Weakly-Acyclic (Internet) Routing Games

Roee Engelberg[1]* and Michael Schapira[2]

[1] Computer Science Department, Technion, Haifa 32000, Israel.
roee@cs.technion.ac.il.
[2] Computer Science Department, Princeton University, NJ, USA.
ms7@cs.princeton.edu.

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 in [4, 12], 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.

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). Under 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 [4, 12].

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.[3] Games for which this holds (e.g., potential games [15]) are called "*weakly acyclic*" [18, 14]. Weak acyclicity has also been shown to imply that simple

* Current affiliation: Google Inc. This work was done while the author was a doctoral student at the Technion.

[3] Observe that this is equivalent to requiring that the game have no "non-trivial" sink equilibria [8, 4] under better-response dynamics (i.e., that it have no sink equilibrium of size greater than 1).

dynamics (e.g., *randomized* better-/best-response dynamics, no-regret dynamics) are guaranteed to reach a PNE [13, 14, 18]. 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., [3, 13]). 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 [14]. [14] studies Rosenthal's congestion games [17] 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., [4, 9, 12, 16]), which rely on the simple, yet important, observation that BGP can be regarded as best-response dynamics in a specific class of routing games [4, 12]. We now provide an intuitive exposition of the class of routing games in [4, 12]. We refer the reader to Sect. 2 for a formal presentation.

In the game-theoretic framework of [4, 12], 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 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 [7] (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 its ranks its induced route to the destination.

Fabrikant and Papadimitriou [4] and, independently, Levin et al. [12], 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 [9, 11, 16]).

1.3 Our Contributions: Weakly-Acyclic Routing Games

We present two interesting subclasses of routing games – (1) routing games with Byzantine players and (2) backup routing games – which capture important aspects of routing on the Internet. Routing games with Byzantine players intuitively capture scenarios where all but a few players are "well behaved", and the remaining players behave in an arbitrary manner. Such erratic "Byzantine" (in distributed computing terminology) misbehavior can, for instance, be the consequence of router configuration errors. Backup routing games model the common practice of backup routing with BGP [6].

We prove that games in both these classes are weakly acyclic, even under best-response (i.e., from every initial state there exists a best-response improvement path to a PNE). 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 best-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 best-response improvement path – whose construction we give explicitly – until a PNE is reached.

Routing games with Byzantine 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 under best-response. 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 [1, 2, 10]).

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 [6]. 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 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 [4, 12]. 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 Byzantine players, and for backup-routing games, in Sects. 4 and 5, respectively. Due to space constraints, all proofs appear in the full version of the paper.

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 $S = S_1 \times \ldots \times S_n$ and $S_{-i} = S_1 \times \ldots \times S_{i-1} \times S_{i+1} \times \ldots \times S_n$. 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 [4, 12] 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 outdegree 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 $\pi_i(\perp) < \pi_i(R)$ 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. 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 $R_i \in P_i$ 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 $\pi(R_i^s)$ if R_i^s is permitted and 0 otherwise.

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 Byzantine 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. Now, consider the case that there is a single Byzantine 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 Byzantine player is weakly-acyclic under best response and, moreover, a PNE in such a game can be found in a computationally-efficient manner.*

In contrast, we show (Appendix A) that shortest-path routing games with a single Byzantine player are no longer necessarily potential games.

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

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 $\ell(i)$ be the length of its route to d. Define $\phi(i)$ to be $\ell(i)$ if i's route is a primary route, and $B + \ell(i)$ if its route is a backup route ($\phi = B^2$ if i has no route to d). Then, it is easy to verify that $\sum_i \phi(i)$ is a valid potential function.

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 under best-response yet are not necessarily potential games.

4 Routing Games with Byzantine Players

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

4.1 Routing Policies

[5] 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. [5] generalizes these two classes of rankings as follows.

Definition 4.1. (policy consistency) *[5] 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 [5] 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. [5] presents the following generalization of export-all.

Definition 4.2. (consistent export) *[5] 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 $R_{ij} = \{Q | \ Q \in P_i$ 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 Byzantine 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 then the game is weakly acyclic under best response even in the presence of a single Byzantine player. Moreover, a PNE in such a game (with a single Byzantine 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. Let m be the (single) Byzantine node. We present a best response improvement path from any given initial strategy, which we construct via what we call the "*stabilization process*".

