SIAM J. COMPUT.
Vol. 42, No. 6, pp. 2063–2113

REDUCIBILITY AMONG FRACTIONAL STABILITY PROBLEMS*

SHIVA KINTALI†, LAURA J. POPLAWSKI‡, RAJMOHAN RAJARAMAN§, RAVI SUNDARAM§, AND SHANG-HUA TENG¶

Abstract. We resolve the computational complexity of a number of outstanding open problems with practical applications. Here is the list of problems we show to be **PPAD**-complete, along with the domains of practical significance: fractional stable paths problem (FSPP)—Internet routing; core of balanced games—economics and game theory; Scarf's lemma—combinatorics; hypergraph matching—social choice and preference systems; fractional bounded budget connection games (FBBC)—social networks; and strong fractional kernel—graph theory. In fact, we show that no fully polynomial-time approximation schemes exist (unless **PPAD** is in **FP**). This paper is entirely a series of reductions that build in nontrivial ways on the framework established in previous work. In the course of deriving these reductions, we created two new concepts—preference games and personalized equilibria. The entire set of new reductions can be presented as a lattice with the above problems sandwiched between preference games (at the "easy" end) and personalized equilibria (at the "hard" end). Our completeness results extend to natural approximate versions of most of these problems.

Key words. complexity, core, economics, game theory, kernels, Nash equilibrium, network connection games, PPAD-completeness, preference systems, Scarf's lemma, social choice, stable paths problem

AMS subject classifications. 68Q15, 68Q17, 91A10, 91A12, 91A10, 91A12

DOI. 10.1137/120874655

1. Introduction.

Intuitively, the notion of stability implies the absence of oscillations over time and encompasses the concepts of fixed points and equilibria. Stability is important in a variety of fields ranging from the practical—the Internet—to the theoretical—combinatorics and game theory. For important practical systems (e.g., the Internet), the existence and computational feasibility of stable operating modes is of profound real-world significance. On the more abstract front, the study of stable solutions to combinatorial problems has a distinguished tradition dating back to, at least, the Gale–Shapley algorithm [16]. It is often the case, as with Nash's celebrated theorem [35], that *fractional* stable points are guaranteed to exist even when integral points don't. In this paper, we focus on fractional stability and resolve the computational complexity of a set of eight problems with applications to a variety of different domains. Six of these are pre-existing problems. Below we provide elaborate motivation for two of the pre-existing problems: *fractional stable paths problem (FSPP)* and *core of balanced games*. The remaining four are *Scarf's lemma*, a fundamental result in combinatorics with several applications [40]; *fractional hypergraph matching* [1], useful for modeling preferences in social-choice and economic systems; *FBBC*, the

*Received by the editors April 24, 2012; accepted for publication (in revised form) July 15, 2013; published electronically November 5, 2013. A preliminary version (extended abstract) of this paper appeared in *Proceedings of the* 50*th Annual IEEE Symposium on Foundations of Computer Science*, 2009 [26].
http://www.siam.org/journals/sicomp/42-6/87465.html
†Department of Computer Science, Princeton University, Princeton, NJ 08540 (kintali@cs.princeton.edu).
‡Raytheon BBN Technologies, Somerville, MA 02144 (laurama@bbn.com).
§College of Computer and Information Science, Northeastern University, Boston, MA 02115 (rraj@ccs.neu.edu, koods@ccs.neu.edu).
¶Computer Science Department, University of Southern California, Los Angeles, CA 90089 (shanghua@usc.edu).

fractional version of the bounded budget connection (BBC) game [29], which models decentralized overlay network creation and social networks; and *strong fractional kernel* [2], of relevance to structural graph theory. In addition, we define two new concepts—personalized equilibria for matrix games and preference games—which are not only useful tools for carrying out reductions but are also of independent interest.

