Theory Comput Syst
DOI 10.1007/s00224-013-9474-z

Weakly-Acyclic (Internet) Routing Games

Roee Engelberg · Michael Schapira

Abstract Weakly-acyclic games—a superclass of potential games—capture distributed environments where simple, globally-asynchronous interactions between strategic agents are guaranteed to converge to an equilibrium. We explore the class of routing games introduced in Fabrikant and Papadimitriou (The Complexity of Game Dynamics: BGP Oscillations, Sink Equilibria, and Beyond, pp. 844–853, 2008) and in Levin et al. (Interdomain Routing and Games, pp. 57–66, 2008), which models important aspects of routing on the Internet. We show that, in interesting contexts, such routing games are weakly acyclic and, moreover, that pure Nash equilibria in such games can be found in a computationally efficient manner.

Keywords Weakly-acyclic games · Routing games · Convergence to Nash equilibrium · Bestresponse dynamics

1 Introduction

1.1 Weakly-Acyclic Games

Convergence to a pure Nash equilibrium (PNE) is an important objective in a large variety of application domains—both computerized and economic. Ideally, this can be achieved via simple and natural dynamics, e.g., better-response or best-response dynamics. Under better-response dynamics, players start at some initial strategy profile and take turns selecting strategies. At each (discrete) time step, a single player selects a strategy that *increases* his utility (given the others' current strategies). Un-

R. Engelberg
Google Inc. and Computer Science Department, Technion, Haifa 32000, Israel
e-mail: roee@cs.technion.ac.il

M. Schapira (✉)
School of Computer Science and Engineering, Hebrew University of Jerusalem, Jerusalem, Israel
e-mail: schapiram@cs.huji.ac.il

Published online: 02 May 2013

der best-response dynamics, at every time step the "active" player chooses a strategy that *maximizes* his utility. Better-response and best-response dynamics are simple, low-cost behaviors to build into distributed systems, as evidenced by today's protocol for routing on the Internet [5, 15].

Convergence of better-/best-response dynamics to PNE is the subject of much research in game theory. Clearly, a *necessary* condition for better-/best-response dynamics to converge to a PNE regardless of the initial state is that, for every such state, there exist *some* better-/best-response improvement path to a PNE, i.e., a sequence of players' better-/best-response strategies which lead to a PNE.[1] Games for which this holds (e.g., potential games [20]) are called "*weakly acyclic*" [18, 24]. Weak acyclicity has also been shown to imply that simple dynamics (e.g., *randomized* better-/best-response dynamics, no-regret dynamics) are guaranteed to reach a PNE [16, 18, 24]. Thus, weak acyclicity captures distributed environments where a PNE can be reached via simple, globally-asynchronous interactions between strategic agents, regardless of the starting state of the system.

While the class of potential games—a subclass of weakly-acyclic games—is the subject of extensive research, relatively little attention has been given to the much broader class of weakly-acyclic games (see, e.g., [4, 16]). As a result, very few concrete examples of weakly-acyclic games that do not fall in the category of potential games are known. One famous result along these lines is that of Milchtaich [18]. Reference [18] studies Rosenthal's congestion games [22] and proves that, in interesting cases where the payoff functions (utilities) are player-specific, such games are weakly acyclic (but not necessarily potential games).

Our focus in this work is on another extensively studied environment: routing on the Internet. We show that weak acyclicity is important for analyzing such environments. Our work, alongside its implication for Internet routing, provides concrete examples of weakly acyclic games that lie beyond the space of potential games, as well as technical insights into the structure of such games.

1.2 (Internet) Routing Games

The Border Gateway Protocol (BGP) establishes routes between the smaller, independently administered, often competing networks that make up the Internet. Hence, BGP can be regarded as the glue that holds today's Internet together. Over the past decade there has been extensive research on the computational and strategic facets of routing with BGP. Recent advances along these lines were obtained via game-theoretic analyses (see, e.g., [5, 11, 15, 21]), which rely on the simple, yet important, observation that BGP can be regarded as best-response dynamics in a specific class of routing games [5, 15]. We now provide an intuitive exposition of the class of routing games in [5, 15]. We refer the reader to Sect. 2 for a formal presentation.

In the game-theoretic framework of [5, 15], the players are source nodes residing on a network graph, which aim to send traffic to a unique destination in the network. Each source node has a (private) ranking of all simple (loop-free) routes between

[1] Observe that this is equivalent to requiring that the game has no "non-trivial" sink equilibria [5, 10] under better-response dynamics (i.e., that it has no sink equilibrium of size greater than 1).

itself and the destination. We stress that, in practice, different source nodes can have very different, often conflicting, rankings of routes, reflecting, e.g., local business interests [9] (in particular, source nodes do not always prefer shorter routes to longer ones). Every source node's strategy space is the set of its neighboring nodes in the network; a choice of strategy represents a choice of a single neighbor to forward traffic to. Observe that every combination of source nodes' strategies thus captures how traffic is forwarded (hop-by-hop) towards the destination. A source node's utility from every such combination of strategies reflects how highly it ranks its induced route to the destination.

Fabrikant and Papadimitriou [5] and, independently, Levin et al. [15], observed that BGP can be regarded as best-response dynamics in this class of routing games and that PNEs in such games translate to the notion of stable routing states, which has been extensively studied in communication networks literature. These observations laid the foundations for recent results regarding the dynamics and incentive compatibility of routing on the Internet (see [11, 13, 21]).

1.3 Our Contributions: Weakly-Acyclic Routing Games

We present two interesting subclasses of routing games—(1) routing games with "misbehaving" players and (2) backup routing games—which capture important aspects of routing on the Internet. Routing games with misbehaving players intuitively capture scenarios where all but a few players are "well behaved", and the remaining players behave in an arbitrary manner. Such erratic behavior can, for instance, be the consequence of router configuration errors. Backup routing games model the common practice of backup routing with BGP [8].

We prove that games in both these classes are weakly acyclic. Our results thus establish that, in these two contexts, a PNE is guaranteed to exist and can be reached via simple, globally-asynchronous interactions between strategic agents regardless of the initial state of the system. Moreover, we prove that not only is a PNE reachable from every initial state via a better-response improvement path, but that this path is "short" (of polynomial length). Hence, in these subclasses of routing games, a PNE can be found in a computationally-efficient manner; simply start at an arbitrary initial state and follow the short better-response improvement path—whose construction we give explicitly—until a PNE is reached.

Routing Games with Misbehaving Players To illustrate this subclass of games, consider the scenario that all source nodes but a single source node m have a "shortest-path ranking" of routes, i.e., they always prefer shorter routes to longer routes. Unlike the other source nodes, m's ranking of routes need not necessarily be a shortest-path ranking and is not restricted in any way, e.g., m might even always prioritize longer routes over shorter routes. We aim to answer the following question: "Can m's erratic behavior render the network unstable?".