The stabilization process:

- **Part I:** Repeat the following two steps until no node that has no route or has a route that goes through m selects a route that does not go through m (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[4] that do not go through m (and allow them to do so), until no such node exists.
  - **Step 2:** Activate m once.

[4] 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.

– **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.

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

Claim. Each individual iteration in Part I is guaranteed to terminate.

This can be established via a simple potential function argument, where the potential function is simply the social welfare of the source nodes, when disregarding nodes whose routes go through m. By definition of Part I of the stabilization process:

Observation 4.1. *A node that does not have a route to d that goes through m at some point in Part I of the stabilization process, will not have such route later in Part I of the stabilization process.*

Consider the nodes whose route go through m in the initial strategy profile. Notice that only these nodes can get disconected. Also, by Observation 4.1, no new nodes are ever added to this set of nodes in Part I, and once nodes leave this set of nodes they never return. Therefore, after sufficiently many iterations the number of nodes whose routes go through m remains fixed. We can thus conclude that:

Claim. Part I of the stabilization process terminates.

To establish the termination of Part II 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 T 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 T.*

We observe that Lemma 4.1 concludes the proof of weak acyclicity. First, observe that it implies that in Part II of the stabilization process the network is effectively equivalent to a network for which policy consistency and consistent export hold. 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.

To prove Lemma 4.1, 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.

Lemma 4.1 now follows from the following proposition:

Proposition 4.1. *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 4.1 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 4.1 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 4.1.

Proof. **(of Proposition 4.1)** For point of contradiction, suppose that one (or more) of the bullets in the statement of Proposition 4.1 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, with the exception of red nodes and red nodes that turned black, which we allow to become white at all times and give precedence to such transitions. Whenever such an event (red node/red node turned black 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 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. Also, any node that will, at some point, want to select a route of length i can do so at the i'th round. So, in the i'th round, each node has at most $|V|$ possible routes to choose from of length i, and the set of these routes can only get smaller as the process goes on. Hence, each round has a polynomial number of steps. To complete the analysis notice that the number of rounds is polynomial. □

Can this result be extended to more than a single Byzantine 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 Byzantine nodes not exceed a certain threshold, and that the Byzantine nodes not be "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.

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 [7]. 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 [7] to handle backup routing. See [7] 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. 1(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, but is only willing to export (all of) its customer routes to neighbors its peers and providers. 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. 1(b), and suppose that all edges are primary. 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".

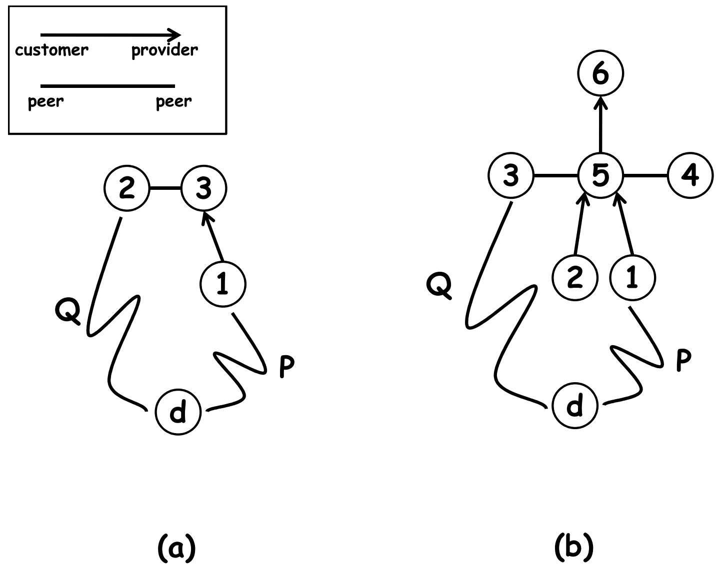

Fig. 1. Commercial Backup Routing

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 under best-response and, moreover, a PNE in such a game can be found in a computationally-efficient manner.*

Proof. To prove the theorem we construct a best-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.