Fractional stable paths problem. Griffin, Shepherd, and Wilfong [19] showed how border gateway protocol (BGP), the routing mechanism of the Internet, can be viewed as a distributed mechanism for solving the stable paths problem (SPP). They showed that there exist SPP instances with no integral stable solutions, a phenomenon that would explain why oscillation has been observed in Internet routes. Route oscillation is viewed as a negative, since it imposes higher system overheads, reorders packets, and creates difficulties for tracing and debugging. Subsequently, Haxell and Wilfong [20] introduced FSPP: a natural fractional relaxation of SPP with the property that a (fractional) stable solution always exists. Intuitively, FSPP can be viewed as a game played between autonomous systems each of which assigns fractional capacities to the different paths leading to a destination in such a way that they maximize their utility without violating the capacity constraints of downstream nodes. Understanding the computational feasibility of finding the equilibria of this game could help to develop techniques for stable routing in the Internet.

Core of balanced games. The notion of *core* in cooperative games is analogous to that of Nash equilibrium in noncooperative games. Informally, a core is the set of all outcomes in which no coalition of players has an incentive to secede and obtain a better payoff, either viewed as a set (transferable utilities) or individually (nontransferable utilities). Necessary and sufficient conditions for the nonemptiness of the core in games with transferable utilities is given by the classic Bondareva–Shapley theorem [5, 42], which also yields a polynomial-time algorithm for finding an element in a nonempty core. Subsequently, in a celebrated paper, Scarf [40] generalized their result, developed certain sufficient *balance* conditions for the nonemptiness of the core in games with nontransferable utilities, and presented an algorithm for finding a point in the core. As noted by Jain and Mahdian in Chapter 15 of [36], "However, the worst case running time of this algorithm (like the Lemke–Howson algorithm) is exponential." Resolving the computational feasibility of finding the core in balanced games is of considerable significance in the theory of cooperative games.

Personalized equilibria for matrix games—a generalization. Imagine a business manufacturing and selling outfits consisting of a pant (solid or striped) and a shirt (cotton or wool). The manager of the location producing pants decides on the ratio of striped pants produced to solid pants produced, while the manager at the location producing shirts decides on the ratio of cotton shirts produced to wool shirts produced. Each manager is then given half the total number of shirts and pants (in the proportions decided) and has to match them into outfits and sell them at his or her own location in such a way as to maximize his or her individual profits. Personalized equilibria for matrix games capture exactly this situation: each player chooses a distribution over his or her own actions, but then each player independently customizes the matching of his or her own actions to the actions of other players in such a way as to maximize individual payoff. The concept of personalized equilibria for matrix games generalizes a number of games and problems, including FSPP and FBBC.

Preference games—a specialization. Consider a world of bloggers where each blogger has a choice of actions. They can fill their blogs with original content or they can copy from the original content on others' blogs. Naturally, each blogger has a preference order over the content of the different bloggers (as well as their own). Also, of course, more cannot be copied from another blog than the amount that the other blogger has written. The preference game models each blogger's choice of what percentage of his or her blog is original and what percentages are copied from which other blogs. Such preference games arise whenever each player has a preference among his or her actions, and his or her distribution over his or her actions is constrained by others' distributions. The definition of a preference game is surprisingly simple, making this a great candidate problem for reductions. In fact, preference games are reducible in polynomial time to all the problems considered in this paper.

1.1. Our contributions. We present a diagram (Figure 1.1) showing the different reductions. The takeaway is that all of the eight problems of interest are **PPAD**-complete. To be precise, we show that for all these problems, the exact versions are in **PPAD**, and our reductions extend to natural approximation versions to show that there are no fully polynomial-time approximation schemes (unless **PPAD** is in **FP**). Our reductions build on prior work in an intricate and involved fashion.

From a conceptual standpoint, we believe there is merit in the definitions of preference games and personalized equilibria. Preference games are very simple to describe and model a number of real-world situations, such as the blogger example mentioned earlier. Yet we can show that the set of equilibria of preference games can be nonconvex and, in fact, are hard even to approximate. As a counterbalance we show that finding equilibria in the subclass of symmetric (for a natural notion of "symmetric") preference games is in **FP**. Personalized equilibria of matrix games are, we believe, a fascinating solution concept worthy of independent study. Not only do they model real-world situations as illustrated earlier by the example of the apparel company, but they also constitute a natural generalization of a variety of predefined games, such as FSPP and FBBC. Our results on the hardness of approximating personalized equilibria for k-player games apply for $k \geq 4$. We show that finding personalized equilibria of 2-player games is in **FP**. The $k = 3$ case is open.

