Proceedings Article

Best-Response Dynamics Out of Sync: Complexity and Characterization

ROEE ENGELBERG, Google & CS Dept., Technion, roee@cs.technion.ac.il
ALEX FABRIKANT, Google Research, alexf@cal.berkeley.edu
MICHAEL SCHAPIRA, Hebrew University, schapiram@huji.ac.il
DAVID WAJC, CS Dept., Technion and Yahoo! Labs, sdavidwa@cs.technion.ac.il

In many computational and economic models of multi-agent interaction, each participant repeatedly "best-responds" to the others' actions. Game theory research on the prominent "best-response dynamics" model typically relies on the premise that the interaction between agents is somehow synchronized. However, in many real-life settings, e.g., internet protocols and large-scale markets, the interaction between participants is *asynchronous*. We tackle the following important questions: (1) When are best-response dynamics guaranteed to converge to an equilibrium even under asynchrony? (2) What is the (computational and communication) complexity of verifying guaranteed convergence? We show that, in general, verifying guaranteed convergence is intractable. In fact, our main negative result establishes that this task is undecidable. We exhibit, in contrast, positive results for several environments of interest, including complete, computationally-tractable, characterizations of convergent systems. We discuss the algorithmic implications of our results, which extend beyond best-response dynamics to applications such as asynchronous Boolean circuits.

Categories and Subject Descriptors: F.2.2 [**Nonnumerical Algorithms and Problems**]: Algorithmic Game Theory

Additional Key Words and Phrases: Best response dynamics; asynchronous models; convergence; game theory; complexity

1. INTRODUCTION

In dynamic environments where computational nodes (be they humans or machines) repeatedly interact, the prescribed behavior for each node is often to repeatedly "best-respond" to the others' actions. Game theory has produced a slew of alternative models of behavior, yet repeated best-response remains a natural, simple, and low-cost behavior to build into distributed systems, as evidenced, for instance, by today's protocols for routing and congestion control on the Internet (see [Fabrikant and Papadimitriou 2008; Jaggard et al. 2011; Griffin et al. 2002; Godfrey et al. 2010; Levin et al. 2008; Suchara et al. 2011]). Game theoretic analysis shows that, in many interesting contexts, the resulting best-response dynamics eventually drive the system to an equilibrium state (e.g., in potential games [Monderer and Shapley 1996; Osborne and Rubinstein 1994; Rosenthal 1973]). However, these positive results typically rely on the premise that nodes' interaction is somehow *synchronized*, i.e., nodes "take turns" at best-responding, and each node's actions are immediately observable to all other nodes.

When analyzing distributed computational systems, or large-scale economic environments such as markets, the premise that agents' interaction is somehow synchronized is usually unrealistic. In any Internet-scale distributed system, a computational node can perform many megaflops of complex computations in the few milliseconds it

EC'13, June 16–20, 2013, Philadelphia, USA.

takes light to cross a continent enabling any information at all to reach a distant node in the system. Consider, e.g., the Border Gateway Protocol (BGP), which establishes routes between the numerous self-interested economic entities that comprise today's Internet (AT&T, Google, Hebrew U, etc.). As observed in [Fabrikant and Papadimitriou 2008; Levin et al. 2008], BGP routers implement best-response dynamics in a specific class of routing games. BGP convergence to a "stable" global routing configuration is an important desideratum from a practical viewpoint, as oscillatory behavior can lead to significant outages and greatly degrade network performance. BGP operates in an inherently asynchronous environment and so analyzing BGP's convergence properties in such a setting is important.

Asynchronous best-response dynamics also arise in many other distributed Internet protocols: congestion control, peer-to-peer protocols, and more. Even in a single chip, especially in low-power or high-density settings, the design can benefit from the removal of a global clock, with circuit components interacting as asynchronous distributed nodes. This too can be modeled as best-response dynamics in the presence of asynchrony [Jaggard et al. 2011]. Such phenomena also arise outside the realm of computational artifacts. Asynchronous best-response can model viral spread of technologies in social networks. Also, in high-frequency equity trading, a transaction hedging across multiple international markets may well require a trader to transmit price data to another system while the market already starts to respond to the transaction. In all of these settings, listed in [Jaggard et al. 2011], stable system states are not only easier to analyze, but also more desirable from a variety of engineering considerations.

We tackle two closely-related questions: (1) Can we *characterize* when best-response dynamics are *guaranteed* to converge to an equilibrium even in the presence of asynchrony?; and (2) What is the *complexity* of checking such a system for guaranteed convergence?

1.1. Asynchronous Best-Response Dynamics

Under "traditional" best-response dynamics, agents take turns selecting strategies, each repeatedly selecting a strategy that maximizes his utility given the others' current strategies. This goes on until a "stable state", i.e., a pure Nash equilibrium, is reached. We present and study a model of asynchronous best-response dynamics that abstracts and unifies models of BGP routing and of TCP congestion control on the Internet [Griffin et al. 2002; Godfrey et al. 2010]).

As in traditional best-response dynamics, in our model the system evolves from some initial combination of agents' strategies and agents repeatedly best-respond to other agents' actions. However, our model of "asynchronous best-response dynamics" captures scenarios in which agents can make decisions simultaneously, and in which the propagation of information about agents' actions is not instantaneous. In our model, whenever an agent selects a new strategy, he announces his newly chosen strategy via an update message to every other agent, sent along a private pairwise communication channel. Update messages sent along each of these channels need not arrive immediately and can experience (arbitrary) delays. We call a multi-agent system "convergent" if convergence of best-response dynamics to an equilibrium is guaranteed for every initial state of the system and for every "schedule" of update-message arrivals. See formal exposition of this message-passing model in Section 2.

Research on traditional best-response dynamics established that determining, for a given game, whether best-response dynamics are guaranteed to coverge to a pure Nash equilibrium is, in general, intractable [Hart and Mansour 2010]. What are the implications of adding asynchrony to the mix? This question can inspire two totally different, and mutually contradictory, intuitions. On the one hand, introducing asynchrony implies a much smaller space of convergent systems, and should therefore render the task

of determining whether a system falls in this category easier. On the other hand, asynchrony also implies a "blow up" in the number of possible sysem configurations (e.g., update-messages in transit), thus making verifying guaranteed convergence harder. We show that both these intuitions are, in fact, false; verifying guranteed convergence is sometimes easy in our message passing model but hard in the traditional model, and vice versa. We now present our (positive and negative) results along these lines.

1.2. Negative Results

We explore the complexity of verifying guaranteed convergence to an equilibrium in our message-passing model — a task we term "Is-Convergent". Our main negative result establishes that Is-Convergent is, in fact, *undecidable*!

Theorem: Is-Convergent is undecidable even for a constant number of agents.

Observe that, in contrast, verifying guaranteed convergence in the traditional best-response dynamics model is in P for a constant number of agents (as one can simply examine all possible configurations of the system and check for loops). Our proof of the above theorem suggests that asynchronous best-response dynamics are sufficiently expressive to capture nondeterministic computations in the sense that such mutli-agent interaction systems can be regarded as a computational artifact where the update-message queues are the "storage" and agents' best-responses simulate the "logic". We leverage this idea to show a long and delicate reduction from the notoriously undecidable HALT-ALL—the language of Turing machines that halt on every input—to Is-Convergent in our message-passing model.