Observation 5.1. (possible color changes) *If $s, s' \in S$ and s' is reachable from s via the best-response of a single node i 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, and activate nodes one by one and let nodes that want to move select their best response, while not allowing black nodes to turn gray. 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.*

Claim. **(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 [7].

Theorem 5.2. *[7] 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 claim 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 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 problemtic 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 a customer neighbor'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 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.

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) at time t, when a backup edge is chosen as a best response by some node. Then,*

– **If** x **has a customer path to** y - *then* x *has a primary customer route to* d *just before* t*.*
– **If** x **has a peer/provider path to** y - *then* x *has a primary route to* d *or a backup customer route just before* t*.*

Proof. The proof is by induction on the length of the 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 just before t.
– **If** x **has a peer/provider path to** y - then x is a peer or a customer of y; If, just before 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 just before t. If, just before 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 hypotesis, w has a primary customer route to d just before 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 just before t.
– **If** x **is a peer of** w **-** w must have a customer path to y. By the induction hypotesis, w has a primary customer route to d just before t. If this route passes through x, then since it is a customer route, x itself must have a customer route just before 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 just before t.
– **If** x **is a customer of** w **-** By the induction hypotesis, w has a primary route to d or a backup customer route just before t. If this route passes through x, then since x is a customer of w, x itself must have a customer route just before 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 just before t.

And the lemma follows. □

Claim. **(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 w 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, and assume without loss of generality that x is the node that chooses a backup edge as its best response at time t. Then, at time t node y had a (backup) customer route and by Lemma 5.1 x has a primary route or a backup customer route just before time t. We get a contradiction. □

Claim. **(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 time t. Then, at time t 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 time t. 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 5.1 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 Claim 5.2).

Proposition 5.1. (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 e (by definition of primary components). 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. □

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 convergence in the Gao-Rexford framework [7], which entails at most a polynomial number of steps.

□

We show (Appendix B) 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.*

References

[1] M. Babaioff, R. Kleinberg, and C. H. Papadimitriou. Congestion games with malicious players. In J. K. MacKie-Mason, D. C. Parkes, and P. Resnick, editors, *ACM Conference on Electronic Commerce*, pages 103–112. ACM, 2007.

[2] K. Eliaz. Fault tolerant implementation. *Review of Economic Studies*, 69:589–610, 2002.

[3] A. Fabrikant, A. D. Jaggard, and M. Schapira. On the structure of weakly acyclic games. In S. C. Kontogiannis, E. Koutsoupias, and P. G. Spirakis, editors, *SAGT*, volume 6386 of *Lecture Notes in Computer Science*, pages 126–137. Springer, 2010.

[4] A. Fabrikant and C. H. Papadimitriou. The complexity of game dynamics: BGP oscillations, sink equilibria, and beyond. In S.-H. Teng, editor, *SODA*, pages 844–853. SIAM, 2008.

[5] J. Feigenbaum, V. Ramachandran, and M. Schapira. Incentive-compatible interdomain routing. In J. Feigenbaum, J. C.-I. Chuang, and D. M. Pennock, editors, *ACM Conference on Electronic Commerce*, pages 130–139, New York, 2006. ACM.

[6] L. Gao, T. Griffin, and J. Rexford. Inherently safe backup routing with BGP. In *INFOCOM*, volume 1, pages 547–556. IEEE, 2001.

[7] L. Gao and J. Rexford. Stable internet routing without global coordination. *IEEE/ACM Trans. Netw.*, 9(6):681–692, 2001.

[8] M. Goemans, V. Mirrokni, and A. Vetta. Sink equilibria and convergence. In *FOCS*, pages 142–151. IEEE Computer Society, 2005.

[9] S. Goldberg, S. Halevi, A. D. Jaggard, V. Ramachandran, and R. N. Wright. Rationality and traffic attraction: Incentives for honest path announcements in BGP. In V. Bahl, D. Wetherall, S. Savage, and I. Stoica, editors, *SIGCOMM*, pages 267–278. ACM, 2008.