We prove a surprising positive result: every routing game of the above form (i.e., with a single "misbehaving" source node) is weakly acyclic even under best-response (i.e., from every initial state there exists a best-response improvement path to a PNE). Hence, in particular, routing games where each player has a shortest-path ranking are guaranteed to posses a PNE even in the presence of an arbitrary change in a single source node's behavior! We generalize this result to a broader class of routing policies. We point out that our work is one of few to explore the impact of "irrational" behavior in game-theoretic settings (see [2, 3, 12]).

Backup Routing Games In this subclass of routing games each edge in the network graph is either categorized as a "primary" edge or as a "backup" edge. A source node with multiple outgoing edges prefers forwarding traffic to neighboring nodes to which it is connected via primary edges over forwarding traffic to neighbors to which it is connected via backup edges. Such "backup relationships" are often established in practice to provide connectivity in the event of network failures via redundancy; the intent is that backup edges be used for carrying traffic only in case of failures in the primary edges [8]. We consider natural restrictions on source nodes' routing policies which capture this notion of "backup routing". We prove that the resulting routing games are weakly acyclic.

1.4 Related Work

Weak Acyclicity Weak acyclicity [18, 24] has been addressed in a handful of other game-theoretic contexts, including games with "strategic complementarities" [7, 14] (a supermodularity condition on lattice-structured strategy sets), and several kinds of succinct games [19]. Milchtaich [18] studied Rosenthal's congestion games [22] with player-specific payoff functions and showed that this is a subclass of weakly-acyclic games. Marden et al. [17] formulated the cooperative-control-theoretic consensus problem as a potential game (implying that it is weakly acyclic); they also defined and investigated a time-varying version of weak acyclicity. Weak acyclicity has been shown to imply that simple dynamics (e.g., randomized better-/best-response dynamics, no-regret dynamics) are guaranteed to reach a PNE in [16, 18, 24]. Fabrikant et al. [4] explore the connection between weakly acyclic games and structural properties of their subgames. Reference [23] showed that integer-splittable weighted congestion games with convex increasing unit-cost functions are weakly acyclic. Recently, [1] presented a classification of weakly acyclic games.

Internet Routing Games Fabrikant and Papadimitriou [5] and, independently, Levin et al. [15], observed that BGP can be regarded as best-response dynamics in a specific class of routing games and that PNEs in such games translate to the notion of stable routing states, which has been extensively studied in communication networks literature. These observations laid the foundations for recent results regarding the dynamics and incentive compatibility of routing on the Internet (see [11, 13, 21]).

1.5 Organization

We present the class of weakly-acyclic games and the class of weakly-acyclic under best-response games in Sect. 2, where we also present the class of routing games of [5, 15]. In Sect. 3, we illustrate the type of results we obtain via two simple families of weakly-acyclic routing games. We present our results for routing games with misbehaving players, and for backup-routing games, in Sects. 4 and 5, respectively. We conclude in Sect. 6.

2 Model

2.1 Weakly-Acyclic Games

We use standard game-theoretic notation. Consider a normal-form game with n players $1, \ldots, n$, where each player i has strategy space S_i and utility function u_i (which specifies player i's utility for every combination of players' strategies). Let $S = S_1 \times \cdots \times S_n$ and $S_{-i} = S_1 \times \cdots \times S_{i-1} \times S_{i+1} \times \cdots \times S_n$. For every $s_i \in S_i$ and $s_{-i} \in S_{-i}$, (s_i, s_{-i}) denotes the combination of players' strategies where player i's strategy is s_i and the other players' strategies are as in s_{-i}.

Definition 2.1 (Better-Response Strategies) We call a strategy $s_i^* \in S_i$ a "*better-response*" of player i to a strategy vector $s = (s_i, s_{-i}) \in S$ if $u_i(s_i^*, s_{-i}) > u_i(s_i, s_{-i})$.

Definition 2.2 (Best-Response Strategies) We call a strategy $s_i^* \in S_i$ a "*best response*" of player i to a combination of other players' strategies $s_{-i} \in S_{-i}$ if $s_i^* \in argmax_{s_i \in S_i} u_i(s_i, s_{-i})$.

Definition 2.3 (Pure Nash Equilibria) A strategy vector $s = (s_1, \ldots, s_n) \in S$ is a *pure Nash equilibrium (PNE)* if s_i is a best response to s_{-i} for every player i.

Definition 2.4 (Better- and Best-Response Improvement Paths) A *better-response (best-response) improvement path* in a game Γ is a sequence of strategy vectors $s^{(1)}, \ldots, s^{(k)} \in S$, each reachable from the previous via a better response (best response) of a single player.

We are now ready to present the class of weakly-acyclic games and the class of weakly-acyclic under best-response games.

Definition 2.5 (Weak Acyclicity and Weak Acyclicity Under Best Response) A game Γ is *weakly acyclic (weakly acyclic under best response)* if, from every $s \in S$ there exists a better-response (best-response) improvement path to a pure Nash equilibrium of Γ.

2.2 (Internet) Routing Games

In the class of routing games in [5, 15] the players are n source nodes $1, \ldots, n$ residing on a network $G = (V, E)$ who wish to send traffic to a *unique* destination node d. Let P_i be the set consisting of all simple (loop-free) routes from source node i to d in G and of the "empty route" $\perp$. Each source node i's strategy is its choice of an outgoing edge $e_i \in E(G)$ (intuitively, a neighboring node to forward traffic to), or the empty set $\emptyset$ (intuitively, not forwarding traffic). Observe that every combination of source nodes' strategies $s \in S$ thus specifies a (directed) subgraph G_s of G in which each source node has out-degree at most 1. Given a combination of nodes' strategies $s \in S$ we define i's *induced route* R_i^s to be i's unique simple route to d in G_s if such a route exists, and $\perp$ otherwise.

We now define source nodes' utility functions. Each source node i has a *routing policy* with two components: (1) a *ranking function* π_i that maps elements in P_i to the integers, such that $0 = \pi_i(\perp) < \pi_i(R)$ for all $R \in P_i \setminus \{\perp\}$; and (2) an *export policy* that, for each neighboring node $j \in V(G)$, specifies a set of routes $R_{ij} \subseteq P_i$ that i is willing to make available to j. This means that for a given strategy s, if $R_i^s \in R_{ij}$, node i is exporting its route R_i^s to j (and only this route), and otherwise, node i is not exporting any route to node j.

To simplify notation, when $\pi_i(R) < \pi_i(Q)$ ($\pi_i(R) \leq \pi_i(Q)$) for some routes $R, Q \in P_i$, we write $R <_i Q$ ($R \leq_i Q$). We say that a route $R \in P_i$ is "*permitted*" if each node on R is willing to export its (sub)route to its predecessor on R. Given a combination of nodes' strategies $s \in S$, i's utility is:

- If $R_i^s \neq \perp$ and R_i^s is permitted, $\pi(R_i^s)$.
- Else, if $s_i = \emptyset$, 0.
- Otherwise, -1.

Notice that each node prefers to use the empty strategy over using a different strategy which does not result a permitted route to d. Last but not least, if R is a route that ends at node j and Q is a route that starts from the node j, we denote by RQ the concatenation of the two routes. When R is a single edge (i, j) we may simply use the notation $(i, j)Q$.