Our reduction consists of three main steps: We first establish, via a subtle reduction, that an interesting variant of HALT-ALL we call "HALT-ALL-CONFIG", in which the objective is to determine whether a given Turing machine halts for every input *and* possible initial configuration of the system (even "illegal" configurations), is also undecidable. We next reduce from HALT-ALL-CONFIG in the Turing machine model to an analogous problem in the (computationally-equivalent) Queue Automaton model. This step lays the foundation for utilizing message-queues as a storage element. The last and most delicate step of our reduction is constructing a game for which best-response dynamics converge in our message-passing model if and only if the given queue automaton halts from every initial configuration. A basic building block in this step of the proof is a game called the "Increment Game", which provides us with the means for synchronizng the system to the extent that any dynamic that does not obey a specific set of rules must converge. We believe that some of the ideas and techniques presented in our proof (e.g., HALT-ALL-CONFIG and the Increment Game) will prove to be useful in the analysis of other asynchronous game-theoretic dynamics.

We complement this computational-complexity result with an exponential, essentially tight, communication-complexity lower bound.

Theorem: Is-Convergent requires exponential communication (in n) even for a constant number of strategies per agent and when agents' best-responses are unique.

This result can be viewed as the analogue, in our asynchronous setting, of the communication complexity results of Hart and Mansour [Hart and Mansour 2010] for verifying guaranteed convergence of traditional best-response (and better-response) dy-

namics[1]. Our proof of this theorem involves an interesting combinatorial argument that makes it possible to reason about agents' beliefs about other agents' actions in the course of the convergence process.

1.3. Positive Results

Our above hardness results for Is-Convergent establish that (1) Is-Convergent is undecidable for a constant number of agents (eleven) with an arbitrary number of strategies per-agent; and (2) Is-Convergent requires exponential communication for an arbitrary number of agents with a constant number (five) of strategies per agent. We show that, in spite of these discouraging results, Is-Convergent is tractable for two relatively broad classes of games.

Our positive results rely on *characterizing* convergent systems. Our main positive result along these lines is the following. We focus on the well-studied scenario that agents have unique best-responses. We prove that the highly-restrictive game-theoretic category of dominance-solvable games—games in which the iterated elimination of dominated strategies results in a single strategy per agent—fully characterizes convergent systems in two complementary settings.

Theorem: When agents' best-responses are unique, dominance solvability fully characterizes guaranteed convergence

— for systems with an arbitrary number of agents and 2 strategies per agent;
— for 2-agent systems with an arbitrary number of strategies per agent.

This characterization of convergent systems has important implications for the (computational and communication) complexity of Is-Convergent. Our characterization implies, in particular, that when agents' best-responses are unique, Is-Convergent is tractable in the two above settings. This result should be contrasted with the hardness result in [Hart and Mansour 2010] for verifying guaranteed convergence of traditional best-response dynamics in systems with an arbitrary number of agents and 2 strategies per agent. We discuss a surprising corollary of this theorem: verifying stability of an asynchronous circuit is in P.

We also show that Is-Convergent is tractable for 2-agent systems even when agents' best-responses are not (necessarily) unique.[2] Our proof of the latter fact highlights an interesting connection between convergence in such systems and permutations over finite spaces.

Theorem: Is-Convergent is in P for two-agent systems.

1.4. More Results: Observability and Randomness

Our message-passing model captures realistic scenarios in which agents' actions are not immediately observable to all other agents (as in Internet protocols, etc.). We also consider the simpler model of agent interaction in [Jaggard et al. 2011]—the "observable-actions model"—which, intuitively, captures a lower level of asynchrony in which agents' actions are observable to all, but arbitrary subsets of agents can act simultaneously in an uncoordinated manner. We show that Is-Convergent is PSPACE-complete in this model, thus closing an open question posed in [Jaggard et al. 2011].

[1] [Hart and Mansour 2010] shows that determining that a game is a potential game (and hence that better-response dynamics are guaranteed to converge to a pure Nash equilibrium) requires exponential communication.

[2] Observe that solving Is-Convergent is not a trivial task even for two-agent systems as the length of agents' message queues is unbounded (and thus the number of possible states of the system is also unbounded).

Theorem: Is-Convergent is PSPACE-complete in the observable actions model even when agents' best-responses are unique.

Our proof of this theorem relies on an interesting "convergence preserving reduction".

A natural question in light of the above results is: "What happens when randomness is introduced into this setting?". We extend the above PSPACE-completeness result for the observable-actions model to a stochastic variant of Is-Convergent, which we call Is-Stochastically-Convergent.

Theorem: Is-Stochastically-Convergent is PSPACE-complete in the observable actions model.

1.5. Future Research

We believe that we have but scratched the surface in the exploration of game-theoretic dynamics under asynchrony. Two important directions for future research are the following:

(1) **Looking at subclasses of games.** Our hardness results for Is-Convergent do not impose *any* constraints on the class of games under consideration. We believe that the key to obtaining more positive results is to examine specific classes of games and investigate the complexity of Is-Convergent (e.g., is Is-Convergent tractable for potential games?).
(2) **Looking at other game-theoretic dynamics.** We have restricted our attention here to the convergence of (asynchronous) best-response dynamics to a pure Nash equilibrium. Understanding the convergence properties of other well-studied game-theroetic dynamics (e.g., fictitious play, no-regret dynamics, etc.) under asynchrony is an important research direction. An interesting recent result in this vain [Lehrer and Lagziel 2012] shows that no-regret dynamics exhibits good convergence properties even under asynchrony (see also [Jaggard et al. 2011]).

Research along the above lines could provide useful insights into environments in which simple and natural distributed protocols are guaranteed to converge to a "stable", "globally-rational", outcome, and also shed light on the behavior of existing protocols.

1.6. Related Work

The immediate precursor of our work is [Jaggard et al. 2011], which brings together research in distributed computing theory and in game theory to study asynchronous dynamic environments where computational nodes' behavior is simple, natural and myopic ("adaptive heuristics" in the language of Hart [Hart 2005]). Our message-passing model extends the framework of [Jaggard et al. 2011] to capture the possibility of outdated information, thus making it possible to reason about the behavior of best-response dynamics in a variety of real-life environments. Our work also rests on the extensive research on best-response dynamics in game theory, on protocol termination in distributed computing, and on networking research on Internet protocols.

Game theory. Much work in game theory and economics deals with best-response dynamics (e.g., [Monderer and Shapley 1996; Osborne and Rubinstein 1994; Rosenthal 1973]). Best-response dynamics have also been approached from a computer science perspective (e.g., [Fabrikant and Papadimitriou 2008; Mirrokni and Skopalik 2009]). Generally speaking, such research has thus far primarily concentrated on synchronized environments in which steps take place in some predetermined prescribed order and agents' choices of strategies are immediately observable to other agents. Nisan et

al. [Nisan et al. 2008] analyze the behavior of best- and better-response dynamics in asynchronous computational environments for restricted classes of games (dominance-solvable games and potential games). Lagziel and Lehrer [Lehrer and Lagziel 2012] recently showed that no-regret guarantees are achievable even when the decision maker is informed of the actual payoffs with delay. This implies that (a specific class of) no-regret dynamics converges to an equilibrium in various environments even under asynchrony. See also some simple observations about no-regret dynamics and asynchrony in [Jaggard et al. 2011].

