowledge and common knowledge in a distributed environment. *J. ACM* 37, 3 (July1990), 549–587. DOI: https://dx.doi.org/10.1145/79147.79161

[20] Sergiu Hart. 2005. Adaptive heuristics. *Econometrica* 73, 5 (2005), 1401–1430. https://www.jstor.org/stable/3598879

[21] Sergiu Hart and Andreu Mas-Colell. 2003. Uncoupled dynamics do not lead to Nash equilibrium. *Am. Econ. Rev.* 93, 5 (2003), 1830–1836. DOI: https://dx.doi.org/10.1257/000282803322655581

[22] Sergiu Hart and Andreu Mas-Colell. 2006. Stochastic uncoupled dynamics and Nash equilibrium. *Games Econ. Behav.* 57, 2 (2006), 286–303. DOI: https://dx.doi.org/10.1016/j.geb.2005.09.007

[23] Maurice Herlihy and Nir Shavit. 1999. The topological structure of asynchronous computability. *J. ACM* 46, 6 (1999), 858–923. DOI: https://dx.doi.org/10.1145/331524.331529

[24] Nicole Immorlica, Jon Kleinberg, Mohammad Mahdian, and Tom Wexler. 2007. The role of compatibility in the diffusion of technologies through social networks. In *Proceedings of the 8th ACM Conference on Electronic Commerce (EC'07)*. ACM, New York, NY, 75–83. DOI: https://dx.doi.org/10.1145/1250910.1250923

[25] Aaron D. Jaggard, Michael Schapira, and Rebecca N. Wright. 2011. Distributed computing with adaptive heuristics. In *Proceedings of the 2nd Symposium on Innovations in Computer Science (ICS'11)*. Tsinghua University Press, Beijing, 417–443.

[26] Robert D. Kleinberg, Katrina Ligett, Georgios Piliouras, and Éva Tardos. 2011. Beyond the Nash equilibrium barrier. In *Proceedings of the 2nd Symposium on Innovations in Computer Science (ICS'11)*. Tsinghua University Press, Beijing, 125–140.

[27] Elias Koutsoupias and Christos Papadimitriou. 2009. Worst-case equilibria. *Comput. Sci. Rev.* 3, 2 (2009), 65–69. DOI: https://dx.doi.org/10.1016/j.cosrev.2009.04.003

[28] Nate Kushman, Srikanth Kandula, Dina Katabi, and Bruce M. Maggs. 2007. R-BGP: Staying connected in a connected world. In *Proceedings of the 4th USENIX Conference on Networked Systems Design & Implementation (NSDI'07)*. USENIX, Berkeley, CA.

[29] Roger Lagunoff and Akihiko Matsui. 1997. Asynchronous choice in repeated coordination games. *Econometrica* 65, 6 (1997), 1467–1477. DOI: https://dx.doi.org/10.2307/2171745

[30] Nancy Lynch. 1989. A hundred impossibility proofs for distributed computing. In *Proceedings of the 8th Annual ACM Symposium on Principles of Distributed Computing (PODC'89)*. ACM, New York, NY, 1–28. DOI: https://dx.doi.org/10.1145/72981.72982

[31] Jason R. Marden and Jeff S. Shamma. 2012. Revisiting log-linear learning: Asynchrony, completeness and payoff-based implementation. *Games Econ. Behav.* 75, 2 (2012), 788–808. DOI: https://dx.doi.org/10.1016/j.geb.2012.03.006

[32] Dov Monderer and Lloyd S. Shapley. 1996. Potential games. *Games Econ. Behav.* 14 (1996), 124–143. DOI: https://dx.doi.org/10.1006/game.1996.0044

[33] Stephen Morris. 2000. Contagion. *Rev. Econ. Stud.* 67 (2000), 57–78. DOI: https://dx.doi.org/10.1111/1467-937X.00121

[34] Noam Nisan and Amir Ronen. 2001. Algorithmic mechanism design. *Games Econ. Behav.* 35, 1–2 (2001), 166–196. DOI: https://dx.doi.org/10.1006/game.1999.0790

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

[36] Christos H. Papadimitriou. 1994. On the complexity of the parity argument and other inefficient proofs of existence. *J. Comput. Syst. Sci.* 48, 3 (1994), 498–532. DOI: https://dx.doi.org/10.1016/S0022-0000(05)80063-7

[37] Christos H. Papadimitriou and Georgios Piliouras. 2016. From Nash equilibria to chain recurrent sets: Solution concepts and topology. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*. ACM, New York, NY, 227–235. DOI: https://dx.doi.org/10.1145/2840728.2840757

[38] Robert W. Rosenthal. 1973. A class of games possessing pure-strategy Nash equilibria. *Int. J. Game Theory* 2 (1973), 65–67. Issue 1. DOI: https://dx.doi.org/10.1007/BF01737559

[39] Michael Saks and Fotios Zaharoglou. 2000. Wait-Free k-Set Agreement is Impossible: The Topology of Public Knowledge. *SIAM J. Comput.* 29, 5 (2000), 1449–1483. DOI: https://dx.doi.org/10.1137/S0097539796307698

[40] Rahul Sami, Michael Schapira, and Aviv Zohar. 2009. Searching for stability in interdomain routing. In *Proceedings of the 28th IEEE International Conference on Computer Communications (INFOCOM'09)*. IEEE Press, Piscataway, NJ, 549–557. DOI: https://dx.doi.org/10.1109/INFCOM.2009.5061961

[41] Yi Wang, Michael Schapira, and Jennifer Rexford. 2009. Neighbor-specific BGP: More flexible routing policies while improving global stability. In *Proceedings of the 11th International Joint Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'09)*. ACM, New York, NY, 217–228. DOI: https://dx.doi.org/10.1145/1555349.1555375

[42] Kiho Yoon. 2004. The effective minimax value of asynchronously repeated games. *Int. J. Game Theory* 32, 4 (2004), 431–442. DOI: https://dx.doi.org/10.1007/s001820300161

[43] Andriy Zapechelnyuk. 2008. Better-reply dynamics with bounded recall. *Math. Oper. Res.* 33, 4 (2008), 869–879. DOI: https://dx.doi.org/10.1287/moor.1080.0323

Received July 2016; revised March 2017; accepted March 2017