3 Illustration: Simple Weakly-Acyclic Routing Games

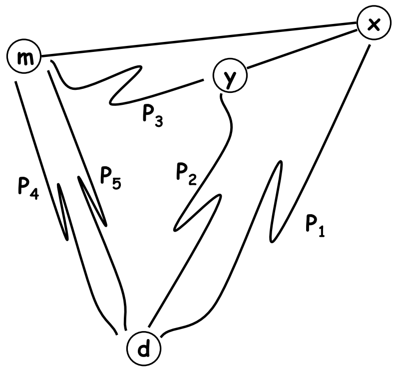
We now illustrate the kind of results we obtain via two simple families of weakly-acyclic games.

Shortest-Path Routing with Misbehaving Players Consider the scenario that all source nodes have shortest-path rankings (where shorter routes are always preferred to longer ones) and "export-all policies", i.e., each node i is willing to make all routes in P_i available to all neighboring nodes. We call games of this form "shortest-path routing games". We make the simple observation that shortest-path routing games are potential games (see formal definition of potential functions and potential games in [20]). Now, consider the case that there is a single misbehaving player, i.e., that the routing policy of a single source node in a shortest-path routing game is changed arbitrarily. We now present the following corollary of a more general result proved in Sect. 4.

Corollary 3.1 *Every shortest-path routing game with a single misbehaving player is weakly-acyclic under best response and, moreover, a PNE in such a game can be found in a computationally-efficient manner.*

To complement this result and better classify shortest-path routing games with a single misbehaving player, we show that, due to the introduction of the misbehaving player, they are no longer necessarily potential games.

Fig. 1 A shortest-path routing game with a single misbehaving player that is not a potential game



Theorem 3.1 *There exists a shortest-path routing game with a single misbehaving player which is not a potential game.*

Proof Consider the network G partially described by Fig. 1. We are interested in three nodes: x, y and m, where m is the misbehaving node. The paths P_1, P_2, P_3, P_4 and P_5 are all disjoint, and their lengths are 8, 6, 2, 2 and 10 respectively.

Hence, x prefers the path $(x,m)P_4$ (length 3) over P_1 (8) which is preferred over $(x,m)P_5$ (11).

Also, y prefers the path $(y,x)(x,m)P_4$ (length 4) over P_2 (6) which is preferred over $(y,x)P_1$ (9).

The misbehaving node m has the following preferences:

$$P_3P_2 <_m P_4 <_m P_5 <_m P_3(y,x)P_1.$$

Notice that m is misbehaving: it prefers longer routes such as P_5 over the shorter route P_4.

We now present a better-response improvement cycle. In this cycle, all nodes but x, y and m are fixed, and the paths P_1, P_2, P_3, P_4 and P_5 are all valid routes resulting from the fixed strategies of these nodes. Now, consider the following sequence of transitions:

x's strategy	y's strategy	m's strategy	Notes
P_1 (length 8)	x (9)	P_3	–
P_1 (8)	P_2 (6)	P_3	y changes to a shorter route
P_1 (8)	P_2 (6)	P_4	$P_3P_2 <_m P_4$
m (3)	P_2 (6)	P_4	x changes to a shorter route
m (3)	x (4)	P_4	y changes to a shorter route
m (11)	x (12)	P_5	$P_4 <_m P_5$
P_1 (8)	x (9)	P_5	x changes to a shorter route
P_1 (8)	x (9)	P_3	$P_5 <_m P_3(y,x)P_1$

Observe that every strategy profile is reachable from the strategy profile that comes before it via the better response of a single node in $\{x, y, m\}$, and that the first strategy and the last strategy in this sequence are identical. Hence, there exists a better-response improvement cycle and so this game is not a potential game. □

We note that the construction in the proof above uses a better response dynamic which is not a best response dynamic (m is not choosing its best response when choosing the route P_4). It is an open question whether it is possible to construct an example with a best response dynamic that does not converge.

Shortest-Path Backup Routing In this setting, every edge in the network graph is either "primary" or "backup". A source node always prefers a route through a neighbor to which it is connected via a primary edge ("primary route") over a route through a neighbor to which it is connected via a backup edge ("backup route"). When faced with a choice between two (or more) primary routes, or two (or more) backup routes, nodes always prioritize shorter routes. Every source node i has an export-all policy (i.e., i is willing to make all routes in P_i available to all neighboring nodes). We prove the following.

Theorem 3.2 *Every shortest-path backup routing game is a potential game and, moreover, a PNE in such a game can be found in a computationally-efficient manner.*

Proof Let $B = n^2$. For each node i, let $\ell(s,i)$ be the length of its route to d in the strategy profile s. Define $\phi(s,i)$ to be $\ell(s,i)$ if i's route is a primary route, and $B + \ell(s,i)$ if its route is a backup route ($\phi(s,i) = B^2$ if i has no route to d). We now show that $\phi(s) = \sum_i \phi(s,i)$ is a valid (ordinal) potential function.

Consider a strategy vector s, and assume that node i has a better response s_i^*. If s_i and s_i^* are both primary (respectively, backup), it must be the case s_i^* leads to a shorter path from i to d and hence $\ell((s_i^*, s_{-i}), i) < \ell(s,i)$ and $\phi((s_i^*, s_{-i}), i) < \phi(s,i)$. Also notice that for every node j, $\ell((s_i^*, s_{-i}), j) \le \ell(s,j)$ and $\phi((s_i^*, s_{-i}), j) \le \phi(s,j)$, which means that $\phi(s_i^*, s_{-i}) < \phi(s)$ in this case. If s_i is the empty strategy, then $\phi((s_i^*, s_{-i}), i) < B^2 = \phi(s,i)$, and a similar argument leads again to $\phi(s_i^*, s_{-i}) < \phi(s)$.

Lastly, if s_i is a backup edge and s_i^* is a primary edge, $\phi((s_i^*, s_{-i}), i) < n$ and $\phi(s,i) > B = n^2$. For every node $j \ne i$, $\ell((s_i^*, s_{-i}), j) \le \ell(s,j) + (n-3)$ and $\phi((s_i^*, s_{-i}), j) \le \phi(s,j) + (n-3)$. Summing up we get,

$$\phi\left(s_i^*, s_{-i}\right) < n - n^2 + (n-1) \cdot (n-3) + \phi(s) < \phi(s).$$

We conclude that $\phi(s)$ is a valid potential function.

Since the potential function is non negative, its maximum value is polynomial in n and in each step it decreases by at least 1—any improvement path has a polynomial length. It follows that a PNE in such a game can be found in a computationally-efficient manner. □

In Sect. 5 we examine a more complex class of backup routing games and show that games in that class are guaranteed to be weakly acyclic yet are not necessarily potential games.

4 Routing Games with Misbehaving Players

We now present our results for the class of routing games with misbehaving players.

4.1 Routing Policies

[6] introduces the notions of policy consistency and of consistent export, which generalize natural classes of routing policies, e.g., shortest-path routing and next-hop routing. We now present these two concepts.