Distributed computing. Over the past three decades, much work has been devoted to characterizing the possibility/impossibility borderline for fault-tolerant computation (see [Lynch 1989; Fich and Ruppert 2003]). While the risk of protocol non-termination is such environments stems from the possibility of failures of nodes or other components, our focus is on settings where the computational nodes and the communication channels are reliably failure-free, and the risk of non-convergence stems from limitations imposed by simplistic node behaviors. Convergence of best-response dynamics to an equilibrium in our models can be viewed as the self stabilization [Dolev 2000] of such dynamics. Recent work on "population protocols" [Aspnes and Ruppert 2007] also deals with convergence of local interactions to a globally stable state, but considers very different environments (nodes are indistinguishable, computation is restricted to direct interaction between two nodes, etc.) and thus does not capture adaptive heuristics [Hart 2005] such as best-response dynamics.

Internet protocols. Research on the dynamics of the Border Gateway Protocol (BGP) [Fabrikant and Papadimitriou 2008; Gao and Rexford 2002; Griffin et al. 2002; Suchara et al. 2011] can be viewed as analyzing convergence of best-response dynamics in the presence of asynchrony in a specific routing environment. Our message-passing model abstracts the standard model for analyzing BGP dynamics, put forth in [Griffin et al. 2002]. Our model also abstracts the model of congestion control on the Internet in [Godfrey et al. 2010].

1.7. Organization

We present our model of asynchronous best-response dynamics in Section 2. Our undecidability result for Is-Convergent is presented in Section 3 and the complementary communication complexity lower bound appears in Section 4. We present our characterizations of convergent systems and discuss their algorithmic implications (e.g., to asynchronous circuits) in Section 5, and present our algorithm for Is-Convergent for two-agent systems in Section 6. Our results for the observable-actions model and for Is-Stochastically-Convergent appear in Section 7.

2. MODEL

We now present our main model of asynchronous best-response dynamics. We will consider other (including randomized) models of system dynamics in Section 7.

Games. We study the following standard n-agent game-theoretic setup: n agents (or players) $1, \ldots, n$, each with finite strategy space S_i and a utility function $u_i : S_1 \times \ldots \times S_n \to R_+$. Let $S = \prod_{j \in [n]} S_j$ denote the space of strategy-vectors and $S_{-i} = \prod_{j \in [n] \setminus \{i\}} S_j$ denote the space of strategy-vectors of all agents but the i'th. We use (s_i, s_{-i}) as shorthand for the strategy-vector in which the strategy of agent i is $s_i \in S_i$, and the strategies of of all other agents are as in $s_{-i} \in S_{-i}$. Strategy $s_i^* \in S_i$ is a "best response" of agent i to $s_{-i}^* \in S_{-i}$ if $s_i^* \in \operatorname{argmax}_{s_i \in S_i} u_i(s_i, s_{-i}^*)$. Strategy-vector $s^* = (s_1^*, \ldots, s_n^*) \in S$ is a "(pure Nash) equilibrium" if, for every $i \in [n]$, s_i^* is a best response to s_{-i}^*.

Best-response dynamics. Under (traditional) best response dynamics, agents take turns best-responding to other agents' strategies. Whenever an agent i is "active", he examines the strategies of the others $s^t_{-i} \in S_{-i}$ and, in the event that his strategy is not a best-response to s_{-i}, i updates his strategy to a best-response strategy.

Best-response dynamics and asynchrony. We present a model of asynchronous best-response dynamics that captures the possibility of non-instantaneous propagation of information about agents' strategy changes. Our model unifies and abstracts the models of BGP routing and of congestion control in [Griffin et al. 2002] and [Godfrey et al. 2010], respectively. In our model, whenever an agent selects a new best-response strategy he sends an update message to every other agents announcing his newly chosen strategy over a private communication channel between them. Whenever an agent $i \in [n]$ receives an update-message from another agent $j \neq i$, this event, called RECEIVE(I,J), triggers an atomic step in which agent i executes the following three steps:

(1) i examines the most recently learned strategies of other agents (his "beliefs" about others' agents) $s^B_{-i} \in S_{-i}$ (stored in a locally-maintained data structure);
(2) i updates his strategy s_i to a best response strategy s^*_i to s_{-i} (if s_i is a best-response to s_{-i} i does not select a new strategy);
(3) if $s^*_i \neq s_i$, i sends an update message to every other agent announcing s^*_i.

Transmission along every communication channel can experience arbitrary (channel-specific) update-message delays. We make the assumption that each channel maintains a FIFO queue of the sent messages (in each direction), and so messages are received in the order of transmission.

We call a complete specification of the order of events of the form RECEIVE(I,J), i.e., update-message arrivals, a "schedule". A is schedule "fair" if no agent is indefinitely starved from receiving messages from another agent. We call a system "convergent" if, for every initial combination of the agents' strategies and fair schedule, from some point in time agents' strategies constitute a pure Nash equilibrium.

3. IS-CONVERGENT IS UNDECIDABLE

Our main negative result establishes that, in general, Is-Convergent is undecidable.

THEOREM 1. *Is-Convergent is undecidable for* 11*-agent systems.*

We prove Theorem 1 via a long and delicate chain of reductions in the full version. Here, we give a high-level overview of the proof. We then present in more depth a basic building block in our construction, the so called "Increment Game".

3.1. High-Level Overview of Proof

High-level ideas. Intuitively, the hardness of Is-Convergent in our message-passing model stems from the fact that nodes' message-queues are unbounded. This suggests that best-response dynamics in this model might be expressive enough to capture non-deterministic computations: agents' best-responses implement the "logic", message queues function as "storage", and the nondeterminism is captured by the fact that the schedule of events is (almost) arbitrary. To formalize this intuition, we identify a computational model that is somewhat "close" to our setting—the Queue Automaton model[3]. While in some sense similar to our context, queue automata are also very different, and so must be approached with caution. Differences include (1) the fact that

[3]Queue Automaton is a computational model equivalent to Turing Machine. While similar to a Pushdown Automaton, instead of a stack it uses a queue.

in our setting there is no clear notion of input; (2) in the Queue Automaton model the initial configuration is predetermined, whereas in our model it is adversarially chosen; and (3) in our context agents' responses are based on very limited (almost no) "memory", and so the intricate interactions and dependencies between agents are key to achieving the desired result. To address the first point, we show a reduction from Halt-All—the language of Turing machines that halt on every input. By doing so, we effectively eliminate the need to have an input (and essentially generate one in a nondeterministic fashion using our nondeterministic schedules). To handle the second point, we present and study a variant of Halt-All, called "Halt-All-Config". A Turing machine is in Halt-All-Config if it halts for every input *and* starting configuration (even "illegal" configurations). We discuss our results for Halt-All-Config below. The last of these points is addressed by presenting a basic building block that introduces synchronization between players. The synchronization is levereged to construct an analogue concept of memory within the game. We now discuss the three steps in our reduction from Halt-All to Is-Convergent.

Step I: Halt-All to Halt-All-Config. The first reduction is from Halt-All to Halt-All-Config, and as explained above, it provides the context of computation from an arbitrary configuration. Given an input to Halt-All—a Turing machine M—the reduction constructs a Turing machine Q such that Q halts from any starting configuration if and only if M halts on every input. We stress that for any arbitrary start configuration of Q, even an "illegal" configuration that does not match any computation of M, Q must either reach a "legal" configuration, or halt. To achieve this, Q keeps track of the entire history of the simulation thus far and verifies that the history is valid after simulating each and every execution step of M. Q's construction ensures that, regardless of its initial state, Q is guaranteed to execute this verification process (unless it halts earlier). In the event that the verification fails, Q halts. Otherwise, we prove that Q successfully simulates the execution step in M and the execution of Q can continue without risking the crucial properties of the reduction, as described above. We defer the many subtle details of this first reduction to the full version.