From a technical standpoint, we particularly wish to highlight our reduction from finding exact personalized equilibria to finding approximate personalized equilibria. To capture exact personalized equilibria, we write a linear program (LP) plus an exponential number of single-variable min constraints. These are constraints specifying that the minimum of a subset of variables is 0. Using this specification, we prove the existence of rational equilibria, i.e., equilibria with rational numbers. Furthermore, we reduce to approximate personalized equilibria by showing that an ϵ-approximate equilibrium for sufficiently small ϵ points us to a subset of the variables that can be set to 0 to simultaneously satisfy all of the min constraints, leaving us with a polynomially sized feasible LP for an exact equilibrium. With this reduction in hand and an additional technical bound on the size of short feasible vectors, we are then able to carry through the reduction to the End of the Line problem to show that personalized equilibria are in **PPAD**.

1.2. Related work. Nash [34, 35] profoundly changed game theory by demonstrating the existence of mixed equilibria. Decades later, on the computational front [36], the complexity class **TFNP** was introduced by Megiddo and Papadimitriou [33]. Papadimitriou's seminal work [38] not only defined a number of syntactic subclasses of **TFNP** (including **PPAD**), but also proved that a variety of problems, including discrete versions of Brouwer's fixed point theorem and Sperner's lemma,

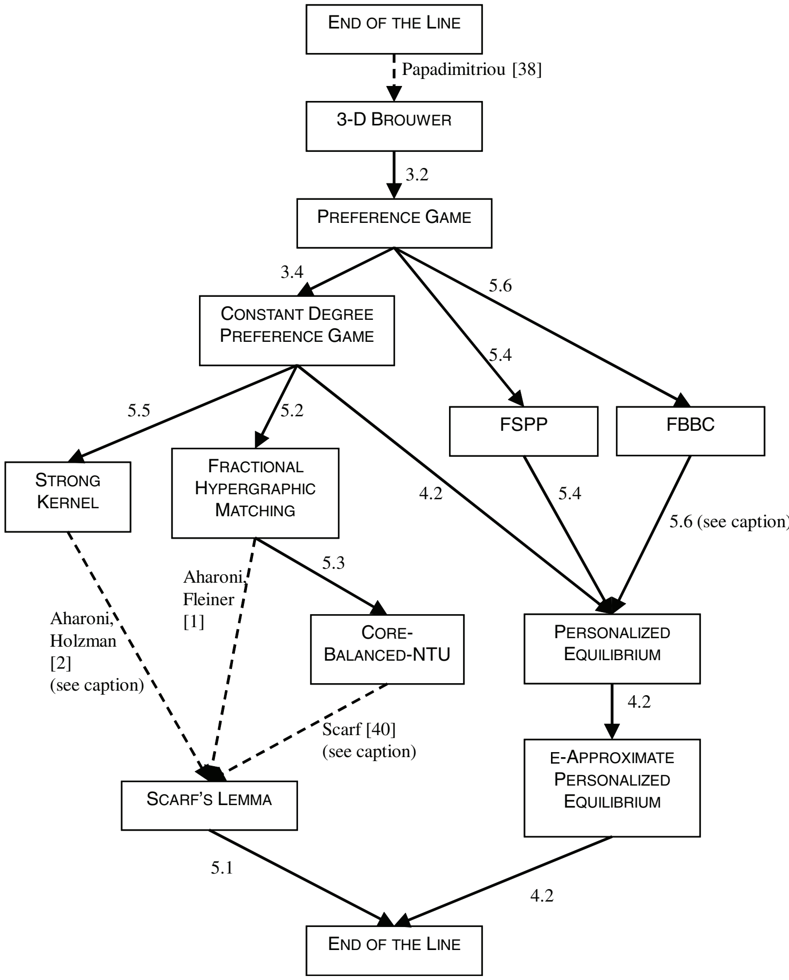


Fig. 1.1. *We show these problems to be* **PPAD***-complete. Each reduction line is labeled with the number of the section or the reference where the reduction can be found. Two of these reductions,* Strong Kernel $\leq_P$ Scarf*, and* Core-Balanced-NTU $\leq_P$ Scarf*, are only polynomial-time reductions for the specific versions of the problems discussed in this paper. In our definition of* Strong Kernel*, formally given in section* 5.5*, we assume that the largest clique in the graph has constant size, since otherwise it is not clear whether the problem is even in* **TFNP**. Core-Balanced-NTU*, as defined in section* 5.3*, assumes that the game description explicitly lists the possible coalitions and their Pareto-optimal outcomes. Finally, for the reduction to* Personalized Equilibrium*, we require that the flow paths in the* Fractional BBC *problem are explicitly given; the reduction from* Preference Game *to* Fractional BBC *extends to this class of instances.*