Policy-Consistent Ranking Two well-studied classes of ranking functions are shortest-path rankings and next-hop rankings. Shortest-path rankings always prioritize shorter routes. Next-hop rankings, in contrast, rank routes based solely on the identity of the “next-hop”—the immediate neighbor—en route to the destination, i.e., a next-hop ranking assigns the same preference to all routes that share the same next-hop node. [6] generalizes these two classes of rankings as follows.

Definition 4.1 (Policy Consistency) [6] Let i and j be two adjacent source nodes in G. We say that i is *policy consistent* with j iff for every two routes $Q, R \in P_j$ such that $i \notin Q, R$, if $R <_j Q$, then $(i, j)R \leq_i (i, j)Q$. We say that *policy consistency* holds if each source node is policy consistent with each of its neighboring source nodes.

Observe that in the scenario that all source nodes have shortest-path rankings, and also in the scenario that all nodes have next-hop rankings, policy consistency indeed holds. (See [6] for more details.)

Consistent Export The simplest export policy is the export-all policy, where a source node i is willing to make all routes in P_i available to all neighboring nodes. Reference [6] presents the following generalization of export-all.

Definition 4.2 (Consistent Export) [6] Let i and j be two adjacent nodes in G. We say that i *consistently exports* with respect to j iff there is some route $R \in P_i$ such that $R_{ij} = \{Q \mid Q \in P_i \text{ and } R \leq_i Q\}$. We say that a node i *consistently exports* if it consistently exports with respect to each neighboring node j. We say that *consistent export* holds if all nodes consistently export.

Observe that when all source nodes have all-export policies then consistent export indeed trivially holds.

4.2 Positive Result

Consider games for which policy consistency and consistent export hold. These games include, among others, routing games with shortest-path rankings and export-all policies, as well as routing games with next-hop rankings and export-all policies, and can easily be shown to be potential games. We turn our attention to the scenario that there exists a single misbehaving player, i.e., that the routing policy of a single player can be changed in an arbitrary manner. We prove the following surprising positive result.

Theorem 4.1 *If policy consistency and consistent export hold for a routing game except for a single misbehaving player, then the game is weakly acyclic under best response. Moreover, a PNE in such a game (with a single misbehaving player) can be found in a computationally-efficient manner.*

Proof Our proof is constructive. Let G be a network for which policy consistency and consistent export hold with the exception of m, the (single) misbehaving node.

Given a strategy s, we introduce the following terminology:

- *Blue nodes*: nodes that appear on m's route to d.
- *Red nodes*: nodes whose routes to d go through m.
- *Black nodes*: nodes that are disconnected from d (i.e., have no route to d).
- *White nodes*: nodes that do not fall within the previous three categories.

We present a best response improvement path from any given initial strategy s, which we construct via what we call the "*stabilization process*".

The stabilization process:

- *Part I*: Repeat the following two steps until the strategy vector is such that there are no black and red nodes whose best response is to become white (but at least once).
  - *Step 1*: Repeatedly activate (one by one, and in arbitrary order) all nodes (but m) that wish to select new routes[2] that do not go through m (and allow them to do so), until no such node exists.
  - *Step 2*: Activate m once.
- *Part II*: Repeatedly activate (one by one, and in arbitrary order) all nodes but m that wish to select new routes (and allow them to do so), until no such node exists.[3]

We refer to each time that the two steps of Part I of the stabilization process are executed as an iteration.

Proposition 4.1 *Each individual iteration in Part I is guaranteed to terminate.*

Proof Consider a single iteration of Part I. During Step 1, the number of red nodes is only decreasing. Hence, it gets to a minimum at some point in time and stays there until the end of Step 1. Notice that these nodes do not move during the iteration. We use the social welfare of the nodes, disregarding m and the set of nodes which stay red (and do not move), as a potential function. The social welfare is the sum of the

[2] Henceforth, we shall sometimes say that a node selects a route when we mean that the node actually selects an outgoing edge (in its strategy set); the selected edge, followed by the induced route to d of the neighboring node to which that edge points determines what we call the node's selected route.

[3] Note that at the end of Part I, some node might want to select a path that goes through m; This was not allowed in Part I, but in Part II we allow it.

utilities of the corresponding nodes. Notice that since m is not moving, and since policy consistency holds for all other nodes, each best response of a node i can only improve the utility of the other nodes. This means that the set of nodes that might move during Step 1, reaches an equilibrium after a finite number of best response moves. Step 2 is a single best response and the proposition follows. □

By definition of Part I of the stabilization process:

Observation 4.1 *A node that is not red at some point in Part I of the stabilization process, will not become red later in Part I of the stabilization process.*

Proposition 4.2 *Part I of the stabilization process terminates.*

Proof Consider the set of red nodes. By Observation 4.1, no nodes turn red in Part I, and once red nodes becomes non-red they never become red again. Therefore, after sufficiently many iterations the set of red nodes remains fixed.

Notice that only red nodes can turn black (due to m's export policy or decision to choose the empty strategy). So after sufficiently many iterations the set of black nodes remains fixed.

Once the set of red and black nodes is fixed, the termination condition of Part I is satisfied. □

To prove that Part II terminates at an equilibrium we have the following lemma.

Lemma 4.1 *Let R be m's route at the end of Part I of the stabilization process, and let Γ be the set of all routes to d that were available to m at the end of Part I of the stabilization process. Then,*

- *m's route throughout Part II and at the end of Part II of the stabilization process remains R.*
- *The set of routes to d that are available to m at the end of Part II of the stabilization process must be a subset of Γ.*

We observe that Lemma 4.1 concludes the proof of weak acyclicity. First, the termination is guaranteed:

Proposition 4.3 *Part II of the stabilization process terminates.*

Proof Since m is not allowed to move during Part II, and since the first part of Lemma 4.1 assures us that its route is fixed throughout Part II, we may think of m as a fixed node with a fixed route that may or may not be exported to each of its neighbors. This can be modeled by a node that is not misbehaving. Hence, in Part II of the stabilization process the network is effectively equivalent to a network for which policy consistency and consistent export hold. Such a network is known to be a potential game, so Part II of the stabilization process is guaranteed to terminate. □

Second, observe that at the end of Part II of the stabilization process every node but, possibly, m, is playing its best response. Hence, to prove that the routing state reached at the end of Part II is a PNE it suffices to show that m is also playing its best response. Consider the edge e chosen by m to result the route R at the end of Part I of the stabilization process. It must be that e is m's best response at that point in time. By Lemma 4.1, R will remain m's route at the end of Part II of the stabilization process. Moreover, e must still be m's best response, since, by Lemma 4.1, no new routes were made available to m since the end of Part I of the stabilization process. This concludes the proof of weak acyclicity.