Step II: Turing Machines to Queue Automata. The second step is reduction from Halt-All-Config to its analogue in the Queue Automaton model, Halt-All-Config-Queue. The motivation for moving to queue automata is the storage model, the queue, which makes reduction to our setting easier. While this might seem as a technical step[4], this too requires special care. Again, merely simulating the Turing machine by a queue automaton is not enough. When starting from a configuration that is not a valid simulation step of the given Turing machine, the queue automaton should either get to a configuration that represents a valid simulation step of the given Turing machine or halt within in a finite number of steps. Our reduction from Halt-All-Config to Halt-All-Config-Queue, which requires modifying the classical reduction for simulating a Turing machine with a queue automaton, is described in the full version.

Step III: Queue Automata to Is-Convergent. The last and most subtle step in our reduction is a reduction from Halt-All-Config-Queue to Is-Convergent. We need to be able to simulate any execution of a given queue automaton (starting from an arbitrary configuration) via best-response dynamics. The key element in our constructed game is a message queue that simulates the queue of the queue automaton. However, to utilize this message queue we must overcome two main challenges:

[4]One might consider skipping the Turing machine model by providing a reduction from the Halt-All variant for queue automata to Halt-All-Config-Queue, yet this is actually far more tedious and nonintuitive, as the queue automaton model is harder to work with.

(1) **Synchornization.** We must eliminate schedules in which messages are removed from the queue "too fast" or "too slowly". In other words, we must somehow synchronize the system to ensure that it evolves in a desired way.
(2) **Memory.** We need to emulate the state machine of the queue automaton, which, in turn, requires a notion of memory in the game.

Our solutions to these two challenges turn out to be strongly related. At the heart of our construction is a building block we term the "Increment Game". Informally speaking, in the Increment Game a single "*leader*" increments a finite counter, while all the other players, the "*followers*", follow the leader's strategy. Careful design guarantees that regardless of the initialization of this game, and of the timing of update message arrivals, the resulting sequence of the leader's chosen strategies is an infinite cyclic sequence of integers. We can therefore use the Increment Game as an internal "system clock" and utilize it to effectively synchronize the system. We introduce memory into the game by tagging the counter with labels that can be manipulated by the players. We show how the Increment Game can be used, in an input-output programming-like fashion, as a building block in a modular design of games. Our construction combines two Increment Games; one Increment Game serves as a Writer—simulating the state transitions of the queue automaton and the writing of symbols into the simulated queue; the other serves as a Reader—reading symbols from the queue head and feeding them back to the Writer.

We prove that (well-behaved) best-response dynamics in the constructed game reflect executions of the queue automaton, and that such dynamics converge to a pure Nash equilibrium if and only if the corresponding execution halts.

3.2. The Increment Game

Here, we give a "taste" of the proof by zooming in on a crucial building block in the last, and most subtle, step of our reduction—the Increment Game. As explained above, the Increment Game effectively synchronizes an inherently asynchronous system, and also provides the means for carrying out complex computations. Below we sketch only how the Increment Game achieves the first of these goals: synchronizing the system. We explain the various more complex functions of the Increment Game in our reduction, and how these are realized, in the full version.

Increment Games. In an Increment Game there are $n \geq 3$ players, all with identical best-response functions, with the exception of a single player, who shall be referred to hereafter as the *Leader* or Inc. Inc's strategy space contains the elements in Z_r for a fixed $r \geq 5$ and another unique strategy called the "*halt* strategy". Intuitively, Inc is in charge of incrementing the global index (modulo r), thus guaranteeing all other players, which we refer to as the *Followers*, do the same. The Followers' strategy spaces include all strategies of the form $(c, bool)$, where $c \in Z_r$ and $bool$ is either true or $false$, and also the *halt* strategy. Intuitively, c in a pair $(c, bool)$ represent the counter being incremented (which we simply refer to as the *counter*), while $bool$ indicates whether the Follower sees any other Followers with a smaller counter than his. In what follows, additions and subtractions of the counter are modulo r.

Valid configurations. Consider a strategy vector s and suppose that Inc's strategy in s is some $C \in Z_r$. We say that s is *valid* if it satisfies at least one of (1) the counters of all Followers equal C, *or* (2) the counters of all Followers equal $C - 1$, *or* (3) the strategies of all Followers are either $(C, true)$ or $(C - 1, false)$.

We say that player i *observes an invalid strategy-vector* s_{-i} if there is no strategy $s_i \in S_i$ such that (s_{-i}, s_i) is a valid strategy vector.

Best-responses in the Increment Game. We define players' best-responses in the Increment Game so as to ensure the following properties: (1) the *halt* strategy is always a best-response (for any player and any strategy-vector) and so once a player chooses the *halt* strategy he will never deviate from it; (2) the *halt* strategy is a player's *sole* best-response to all invalid strategy vectors, and also to all strategy vectors in which some other player's strategy is *halt*.

Observe that once even a single player halts, the game is bound to reach an equilibrium in which all players play *halt* within a finite number of steps. We call an initial strategy-vector and (fair) schedule for which the system does not converge to this "all-halt equilibrium" a *non-halting evolution* of the system.

We are left with defining a player's best-response to a valid strategy vector s:

— **Leader:** Let c be the maximum counter of a Follower in s. Then, the best-response of the Leader is $c+1$.
— **Follower:** Let C be Inc's strategy in s. If there is another Follower whose strategy is $C-1$, the best-response of the Follower is $(C, true)$. Otherwise (all other Followers' counters are C), the best-response is $(C, false)$.

Why are Increment Games useful? Increment Games are carefully designed to have the following useful properties, which we repeatedly leverage in our reduction.

THEOREM 2 (INCREMENT). *In any Increment Game G*

— *The strategy sequence of the Leader in any non-halting evolution has the form $\ldots, C, C+1, \ldots, r-1, 0, 1, \ldots, r-1, 0, 1, \ldots$*
— *A non-halting evolution exists.*

Below we only prove that there indeed exists a non-halting evolution of the system where the strategy sequence of the Leader is of the form $\ldots, C, C+1, \ldots, r-1, 0, 1, \ldots, r-1, 0, 1, \ldots$ We defer the more delicate proof that *any* non-halting evolution is of this nature (which plays a crucial role in our reduction) to the full version.

A non-halting evolution which mimics a system clock exists. We next exhibit such a non-halting evolution. Let the initial strategies and beliefs of all the players be that the Leader is playing 0 and that all the Followers are playing $(0, false)$. We first allow all initial messages from all Followers and the Leader to be received by all Followers. Observe that this does not change the Followers' strategies. We next let all the messages from the Followers reach the Leader. This causes the Leader to change his initial strategy from 0 to 1.