[10] R. Gradwohl. Fault tolerance in distributed mechanism design. In C. H. Papadimitriou and S. Zhang, editors, *WINE*, volume 5385 of *Lecture Notes in Computer Science*, pages 539–547. Springer, 2008.

[11] A. D. Jaggard, M. Schapira, and R. N. Wright. Distributed computing with adaptive heuristics. In *ICS*, pages 417–443. Tsinghua University Press, 2011.

[12] H. Levin, M. Schapira, and A. Zohar. Interdomain routing and games. In C. Dwork, editor, *STOC*, pages 57–66. ACM, 2008.

[13] J. R. Marden, H. P. Young, G. Arslan, and J. S. Shamma. Payoff-based dynamics in multi-player weakly acyclic games. *SIAM J. on Control and Optimization*, 48:373–396, 2009.

[14] I. Milchtaich. Congestion games with player-specific payoff functions. *Games and Economic Behavior*, 13:111–124, 1996.

[15] D. Monderer and L. S. Shapley. Potential games. *Games and Economic Behavior*, 14(1):124–143, 1996.

[16] N. Nisan, M. Schapira, G. Valiant, and A. Zohar. Best-response mechanisms. In *ICS*, pages 155–165. Tsinghua University Press, 2011.

[17] R. W. Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.

[18] H. P. Young. The evolution of conventions. *Econometrica*, 61(1):57–84, 1993.

A Shortest-Path Routing Games with a Single Byzantine Player are not Necessarily Potential Games

Consider the network G partially described by Fig. 2. We are interested in three nodes: x, y and m, where m is the Byzantine 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 Byzantine node m has the following preferences:

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

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 chooses P_1, y chooses x and m chooses P_3.
- x chooses P_1, y chooses P_2 and m chooses P_3.
- x chooses P_1, y chooses P_2 and m chooses P_4.
- x chooses m, y chooses P_2 and m chooses P_4.
- x chooses m, y chooses x and m chooses P_4.

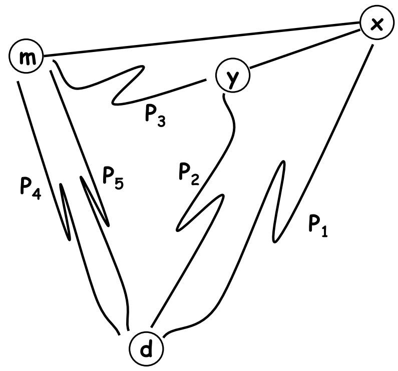

Fig. 2. A shortest-path routing game with a single Byzantine player that is not a potential game

- x chooses m, y chooses x and m chooses P_5.
- x chooses P_1, y chooses x and m chooses P_5.
- x chooses P_1, y chooses x and m chooses P_3.

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.

B Commercial Backup-Routing Games are not Necessarily Potential Games

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.

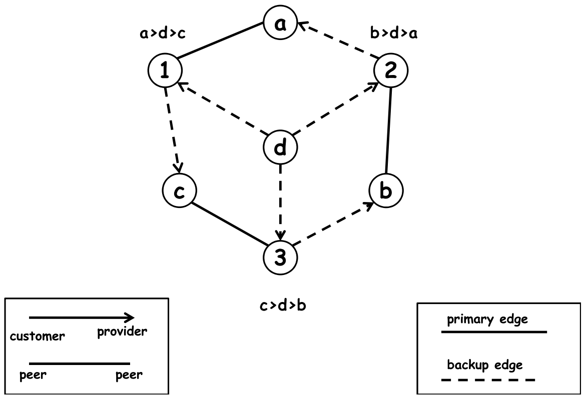

Fig. 3. A commercial backup routing game that is not a potential game

Observe that this routing game possesses (multiple) PNE, e.g., the routing state in which nodes 1 and 2 forward traffic directly to d, and node 3 forward traffic to node 1. 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): $((1d),(2d),(3c)) \rightarrow ((1a),(2d),(3c)) \rightarrow ((1a),(2d),(3d)) \rightarrow ((1a),(2b),(3d)) \rightarrow ((1d),(2b),(3d)) \rightarrow ((1d),(2b),(3c)) \rightarrow ((1d),(2d),(3c))$. 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.