Lemma 4.1 now follows from the following proposition:

Proposition 4.4 *The following holds*:

1. *If a node is blue at the end of Part I*, *it will have the same route throughout Part II.*
2. *If a node is red at the end of Part I or becomes red during Part II*, *it shall remain red henceforth.*
3. *If a node is black at the end of Part I*, *it shall be either black or red at the end of Part II.*
4. *If a node is white at the end of Part I*, *it will either have the same route throughout Part II* (*and thus remain white*), *or be red at the end of Part II.*

The first bullet in the statement of Proposition 4.4 implies the first half of Lemma 4.1; if all blue nodes have the same route at the end of Part I and at the end of Part II, then m's route at the end of Part I is the same route it has at the end of Part II. The other bullets in the statement of Proposition 4.4 imply the second half of Lemma 4.1; m's route to d cannot go through black nodes (that are disconnected from d), or red nodes (whose routes go through m). Hence, if white nodes either have the exact same route at the end of Part I and at the end of Part II, or become red, and no other node becomes white, then m's set of available routes at the end of Part II is a subset of its set of available routes at the end of Part I. To conclude the proof, we now prove Proposition 4.4.

Proof of Proposition 4.4 For point of contradiction, suppose that one (or more) of the bullets in the statement of Proposition 4.4 is false. Let i be the first node to violate the statement of one of the bullets when selecting a route in Part II. We handle the four cases:

- *Case I*: *i is a blue node whose route changes*. Consider the moment in time in which i's route changed. Let j be i's next hop node on the new route.
- *Case II*: *i is a red node that becomes non-red*. Consider the moment in time in which i becomes non-red. Let j be i's next hop on its new (non-red) route.
- *Case III*: *i turns from black to blue/white*. Consider the moment in time in which i turned to blue/white. Let j be i's next hop on its new (non-black) route.
- *Case IV*: *i is a white node that changes its route and becomes non-red*. Consider the moment in time in which i changed its route and became non-red. Let j be i's next hop on its new (non-red) route.

For all the cases, observe that j cannot be black or red. Hence, j must be either blue or white. If so, j had the exact same route at the end of Part I (or we get a contradiction to the choice of i). i could therefore choose to route through j at an earlier time and did not do so—a contradiction. □

Computational Complexity of Finding a PNE We are left with showing that the length of the improvement path constructed in the stabilization process is polynomial. Notice that the number of iterations in Part I is at most the number of nodes, as in every iteration there must be a red or a black (former-red) node that turns white, only red nodes turn black and no nodes become red. To complete the analysis we show that Step 1 of Part I (and Part II) can be made to run in a polynomial number of steps.

Consider the following variant of Step 1 (and Part II). We activate nodes in *rounds*, where in each round we increase the length of the routes we permit nodes to select. That is, in round i we let all nodes that want to select a route with length at most i, do so. We do so with one exception—we allow red and black nodes to become white at all times and give precedence to such transitions. Whenever such an event (red/black node becomes white) occurs, we restart the rounds (go back to length 1). Observe that such a "restart" can happen at most $|V|$ times.

We have already shown that Step 1 (and Part II) terminates regardless of the activation order. Let s be the resulting strategy profile. The graph G_s is a directed tree, rooted in d (apart from black nodes). The crucial observation is that nodes at distance i from d in G_s will have their route available and fixed by the i'th round, and from that point on, they will not move (nodes only improve their route). Also, any node that will, at some point, want to select a route of length i can and will do so at the i'th round. So, in the i'th round, each node has at most $|V|-1$ possible routes to choose from of length i, and the set of these routes can only get smaller as the round goes on. That means that a round has at most $|V|^2$ steps. The number of restarts possible is at most $|V|$, and in between any two restarts there can be up to $|V|$ rounds. We get at most $|V|^2$ rounds. This completes the analysis to give polynomial number of steps. □

Can this result be extended to more than a single misbehaving player? Simple examples show that the answer to this question is, in general, No. We believe, however, that under certain reasonable conditions (e.g., that the number of misbehaving nodes does not exceed a certain threshold, and that the misbehaving nodes are not "too concentrated" in a single part of the network) our result can be made to hold more generally. We leave this as an interesting direction for future research.

5 Backup Routing Games

In Sect. 3, we considered the following simple setting. Every edge in the network graph is either categorized as a "primary" edge or as a "backup" edge. A source node with multiple outgoing edges ranks routes in which it is connected to the next-hop node via a primary edge ("primary routes") more highly than routes in which it is connected to the next-hop node via a backup edge ("backup routes"). When faced with a choice between multiple routes in the same category (primary/backup) a source node always prioritizes shorter routes over longer routes. In addition, every source node has an export-all policy. We have shown that games that fall within this subclass of routing games are weakly-acyclic (and, in fact, even potential games). Next, we present a more realistic model, inspired by today's commercial Internet.

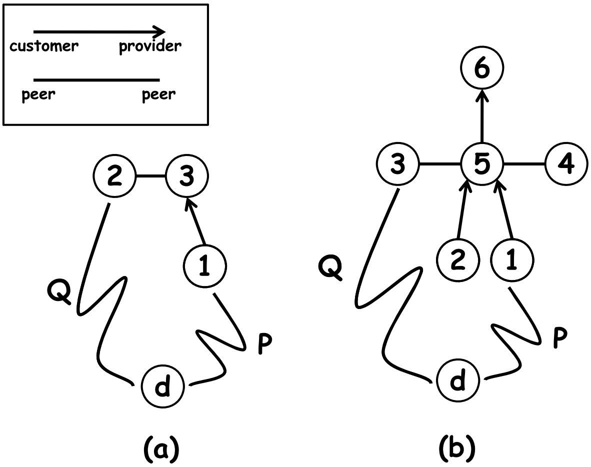


Fig. 2 Commercial backup routing. No assumption is made of the length of the paths P and Q

5.1 Commercial Backup-Routing Games

As before, each edge in the network graph is either "primary" or "backup". In addition, neighboring nodes in the network graph have one of two business relationships: either one node is a *customer* of the other (which is its *provider*) or the two nodes are *peers*. We make the standard assumption that no node is an indirect customer of itself, i.e., that there are no customer-provider cycles in this business hierarchy [9]. We now present constraints on source nodes' ranking functions and export policies that are naturally induced by this business hierarchy and extend the famous economic Gao-Rexford constraints [9] to handle backup routing. See [9] for a detailed explanation of this economic framework.

Ranking A source node with multiple outgoing edges ranks primary routes more highly than backup routes. When faced with a choice between multiple routes in the same category (primary/backup), a source node always prioritizes (revenue-generating) routes in which its next-hop is its customer ("customer routes") over routes in which its next-hop is its peer/provider ("peer/provider routes"). Consider the network in Fig. 2(a). In the event that node 3 has a primary edge to its peer, node 2, and a backup edge to its customer, node 1, node 3 should prefer routes through 2 over routes through 1. However, if 3's edges to nodes 1 and 2 are both primary or both backup, node 3 should prefer routes through 1 over routes through 2.

Export A source node is willing to export all routes through it to its customers; it is only willing to export (all of) its customer routes to neighbors that are its peers and providers. Formally, $R_{ij} = P_i$ when j is a customer of i. If j is a peer or a provider of i, R_{ij} is the set of customer routes of i (that is, all routes that leave i to a customer of i). Intuitively, this captures a source node's willingness to carry transit traffic for its customers, but not for its peers and providers (by whom it is not paid). Consider the network in Fig. 2(b). Node 5 should announce routes through node 1, its customer, to all neighboring nodes. However, node 5 should only announce routes through node 3, its peer, to its customers (nodes 1 and 2), and not to its other peer (node 4) and provider (node 6).