The system evolves in iterations, with each iteration i beginning in a global system configuration similar to that in the end of the above short initialization phase, in which (1) no Forwarders' messages are in transit and all the Leader's messages about the counter increment from $i-1$ to i are still pending; and (2) the Leader believes that all the Followers are playing $(i-1, false)$, and each Follower believes that the leader is playing $i-1$ and all other Followers are playing $(i-1, false)$. The i'th iteration consists of the following sequence of steps:

(1) The Leader's messages about his counter increment to i are received by all Followers (who respond by choosing $(i, true)$).
(2) All Followers' $(i, true)$ messages are received by all players. Once a Follower receives all $n-2$ such messages, he responds with $(i, false)$. Notice that by the end of this step no player observes any Follower playing $(i-1, false)$. The Leader responds with $i+1$ messages which are not received until the next iteration.
(3) All Followers' $(i, false)$ messages are received by all the players.

Observe that this is indeed a non-halting evolution since each player observes a valid strategy-vector at all times.

4. COMMUNICATION COMPLEXITY LOWER BOUND

We present the following exponential communication lower bound for Is-Convergent.

THEOREM 3. Is-Convergent requires $\Omega(\alpha^n)$ communication for some constant $\alpha > 1$ even when agents' best-responses are unique.

We now present a very high-level exposition of the proof of Theorem 3. See full proof in the full version. We show a reduction from the classical communication Set Disjointness setting, in which there are n parties $\{1, \ldots, n\}$, each party i holding a subset $A_i \subseteq \{1, \ldots, t\}$, and the objective is to distinguish between the two following extreme scenarios: (1) $\bigcap_{i=1}^{n} A_i \neq \emptyset$, vs. (2) for every $i \neq j$, $A_i \cap A_j = \emptyset$.

Classical results in communication complexity establish that solving Set Disjointness entails (in the worst case) transmitting $\Omega(\frac{t}{n})$ bits [Alon et al. 1996; Nisan 2002].

We construct, for a given instance of Set Disjointness, an instance of Is-Convergent with n agents, representing the parties in Set Disjointness, each with action space $\{0, 1, 2, 3, 4\}$. We prove that our construction ensures that if $\bigcap_{i=1}^{n} A_i \neq \emptyset$ in Set Disjointness then the constructed Is-Convergent instance is not convergent, and if for every $i \neq j$, $A_i \cap A_j = \emptyset$, the constructed Is-Convergent instance is convergent.

We first establish the existence of an exponentially-large family of vectors in $\{0, 1, 2, 3\}^n$ with useful combinatorial properties.

LEMMA 4. Given $n > 0$, there exists a set $V \subseteq \{0, 1, 2, 3\}^n$ such that

— the hamming distance between every two vectors in V is strictly greater than $\frac{2n}{3}$;
— for every vector $(v_1, \ldots, v_n) \in V$, $((v_1 + 1) \mod 4, \ldots, (v_n + 1) \mod 4) \in V$ as well;
— $|V| = \Omega(\alpha^n)$ for some constant $\alpha > 1$.

We associate each element of $\{1, \ldots, t\}$ in Set Disjointness (for a large enough t) with a distinct set of 4 vectors in V that are closed under the $(+1 \mod 4, \ldots, +1 \mod 4)$ operation. We let V_e denote the set of 4 vectors in V associated element $e \in \{1, \ldots, t\}$. We now define each agent's best-responses in the Is-Convergent instance as follows: (1) in the event that the other agents' strategies (as seen by agent i) are as in some vector $v \in V_e$ for $e \in A_i$, agent i's (unique) best-response is $(v_i + 1) \mod 4$; (2) otherwise, i's best-response strategy is always 4.

We let I denote the given Set Disjointness instance, and $IC(I)$ denote the constructed instance of Is-Convergent in the message. We prove the following lemma.

LEMMA 5. IC(I) is not convergent if $\bigcap_{i=1}^{n} A_i \neq \emptyset$ in I, and is convergent if for every $i \neq j \in [n]$, $A_i \cap A_j = \emptyset$ in I.

As in our reduction $t = \Omega(\alpha^n)$ for some constant $\alpha > 0$, the lower bound on communication for Set Disjointness now implies Theorem 3. We now give the high-level intuition for the proof of Lemma 5.

Consider first the (simpler) scenario in which $\bigcap_{i=1}^{n} A_i \neq \emptyset$ in I. Consider a vector $v = (v_1, \ldots, v_n) \in V_e$ for an element $e \in \bigcap_{i=1}^{n} A_i$ and observe that in our construction of $IC(I)$ allowing all agents to best respond to v simultaneously transitions the system to the strategy-vector $((v_1 + 1) \mod 4, \ldots, (v_n + 1) \mod 4) \in V_e$. We conclude that, in this scenario, when agents' initial strategy-vector is in V_e, agents' repeated simultaneous best responses induce a (fair) oscillation in $IC(I)$ between strategy-vectors in V_e.

We now turn our attention to the more challenging scenario in which for every $i \neq j \in [n]$, $A_i \cap A_j = \emptyset$ in I. We observe that every fair oscillation must involve multiple agents repeatedly selecting strategies in $\{0, 1, 2, 3\}$ which, in turn, implies that agents

see strategy-vectors in V infinitely often. Let $V_i = \bigcup_{e \in A_i} V_e$. We prove that as the hamming distance between every two vectors in V is large, and as $V_i \cap V_j = \emptyset$ for every $i \neq j \in [n]$, our construction ensures that (1) agents' views of other agents' strategies must change drastically over time; and (2) agents' views of other agents' strategies must be very inconsistent across nodes. We show that the number of generated update messages is not sufficient to maintain such contradicting views.

5. CHARACTERIZATION RESULTS

We now focus on the well-studied scenario that agents have unique best-responses. We show that the game-theoretic category of dominance-solvable games fully-characterizes convergent systems in two complementary settings: (1) for an arbitrary number of agents with 2 strategies per agent; and (2) for 2-agent systems with an arbitrary number of strategies per agent. We discuss the interesting implications of this characterization to the (computational and communication) complexity of Is-Convergent.

5.1. Dominance Solvability *vs.* Convergence

An agent's strategy is said to be dominated if it is never a best-response strategy.

Definition 5.1. A game is *dominance-solvable* if the iterated removal of dominated strategies results in a single strategy-vector $s^* \in S$.

Dominance-solvable games can easily be seen to be convergent (see [Nisan et al. 2008]). We now show that the converse is also true in two interesting contexts and so, in these contexts, dominance-solvability fully-characterizes guaranteed convergence. (See proofs in the full version.)

THEOREM 6. *When agents' best-responses are unique, dominance solvability fully characterizes guaranteed convergence:*

6(a). for multi-agent systems with 2 strategies per agent;
6(b). for 2-agent systems.

We will now prove the more challenging part of the Theorem 6(a). We defer the proof of 6(b) to the full version.

5.2. Proof of Theorem 6(a)

Consider a multi-agent system Θ where each agent has (at most) 2 possible strategies. As observed in [Nisan et al. 2008], if Θ is dominance-solvable, then it is convergent in our message-passing model (presented in Section 2). Hence, we are left with showing that if Θ is not dominance-solvable then it is also not convergent. We will now show that if Θ is not dominance-solvable then there indeed exists a fair oscillation.

The act-and-tell model. To prove this result we consider a model of agent interaction that we term the "*act-and-tell model*", which captures a simplified variant of model and is inspired by the state-transition graph in [Sami et al. 2009]. We show that a non-convergence result in the act-and-tell model also holds in our message-passing model, and so, to prove Theorem 6(a), we can restrict our attention to the act-and-tell model.