We call routing games where each node has a ranking function and export policy as above "commercial backup-routing games". Note that every permitted valid route from node i to d has edges that are ordered according to their type: first, edges from customer to provider (if any), then a single peer edge (if at all), and lastly, edges from provider to customer (if any).

5.2 Positive Result

We prove the following positive result for the class of commercial backup-routing games.

Theorem 5.1 *Every commercial backup-routing game is weakly acyclic and, moreover, a PNE in such a game can be found in a computationally-efficient manner.*

Proof To prove the theorem we construct a better-response improvement path from any given strategy profile to a PNE. The construction relies on an iterative process which is designed to gradually ensure that nodes do not get disconnected. Prioritizing primary edges is at the heart of the gradual process.

We use the following terminology throughout the proof.

Coloring nodes Let $s \in S$ and let G_s be the matching subgraph of G.

– *Black nodes in s* are nodes that do not have a permitted route to d in G_s.
– *White nodes in s* are nodes that have permitted primary routes to d in G_s.
– *Gray nodes in s* are nodes that have permitted backup routes to d in G_s.

We say that a strategy vector s is *clean* if every black node is playing its empty strategy. The improvement path we construct will move from one clean strategy to another by having a single node choose its best response and then letting all the black nodes that are not playing their empty strategy to do so (note that by our definition this is a better response for all such nodes). From the definition of a clean strategy, in every clean strategy s, each black node has no incoming edges selected in s. From this point and on, we ignore non-clean intermediate strategies.

Observation 5.1 (Possible Color Changes) *Let $s, s' \in S$ be two clean strategy vectors where s' is reachable from s via a single best response of node i (and then any better response of a black node needed to guarantee that s' is clean), then the possible color changes of nodes between s and s' are:*

– *Node i's color can (1) change from gray to white (if i has a backup route and selects a primary route); and (2) change from black to gray/white (if i had no route to d).*
– *Other nodes' colors can only change from white/gray to black. This occurs if i has a backup customer route and selects a primary peer/provider route, thus making its new route non-exportable to peers and providers (under commercial export policies).*

The Stabilization Phase The basic building block in our construction is the following procedure called the "*stabilization phase*". We start with a given strategy profile (assume it is clean), and activate nodes one by one, letting nodes that want to move select their best response, while not allowing black nodes to turn gray.[4] We give precedence to certain nodes as follows:

– First, let nodes that want to move to a primary customer route select new routes.
– Only if no node wishes to move to a primary customer route, allow a node that wants to move to a primary peer/provider route select a new route.
– Only allow a gray node that wants to move to (yet another) backup route to move if no nodes that belong to the previous categories wish to select new routes.

Observation 5.1 and the definition of the stabilization phase imply the following.

Observation 5.2 (Non-Gray Nodes Remain Non-Gray Throughout the Stabilization Phase) *If a node i is not gray at some point in the stabilization phase it shall not change its color to gray throughout the stabilization phase.*

Proposition 5.1 (The Stabilization Phase Terminates) *The stabilization phase terminates regardless of the initial strategy profile.*

Proof Here we use the following important result by Gao and Rexford [9].

Theorem 5.2 [9] *If all the edges are of one type* (*e.g.*, *all primary edges*), *then the routing game is a potential game.*

Observations 5.1 and 5.2, and the definition of the stabilization phase, imply that, from some moment in the stabilization phase onwards, each node's selected strategies are all primary edges or all backup edges. The game then is equivalent to a game from which all other strategies are removed. Observe that, in this new game, the distinction between primary and backup edges is meaningless (as nodes are never faced with a choice between a primary route and a backup route). Hence, according to Theorem 5.2 it is a potential game where every best response dynamic converges. The proposition follows. □

The Iterative Stabilization Process To construct a best-response improvement path that reaches a PNE we start at with any strategy profile s and execute the following "*iterative stabilization process*":

Repeat the following steps until a PNE is reached.

– Run the stabilization phase.
– Let V be the set of black nodes that can change colors to gray via best-response. If $V \neq \emptyset$ choose a black node $i \in V$ and change i's strategy to be its best-response.

What are problematic structures? A stabilization phase will reach a PNE if at the end of the phase, no more black nodes wish to move. We would like to avoid cases in

[4] After each such activation, we make sure we get to a clean strategy, as described above.

which nodes turn black during the stabilization phase. Hence, we explore the reasons that may cause a non-black node to become disconnected from d (thus turning black). According to Observation 5.1, the answer to this question necessitates the study of the circumstances that can cause a gray node with a customer route to move to a primary peer/provider route (and thus turn white). We refer to such a move as a *bad move*. Our goal is to try to prevent the repetition of bad moves. This shall be done by studying *problematic structures* that may appear in G_s. We first characterize the problematic structures, and then show that the iterative stabilization process eliminates such problematic structures while not adding new ones. Finally, we will prove the connection between problematic structures and bad moves and conclude the proof.

We shall require the following definitions:

Definition 5.1 (Primary Components) We say that two source nodes x and y are in the same primary component if there is a path in G that leads from x to y that does not violate the export constraint (i.e., in which no node goes through its customer's peer/provider route), such that all edges on the path are primary edges.

Remark 5.1 (Symmetry) Observe that if there is an exportable path (one that does not violate the export constraint) that leads from x to y in G then the same path reversed is also exportable and leads from y to x. Hence, the order of x and y is insignificant.

Definition 5.2 (Problematic Structures) Let s be a strategy profile. We refer to the two following events as "problematic structures" (in s):

- *Type I*: Let x and y be two nodes in the same primary component. Then, x and y are said to form a problematic structure of type I if both x and y have a backup customer route to d in s.
- *Type II*: Let x be a node in the same primary component as d. Then, x is said to form a problematic structure of type II if x has a backup customer route to d in s.

No new problematic structures are formed. In what follows we prove that the iterative stabilization process is such that no new problematic structures are formed during its execution. To set the ground for the proof, the following lemma states a strong property of the iterative stabilization process. We use the notation s_t to be the tth strategy vector.

Lemma 5.1 *Suppose x is a node in the same primary component with a node y that has a customer route* (*primary or backup*) *in G_{s_t}, and at the* $(t+1)$*th step a backup edge is chosen as a best response by some node. Then*,

- *If x has a customer path to y in G—then x has a primary customer route to d in G_{s_t}.*
- *If x has a peer/provider path to y in G—then x has a primary route to d or a backup customer route in G_{s_t}.*

Proof The proof is by induction on the length of the primary path between x and y. For the base case, there is a primary edge between x and y. Hence,

– *If x has a customer path to y*—then x is a provider of y and since y has a customer route to d, this route cannot pass through x. Moreover, y exports its route to x, hence x can choose a primary customer route and must have one in G_{s_t}.
– *If x has a peer/provider path to y*—then x is a peer or a customer of y; If in G_{s_t} y's customer route to d doesn't pass through x, y exports its route to x, and x can choose a primary route and must have one in G_{s_t}. If in G_{s_t} y's customer route to d passes through x, then since it is a customer route, x itself must have a customer route.