In the act-and-tell model, as in our message-passing model, the system evolves from an initial combination of agents' strategies over infinitely-many discrete time steps and, at each time step, a subset of the agents is activated. In this new model, though, whenever an agent i is activated, it must choose between the two following possibilities.

— **Act**: pick a best-response strategy in S_i to what i thinks are the current strategies of the other agents. i's choice of strategy is *not* immediately observable to other nodes.
— **Tell**: send a strategy-update to one other agent j, informing j of i's current strategy. This message arrives at node j *immediately* (i.e., at the end of that time step).

Thus, in the act-and-tell model, whenever an agent is activated it can either update its strategy to be a best-response to its view of others, or tell a single other agent about its current strategy. We call a schedule of agent activation "fair" if (1) each agent i "acts" (i.e., updates its strategy) in infinitely many time steps; and (2) each agent i "tells" every other agent j (i.e., sends a strategy-update to j) in infinitely many time steps. The act-and-tell model is much easier to reason about than our message-passing model as there are never messages in transit (when a strategy-update is sent it arrives immediately).

We prove (the proof, which follows in the footsteps of [Sami et al. 2009], is omitted) that if Θ is not dominance-solvable there must exist a fair oscillation in the act-and-tell model. The following claim shows that this implies a fair oscillation in the message-passing model of Section 2.

CLAIM 7. *If there exist an initial strategy-vector and a fair schedule for which the system dynamics oscillates indefinitely in the act-and-tell model then there also exist an initial strategy-vector and a fair schedule for which the system dynamics oscillates indefinitely in the message-passing model.*

Two or more pure Nash equilibria imply an oscillation. [Jaggard et al. 2011] shows that in a large variety of settings the mere existence of two stable states to which dynamics can converge implies the possibility of a persistent oscillation (see [Jaggard et al. 2011] for a formal exposition of this result). We observe that the result in [Jaggard et al. 2011] (which is proven in the observable-actions model) implies the following.

COROLLARY 8. *[Jaggard et al. 2011] If agents have unique best-responses and multiple pure Nash equilibria exist, the system is not convergent in the act-and-tell model.*

Hence, we can henceforth restrict our attention to the case that Θ contains a unique stable state (for otherwise a fair oscillation exists and so 6(a) follows). Observe that if some agent has a dominated strategy then Θ is convergent iff the sub-system obtained from Θ via the elimination this dominated strategy is convergent. Hence, we shall also make the assumption that no agent in Θ has a dominated strategy.

System dynamics under traditional best-response dynamics. To construct a fair oscillation in the act-and-tell model, we first consider the system dynamics under traditional best-response dynamics. Let $S_i = \{0, 1\}$ for each agent i (recall that $|S_i| = 2$). W.l.o.g., let $\overrightarrow{1} = (1, \ldots, 1)$ be the unique pure Nash equilibrium of the system. Now, consider the evolution of the system under traditional best-response dynamics when the initial state is $\overrightarrow{0} = (0, \ldots, 0)$ and agents are activated one at a time in some arbitrary predefined cyclic order. In the event that the resulting system dynamics do not eventually reach the pure Nash equilibrium $\overrightarrow{1}$ we have found a fair oscillation of traditional best-response dynamics, which can easily be shown to imply a fair oscillation in the act-and-tell model (using arguments similar to those in the following paragraph). We need therefore only consider the case that the resulting system dynamics converges to the equilibrium $\overrightarrow{1}$. Observe that, for each agent i, there exists a unique point in time t_i where i's strategy changed from 0 to 1 and never changed back to 0 thereafter. W.l.o.g., let $t_1 < t_2 < \ldots < t_n$.

System dynamics in the act-and-tell model. We now observe that the system behavior in the act-and-tell model under the following dynamics mimics the system behavior under traditional best-response dynamics, as described above. Fix the same cyclic order over the agents as before. Start at the initial state $\overrightarrow{0}$. Repeatedly allow a single agent, chosen according to the cyclic order, to update its strategy ("act") and then immediately inform all other agents of its newly chosen strategy ("tell" all others) until the system has reached the same state as after t_1 time steps in the original best-response dynamics. Let $\overline{t_1}$ be the point in time at which this happens. From time $\overline{t_1}$ onwards do not activate agent 1 again. Now, continue activating agents $2, \ldots, n$ according to the same cyclic order (after 1 is removed), letting each agent update its strategy and then immediately informing all other agents in $\{2, \ldots, n\}$ of its newly chosen strategy (but not agent 1). This continues until the system has reached the same state as after t_2 time steps in the original best-response dynamics. Let $\overline{t_2}$ be the point in time at which this happens. From time $\overline{t_2}$ onwards do not activate node 2 again. And so on. Observe that the system behavior in the act-and-tell model under these dynamics indeed mimics the system behavior under traditional best-response dynamics, as described above, and, in particular, reaches the unique pure Nash equilibrium $\overrightarrow{1}$.

Constructing a fair oscillation in the act-and-tell model. Consider an agent i. As 0 is not a dominated strategy for i there must exist some strategy vector $x_{-i} \in \{0,1\}^{n-1} = S_{-i}$ such that i's (unique) best-response to x_{-i} is 0. Consider the evolution of the system in the act-and-tell model described above. We make the following crucial observation. Consider the system after time $\overline{t_1}$, i.e., after agent 1 changes its strategy from 0 to 1 and is never activated again. Observe that, at that point in time, every other agent in Θ either has strategy 0 and will select strategy 1 at a later point in time (as its strategy in the equilibrium is 1) or has strategy 1 and will select strategy 0 at a later point in time (for otherwise that would contradict the definition of t_1). Hence, after time $\overline{t_1}$, by allowing agents $\{2, \ldots, n\}$ to "tell" agent 1 of their astrategies at the appropriate times we can make 1's view of the other agents' strategies be precisely x_{-1}. Similarly, after time $\overline{t_2}$ every agent in $\{3, \ldots, n\}$ either has strategy 0 and will select strategy 1 at a later point in time or has strategy 1 and will select strategy 0 at a later point in time. Hence, after time $\overline{t_1}$, by allowing agents $\{3, \ldots, n\}$ to "tell" agent 2 of their strategies at the right times we can make 2's view of the strategies of the agents in $\{3, \ldots, n\}$ be as in x_{-2}. In general, after $\overline{t_i}$ time has passed, it is possible to ensure that agent i's view of the strategies of the agents in $\{3, \ldots, n\}$ be precisely their strategies in x_{-i}. Observe, though, that agent i's view of the strategy of every agent in $\{1, \ldots, i-1\}$ is 1.

Now, consider the time when the pure Nash equilibrium $\overrightarrow{1}$ is reached — when each agent's strategy is 1. By the above arguments, agent 1's view at that time is x_{-1}, agent 2's view is x_{-2}, but with a possible exception of agent 1's strategy, agent 3's view is x_{-3}, but with a possible exceptions of agents 1 and 2's strategies, etc. We can now ensure that each agent i's view be exactly x_{-i} as follows. Allow agent 1 to update its strategy to 0 (recall that its view is x_{-1}) and tell its new strategy to every agent $i > 1$ such 1's strategy in x_{-i} is 0. Observe that at this point, agent 2's view is precisely x_{-2}. The next step is allowing agent 2 to update its strategy to 0 and tell all agents $i > 2$ for which 2's strategy in x_{-i} is 0. This process enables us ensure that all agents update their strategies to 0. We now allow all agents to tell all other agents their current strategies (that are 0 for all agents). Observe that we are now back at the initial state of the dynamics. Note also that in this process all agents were allowed to "act" and also to "tell" all other agents. Thus, we have constructed a fair oscillation.

5.3. Algorithmic Implications

Observe that as agents' message queues are unbounded, the system dynamics cannot, in general, be described by a finite "state-transition graph". Importantly, the above characterization shows that, in spite of this, convergent systems do have a succinct description in interesting contexts.

When determining whether a strategy is dominated is a tractable feat, the following simple algorithm can be executed to verify the dominance-solvability of a given game: Go over agents in some cyclic (round-robin) order and iteratively eliminate dominated strategies until each agent has a single surviving strategy (in which case the game is dominance-solvable) or no more dominated strategies exist yet some agent has more than a single surviving strategy (in which case the game is not dominance-solvable). Thus, in this scenario Is-Convergent is solvable in $O(\Sigma_i|S_i|)$ time.

Unfortunately, determining whether a strategy in S_i is dominated can be NP-hard (simple proof omitted). We observe, however, that the fact that whether a strategy in S_i is dominated depends solely on agent i's utility u_i immediately implies the following communication complexity upper bound.

COROLLARY 9. When agents' best-responses are unique, Is-Convergent is solvable in a communication-efficient manner for an arbitrary number of agents with 2 strategies per agent.

This result should be contrasted with our exponential communication lower bound for 5 strategies per agent and unique best-responses (in Section 4).

Another scenario in which determining whether a strategy is dominated is tractable, and so verifying dominance-solvability is also tractable, is in the well-studied class of graphical games, i.e., when agents reside on a graph and each agent's utility is explicitly given as a function of his neighbors' strategies. Hence, another corollary Theorem 6 is the following.

COROLLARY 10. Is-Convergent $\in P$ in graphical games with 2 strategies per agent.

This fact has the following interesting implication for asynchronous Boolean circuits:

5.4. Application: Asynchronous Circuits

Work in computer architecture research on asynchronous circuits — circuits in which the components aren't governed by global clock — explores the implications of asynchrony for circuit design (see, e.g., [Davis and Nowick 1996]). [Jaggard et al. 2011] observes that asynchronous Boolean circuits can be modeled as best-responses dynamics in which agents' best-responses are unique and each agent has 2 possible strategies. Agents are then the logic gates, and the strategy space of each is the Boolean values $\{T, F\}$. See [Jaggard et al. 2011] for more details. The above results hence imply a complete characterization of "inherently stable" asynchronous Boolean circuits when changes in logic gates' outputs might only propagate to other circuit components after some delay. We observe that when the fan-in of each logic gate is a constant (as with all standard logic primitives), each node's reaction function in this formulation of asynchronous Boolean circuits is explicit and so the asynchronous circuit can be modeled as a graphical game with 2 strategies ("T" and "F") per agent (gate).

COROLLARY 11. Determining if an asynchronous Boolean circuit with feedback comprised of primitives with constant fan-in is inherently stable is in P.

6. ALGORITHM FOR 2-AGENT SYSTEMS

We next present a computationally-efficient algorithm for solving Is-Convergent for 2-agent systems (even when agents' best-responses are not unique), establishing:

THEOREM 12. *Is-Convergent is in P for two-agent systems.*

We point out that even when there are only two agents Is-Convergent is far from trivial as the number of possible configurations of the system is seemingly infinite (since agents' message queues are not bounded). Our algorithm for 2-agent systems exploits an interesting connection between guaranteed convergence and cycles in permuations over finite spaces. This positive result should be contrasted with our undecidability result for $n \geq 11$ agents in Section 3.

Notation and definitions. Before presenting the algorithm we introduce some notation and definitions. $BR_i(x)$ denotes the set of best-response strategies of agent i when the other agent's strategy is x. $BR_i^*(x)$ denotes the unique best-response strategy that agent i selects upon receiving an update that the other agent's strategy is x if i's strategy at that time is not a best-response to x.

A *strategy cycle* c is a sequence of $|c|$ strategies for each agent $i \in \{0,1\}$, $s_i^0, s_i^1, \ldots, s_i^{|c|-1} \in S_i$. Of special interest to us are cycles with the following *"transition property"*: for every $0 \leq i < |c|$, $s_2^i = BR_2^*(s_1^i)$ and $s_1^i = BR_1^*(s_2^{i-1 \mod |c|})$.

Definition 6.1. Two cycles with the transition property $c = \{s_1^i\}_{0 \leq i < |c|} \cup \{s_2^i\}_{0 \leq i < |c|}$ and $d = \{t_1^j\}_{0 \leq j < |d|} \cup \{t_2^j\}_{0 \leq j < |d|}$ *"can follow each other"* if there exists $0 \leq \delta < \alpha = \gcd(|c|, |d|)$ such that:

— For every $0 \leq i < |c|$ and $0 \leq j < |d|/\alpha$, $s_1^i \notin BR_1(t_2^{(i+\delta+|c|\cdot j) \mod |d|})$ and $s_2^i \notin BR_2(t_1^{(1+i+\delta+|c|\cdot j) \mod |d|})$

— For every $0 \leq j < |d|$ and $0 \leq i < |c|/\alpha$, $t_1^{(j+\delta+1) \mod |d|} \notin BR_1(s_2^{(j+|d|\cdot i) \mod |c|})$ and $t_2^{(j+\delta) \mod |d|} \notin BR_2(s_1^{(j+|d|\cdot i) \mod |c|})$

Algorithm. We are now ready to present our algorithm for 2-agent systems. Our algorithm consists of the following steps:

(1) Build a directed bipartite graph $G = (S_1, S_2, E)$, where S_i is the strategy space of agent i and $E = \{(s_1, s_2) | s_i \in S_i, BR_2^*(s_1) = s_2\} \cup \{(s_2, s_1) | s_i \in S_i, BR_1^*(s_2) = s_1\}$.
(2) Recursively remove any vertex with zero in-degree and its outgoing edges. Let L and R denote the remaining vertices on the lefthand and righthand) side, respectively. Note that each remaining vertex must have in- and outdegree 1, and that $|L| = |R|$.
(3) If $|L| = |R| = 1$, return CONVERGENT.
(4) Set $\pi_1(s_1) := BR_1^*(BR_2^*(s_1))$ for all $s_1 \in S_1$ and $\pi_2(s_2) := BR_2^*(BR_1^*(s_2))$ for all $s_2 \in S_2$. Note that π_1 and π_2 are permutations on L and R respectively; π_1 and π_2 have the same number of cycles; and each cycle in π_1 has a same-length corresponding cycle in π_2 (these two cycles are defined by the same cycle in the bipartite graph). Let $c_1^1, \ldots, c_1^m$ be the cycles in permutation π_1 and $c_2^1, \ldots, c_2^m$ be the cycles in π_2.
(5) Check, for all $k \in [m]$ such that $|c_1^k| > 1$, that both (1) for each $s_1 \in c_1^k$, $s_1 \notin BR_1(BR_2^*(s_1))$; and (2) for each $s_2 \in c_2^k$, $s_2 \notin BR_2(BR_1^*(s_2))$. Output NOT CONVERGENT if true for some $k \in [m]$.
(6) Check for every two (possibly identical) $i, j \in [m]$ whether c_i and c_j can follow each other and return NOT CONVERGENT if true for some $i, j \in [m]$.
(7) Return CONVERGENT.