Now assume that the lemma holds for primary paths of length ℓ. Let x and y be nodes as described in the lemma, and specifically, assume there is a primary path of length $\ell + 1$ between them. Let w be the first node after x on that path. Notice that the lemma holds for w. Hence,

– *If x is a provider of w*—w must have a customer path to y. By the induction hypothesis, w has a primary customer route to d in G_{s_t}. This route cannot pass through x. Moreover, w exports its route to x, hence x can choose a primary customer route and must have one in G_{s_t}.
– *If x is a peer of w*—w must have a customer path to y. By the induction hypothesis, w has a primary customer route to d in G_{s_t}. If this route passes through x, then since it is a customer route, x itself must have a customer route in G_{s_t}. If this route doesn't pass through x, w exports its route to x, and x can choose a primary route and must have one in G_{s_t}.
– *If x is a customer of w*—By the induction hypothesis, w has a primary route to d or a backup customer route in G_{s_t}. If this route passes through x, then since x is a customer of w, x itself must have a customer route in G_{s_t}. If this route doesn't pass through x, w exports its route to x, and x can choose a primary route and must have one in G_{s_t}.

And the lemma follows. □

Proposition 5.2 (Problematic Structures of Type I Are Not Formed) *Let x and y be two nodes in the same primary component. If at some point in the stabilization phase x and y do not form a problematic structure of type I, they will not form such a problematic structure later in the stabilization process.*

Proof Assume to the contrary that x and y form a problematic structure of type I which is created at the tth step, and assume without loss of generality that x is the node that chooses a backup edge as its best response at the tth step. By the definition of a problematic structure of type I, at the tth step node y had a (backup) customer route. By Lemma 5.1, x has a primary route or a backup customer route just before the tth step. We get a contradiction: if x had a primary route just before the tth step, it would not choose a backup route at the tth step as a best response; if x had a backup customer route just before the tth step we get a contradiction to the choice of t as the step that the problematic structure is formed. □

Proposition 5.3 (Problematic Structures of Type II Are Not Formed) *Let x be a node that is in the same primary component as d. If, at some point x does not form* *a problematic structure of type II, it will not form such a structure at a later stage in the iterative stabilization process.*

Proof Assume to the contrary that x forms a problematic structure of type II when it chooses a backup edge as its best response at the tth step. Then, at the tth step node d has a (trivial) customer route to itself and by Lemma 5.1 x has a primary route or a backup customer route just before the tth step. We get a contradiction. □

Concluding the Proof Recall, that we have defined "bad moves" as cases in which nodes switch from backup customer routes to primary peer/provider routes, thus possibly disconnecting nodes that are connected to d through them (by making their new routes non-exportable). In Proposition 5.4 below, we prove that bad moves are linked to the existence of problematic structures. Specifically, any bad move in the stabilization phase can only occur if a problematic structure exists, and after every such bad move a problematic structure ceases to exist. So, by iteratively applying the stabilization phase, we remove problematic structures, and add no new such structures along the way. Eventually, no bad moves will take place. Once this happens, no non-black (connected) node will ever turn black (disconnected) again. From that point, the number of iterations is bounded and hence, the iterative stabilization process terminates. Consider the last iteration. At the end of that iteration there are no black nodes (that can be connected to d). Hence, in the last stabilization phase the constraint that nodes do not change from black to gray is redundant. The stabilization phase must therefore terminate at a PNE (using the same arguments as in the proof of Proposition 5.1).

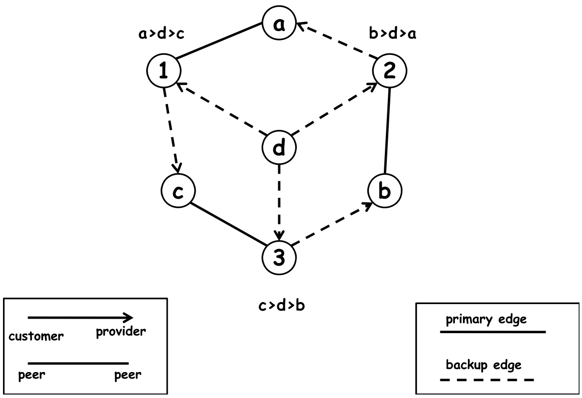
Proposition 5.4 (Problematic Structures and Bad Moves) *Consider a bad move where a gray node x moves from a backup customer route to a primary peer/provider route. Then, x is part of a problematic structure which is eliminated by the bad move.*

Proof If x is in the same primary component as d, it forms a problematic structure of type II which stops existing once x turns white.

If x is not in the same primary component as d, its newly selected route to d must go through a backup edge (by definition of primary components). Let e be the closest backup edge to x on its new route to d. Observe that if $e = (u, v)$ where v is u's customer, then u must form (along with x) a problematic structure of type I, that existed prior to the bad move. This structure is eliminated by the bad move as x no longer uses a backup edge. We shall now show that it must be the case that the backup edge e is indeed a customer edge (that is, from a provider to a customer), thus concluding the proof.

Let y be x's peer/provider through which x decides to send traffic in the bad move. Let z be y's next hop on x's route through y. If z is y's customer, since (x, y) is a primary edge, the backup edge has to be a customer edge (and so we are done). Consider the case that z is not y's customer. It follows that y is x's provider and that y prefers x over z. However, this means that y wants to move from z to x (or even a better response) at the same time that x wants to move from its backup customer route to y. Hence, by the definition of the stabilization phase we must give precedence to y, and so the bad move would not have even taken place via z in that case–a contradiction. □

Fig. 3 A commercial backup routing game that is not a potential game



Computational Complexity of Finding a PNE We now analyze the length of the improvement path that is constructed in the stabilization process. Notice that the number of possible problematic structures is polynomial, and so the number of possible bad moves is polynomial as well. Thus, every node can turn black only polynomially many times. This, in turn, implies that the number of stabilization phases is polynomial as well. To complete the analysis we note that each stabilization phase is equivalent to convergence in the Gao-Rexford framework [9], which entails at most a polynomial number of steps. □

Lastly, we show that commercial backup-routing games are, in fact, not contained in the class of potential games.

Theorem 5.3 *There exists a commercial backup-routing game which is not a potential game.*

Proof Consider the network in Fig. 3. There are 6 source nodes $1, 2, 3, a, b, c$ and a unique destination node d. The business relationships between nodes, and the classification of edges into primary edges and backup edges are described in the figure. Each of the source nodes 1, 2, and 3 has a next-hop ranking, and its preferences over next-hops are as in the figure (e.g., node 1 prefers all routes through its peer a over the direct route to its customer d, over all routes through its provider c). Each of the source nodes a, b, and c prefers routes through its peer (to which it is connected via a primary edge) over routes through its customer (to which it is connected via a backup edge). Observe that these rankings are indeed backup/primary commercial rankings (each node prefers primary routes over backup routes and, within each category of routes, prefers customer routes to peer/provider routes). Each source node has a commercial export-all policy, that is, it exports all routes to its customers and all customer routes to its peers/providers.

Observe that this routing game possesses (multiple) PNE, e.g., the routing state in which nodes 1 and 2 forward traffic directly to d, a and c forward traffic to 1, b to 2 and 3 forwards traffic to c. We now show that the game is not a potential game by presenting a better-response improvement cycle. Consider the case that each of the source nodes a, b, and c's strategy is fixed to be the outgoing link to its customer (e.g., c sends traffic to 1). Now, consider the following sequence of transitions between 3-tuples of source nodes $1, 2$, and 3's strategies (listed in that order): $((1d), (2d), (3c)) \to ((1a), (2d), (3c)) \to ((1a), (2d), (3d)) \to ((1a), (2b), (3d)) \to ((1d), (2b), (3d)) \to ((1d), (2b), (3c)) \to ((1d), (2d), (3c))$. Observe that every 3-tuple of strategies is reachable from the 3-tuple that comes before it via the best-response of a single node in $\{1, 2, 3\}$, and that the first 3-tuple and last 3-tuple in this sequence are identical. Hence, there exists a best-response improvement cycle and so this game is not a potential game. □

6 Conclusion and Open Questions

We have explored the class of weakly-acyclic games and presented two interesting families of routing games which fall within this category of games: (1) routing games with misbehaving players; and (2) backup routing games. Weak acyclicity of a game captures the possibility of reaching pure Nash equilibria via simple, local, globally-asynchronous interactions between strategic agents, independently of the starting state, and thus weak acyclicity embodies a realistic notion of global system stability in many distributed systems. Yet, while the class of potential games—a subclass of weakly-acyclic games—has been extensively studied, weakly-acyclic games remain largely unexplored. An important direction for future research is identifying broad, applicable and computationally-tractable conditions for weak acyclicity. We believe that weak-acyclicity arguments can prove to be useful in analyzing system convergence in other distributed economic and computational environments of interest (beyond BGP routing).

References

1. Apt, K.R., Simon, S.: A classification of weakly acyclic games. In: Serna, M. (ed.) SAGT. Lecture Notes in Computer Science, vol. 7615, pp. 1–12. Springer, Berlin (2012)
2. Babaioff, M., Kleinberg, R., Papadimitriou, C.H.: Congestion games with malicious players. In: MacKie-Mason, J.K., Parkes, D.C., Resnick, P. (eds.) ACM Conference on Electronic Commerce, pp. 103–112. ACM, New York (2007)
3. Eliaz, K.: Fault tolerant implementation. Rev. Econ. Stud. **69**, 589–610 (2002)
4. Fabrikant, A., Jaggard, A.D., Schapira, M.: On the structure of weakly acyclic games. In: Kontogiannis, S.C., Koutsoupias, E., Spirakis, P.G. (eds.) SAGT. Lecture Notes in Computer Science, vol. 6386, pp. 126–137. Springer, Berlin (2010)
5. Fabrikant, A., Papadimitriou, C.H.: The complexity of game dynamics: BGP oscillations, sink equilibria, and beyond. In: Teng, S.-H. (ed.) SODA, pp. 844–853. SIAM, Philadelphia (2008)
6. Feigenbaum, J., Ramachandran, V., Schapira, M.: Incentive-compatible interdomain routing. In: Feigenbaum, J., Chuang, J.C.-I., Pennock, D.M. (eds.) ACM Conference on Electronic Commerce, pp. 130–139. ACM, New York (2006)
7. Friedman, J.W., Mezzetti, C.: Learning in games by random sampling. J. Econ. Theory **98**, 55–84 (2001)
8. Gao, L., Griffin, T., Rexford, J.: Inherently safe backup routing with BGP. In: INFOCOM, vol. 1, pp. 547–556. IEEE, New York (2001)
9. Gao, L., Rexford, J.: Stable internet routing without global coordination. IEEE/ACM Trans. Netw. **9**(6), 681–692 (2001)

10. Goemans, M., Mirrokni, V., Vetta, A.: Sink equilibria and convergence. In: FOCS, pp. 142–151. IEEE Computer Society, Los Alamitos (2005)
11. Goldberg, S., Halevi, S., Jaggard, A.D., Ramachandran, V., Wright, R.N.: Rationality and traffic attraction: Incentives for honest path announcements in BGP. In: Bahl, V., Wetherall, D., Savage, S., Stoica, I. (eds.) SIGCOMM, pp. 267–278. ACM, New York (2008)
12. Gradwohl, R.: Fault tolerance in distributed mechanism design. In: Papadimitriou, C.H., Zhang, S. (eds.) WINE. Lecture Notes in Computer Science, vol. 5385, pp. 539–547. Springer, Berlin (2008)
13. Jaggard, A.D., Schapira, M., Wright, R.N.: Distributed computing with adaptive heuristics. In: ICS, pp. 417–443. Tsinghua University Press, Tsinghua (2011)
14. Kukushkin, N.S., Takahashi, S., Yamamori, T.: Improvement dynamics in games with strategic complementarities. Int. J. Game Theory **33**(2), 229–238 (2005)
15. Levin, H., Schapira, M., Zohar, A.: Interdomain routing and games. In: Dwork, C. (ed.) STOC, pp. 57–66. ACM, New York (2008)
16. Marden, J.R., Young, H.P., Arslan, G., Shamma, J.S.: Payoff-based dynamics in multi-player weakly acyclic games. SIAM J. Control Optim. **48**, 373–396 (2009)
17. Marden, J.R., Arslan, G., Shamma, J.S.: Connections between cooperative control and potential games illustrated on the consensus problem. In: Proceedings of the European Control Conference, July (2007)
18. Milchtaich, I.: Congestion games with player-specific payoff functions. Games Econ. Behav. **13**, 111–124 (1996)
19. Mirrokni, V.S., Skopalik, A.: On the complexity of Nash dynamics and sink equilibria. In: ACM Conference on Electronic Commerce, pp. 1–10 (2009)
20. Monderer, D., Shapley, L.S.: Potential games. Games Econ. Behav. **14**(1), 124–143 (1996)
21. Nisan, N., Schapira, M., Valiant, G., Zohar, A.: Best-response mechanisms. In: ICS, pp. 155–165. Tsinghua University Press, Tsinghua (2011)
22. Rosenthal, R.W.: A class of games possessing pure-strategy Nash equilibria. Int. J. Game Theory **2**, 65–67 (1973)
23. Long, T.-T., Polukarov, M., Chapman, A.C., Rogers, A., Jennings, N.R.: On the existence of pure strategy Nash equilibria in integer-splittable weighted congestion games. In: Persiano, G. (ed.) SAGT. Lecture Notes in Computer Science, vol. 6982, pp. 236–253. Springer, Berlin (2011)
24. Young, H.P.: The evolution of conventions. Econometrica **61**(1), 57–84 (1993)