The algorithm can easily be seen to be computationally-efficient. We prove (in the full version) that the algorithm returns CONVERGENT if and only if the two-agent system is indeed convergent.

7. OTHER MODELS: OBSERVABLE ACTIONS AND RANDOMNESS

We now consider best-response dynamics in the model studied in [Jaggard et al. 2011], which we call the "observable actions model". This model captures agent interaction when agents' actions are immediately observable to all other agents, but subsets of agents can act simultaneously in an uncoordinate manner. We study the complexity of Is-Convergent in this model. We then introduce and study a natural stochastic variant of Is-Convergent, which we call Is-Stochastically-Convergent.

7.1. Observable-Actions Model

7.1.1. The Model. **Agent interaction.** Agents interact over infinitely long discrete time $t = 1, 2, \ldots$. The system evolves from an initial state (combination of agents' strategies) $s^0 \in S$. At each time step $t \in N$, a subset of agents $\sigma(t) \subseteq [n]$ is "activated", and (if necessary) each activated agent $i \in \sigma(t)$ updates his strategy to be a best-response strategy to the others' current strategies (strategies of $i \notin \sigma(t)$ remain unchanged).

Convergent systems. A complete specification of which subset of nodes $\sigma(t)$ is activated at each time step $t \in N$ is called a "schedule". A schedule is "fair" if every agent $i \in [n]$ is activated infinitely many times, i.e., no agent is indefinitely starved from acting. In the observable actions model, a system is "convergent" if, for every initial state of the system and for every fair schedule, from some point in time onwards agents' strategies constitute a pure Nash equilibrium.

7.1.2. IS-CONVERGENT is PSPACE-Complete. Is-Convergent in the observable actions model is the task of determining if a given system is convergent in this model. In contrast to our undecidability result for Is-Convergent in our message-passing model, when the number of agents is constant, Is-Convergent in the observable-actions model is, in fact, in P. One can simply examine all possible configurations of the system (i.e., action profiles) and search for "fair" best-response loops in this polynomial-size graph. We now show the a computational hardness result for Is-Convergent in the observable-actions model, which closes an open question from [Jaggard et al. 2011].

THEOREM 13. *Is-Convergent is PSPACE-complete in the observable-actions model even when agents' best-responses are unique.*

We prove Theorem 13 in the full version by exhibiting a general "convergence-preserving reduction" from multi-agent systems in which agents' reactions only depend on the present to best-response dynamics when agents' best responses are unique.

7.2. Randomness Does Not (Always) Help

We also investigate the complexity of the stochastic variant of Is-Convergent in the observable-actions model, which we term Is-Stochastically-Convergent. In Is-Stochastically-Convergent, unlike Is-Convergent, the schedule is randomized, i.e., each agent i is activated with some nonnegative probability $p_i > 0$ at each time step, and the goal is to determine whether convergence to a stable state occurs with probability 1. This task is equivalent to checking whether the game is "weakly acyclic" [Young 1993; Mirrokni and Skopalik 2009; Fabrikant et al. 2010], i.e., that from every strategy-vector there exists a path leading to a pure Nash equilirium. We prove in the full version that Is-Stochastically-Convergent also is PSPACE-complete in the observable-actions model.

THEOREM 14. *Is-Stochastically-Convergent is PSPACE complete in the observable actions model.*

Acknowledgements

We thank Lance Fortnow, Aaron D. Jaggard, Jennifer Rexford, and Rebecca N. Wright for helpful discussions.

REFERENCES

ALON, N., MATIAS, Y., AND SZEGEDY, M. 1996. The space complexity of approximating the frequency moments. In *Proc. of STOC*.

ASPNES, J. AND RUPPERT, E. 2007. An introduction to population protocols. *Bulletin of the EATCS 93*, 98–117.

DAVIS, A. AND NOWICK, S. M. 1996. An introduction to asynchronous circuit design. Tech. Rep. UUCS-97-013, CS Dept., U. of Utah.

DOLEV, S. 2000. *Self stabilization*. MIT Press.

FABRIKANT, A., JAGGARD, A., AND SCHAPIRA, M. 2010. On the structure of weakly acyclic games. In *Proc. of SAGT*.

FABRIKANT, A. AND PAPADIMITRIOU, C. H. 2008. The complexity of game dynamics: BGP oscillations, sink equilibria, and beyond. In *Proc. of SODA*. 844–853.

FICH, F. AND RUPPERT, E. 2003. Hundreds of impossibility results for distributed computing. *Distributed Computing 16,* 2–3, 121–163.

GAO, L. AND REXFORD, J. 2002. Stable internet routing without global coordination. *IEEE/ACM Trans. Netw. 9,* 6, 681–692.

GODFREY, P. B., SCHAPIRA, M., ZOHAR, A., AND SHENKER, S. 2010. Incentive compatibility and dynamics of congestion control. In *Proc. SIGMETRICS*. 95–106.

GRIFFIN, T. G., SHEPHERD, F. B., AND WILFONG, G. 2002. The stable paths problem and interdomain routing. *IEEE/ACM Trans. Netw. 10,* 2, 232–243.

HART, S. 2005. Adaptive heuristics. *Econometrica 73*, 1401–1430.

HART, S. AND MANSOUR, Y. 2010. How long to equilibrium? the communication complexity of uncoupled equilibrium procedures. *GEB 69,* 1, 107–126.

JAGGARD, A. D., SCHAPIRA, M., AND WRIGHT, R. N. 2011. Distributed computing with adaptive heuristics. In *Proc. ICS*. 417–443.

LEHRER, E. AND LAGZIEL, D. 2012. No regret with delayed information. Tech. rep., Tel-Aviv Univ. Working paper, Google Electronic Markets and Auctions initiative.

LEVIN, H., SCHAPIRA, M., AND ZOHAR, A. 2008. Interdomain routing and games. In *Proc. of STOC*. 57–66.

LYNCH, N. 1989. A hundred impossibility proofs for distributed computing. In *Proc. ACM PODC*. 1–28.

MIRROKNI, V. S. AND SKOPALIK, A. 2009. On the complexity of Nash dynamics and sink equilibria. In *Proc. EC*. 1–10.

MONDERER, D. AND SHAPLEY, L. S. 1996. Potential games. *Games and Economic Behavior 14*, 124–143.

NISAN, N. 2002. The communication complexity of approximate set packing and covering. In *Proc. of ICALP*.

NISAN, N., SCHAPIRA, M., AND ZOHAR, A. 2008. Asynchronous best-reply dynamics. In *Proc. WINE '08*. 531–538.

OSBORNE, M. J. AND RUBINSTEIN, A. 1994. *A Course in Game Theory*. MIT Press.

ROSENTHAL, R. W. 1973. A class of games possessing pure-strategy Nash equilibria. *Int. J. Game Theory 2*, 65–67.

SAMI, R., SCHAPIRA, M., AND ZOHAR, A. 2009. Searching for stability in interdomain routing. In *Proc. of INFOCOM*.

SUCHARA, M., FABRIKANT, A., AND REXFORD, J. 2011. BGP safety with spurious updates. In *Proc. of INFOCOM*.

YOUNG, H. P. 1993. The evolution of conventions. *Econometrica 61,* 1, 57–84.