are **PPAD**-complete. The problem of finding Nash equilibria was left open. Recently, a series of papers comprising different author combinations of the two teams, Daskalakis, Goldberg, and Papadimitriou [18, 13] and Chen, Deng, and Teng [6, 7, 8, 9] culminated in establishing that *approximating* Nash equilibria with two players, 2-Nash, is hard. The reductions in our work build on the framework established in these papers.

BGP has been the focus of much attention since its inception [39, 44]. As mentioned earlier, SPP was introduced by Griffin, Shepherd, and Wilfong [19] to explain the nonconvergence of BGP [46]. Haxell and Wilfong [20] defined FSPP and proved the existence of an equilibrium using Scarf's lemma and a compactness-type argument. They left open the complexity of finding an equilibrium. Our reduction from personalized equilibria to the END OF THE LINE problem is a different approach that generalizes the Haxell–Wilfong existence result while preserving computational tractability. Kintali [25] presented a distributed algorithm for finding an ϵ-approximation for FSPP that is guaranteed to converge, although no bounds are given on the time-to-convergence (our results imply that a polynomial-time bound is unlikely).

Cooperative games, the study of mechanisms used to sustain and enforce cooperation among willing agents, has a rich and extensive literature [10, 17, 11, 15, 28]. As mentioned earlier, in a celebrated paper Scarf [40] generalized the classical Bondareva–Shapley theorem [5, 42] result and developed an algorithm for finding a point in the core of balanced games with nontransferable utilities. More recently, Markakis and Saberi [32] and Immorlica, Jain, and Mahdian [22] studied certain classes of games with nontransferable utilities in the context of the Internet; however, it is unclear whether their problems are even in **TFNP**. Scarf's paper [40] also contains Scarf's lemma, an important result in combinatorics which played a crucial part in the FSPP existence proof of Haxell and Wilfong [20]. Aharoni and Holzman [2] proved that every clique-acyclic digraph has a strong fractional kernel, and Aharoni and Fleiner [1] proved that every hypergraphic preference system has a fractional stable matching. Both of these proofs are based on Scarf's lemma. The computational complexity of these problems was left unresolved.

The BBC game, introduced in [29, 30], builds on a large body of work in network formation games [23, 4]. A direct precursor to BBC games was introduced by Fabrikant et al. [14]. Fractional BBC games were introduced in [30], but the problem of finding an equilibrium was left open.

2. The class PPAD. A major contribution of this paper is to expand the set of problems known to be **PPAD**-complete. The class **PPAD** (*Polynomial Parity Argument in a Directed graph*) was introduced by Papadimitriou in [38], which defined a number of syntactic classes in the semantic class **TFNP**, or the set of all total search problems. A search problem $\mathcal{S}$ consists of a set of inputs $I_{\mathcal{S}} \subseteq \Sigma^*$ such that for each $x \in I_{\mathcal{S}}$ there is an associated set of solutions $\mathcal{S}_x \subseteq \Sigma^{|x|^k}$ for some integer k. For each $x \in I_{\mathcal{S}}$ and $y \in \Sigma^{|x|^k}$, it is decidable in polynomial time whether or not y is in $\mathcal{S}_x$. A search problem is *total* if $\mathcal{S}_x \neq \emptyset$ for all $x \in I_{\mathcal{S}}$. **TFNP** is the set of all total search problems [33]. Since every member of **TFNP** is equipped with a mathematical proof that it belongs to **TFNP**, a number of syntactic classes can be defined based on their proof styles. The complexity class **PPAD** is the class of all search problems whose totality is proved using a directed parity argument.

Problems in **PPAD** are reducible to the END OF THE LINE problem. In END OF THE LINE, we are given a finite directed graph in which each node has at most one outgoing edge and at most one incoming edge. The input to the problem is not a complete list of the nodes and edges; such a list may be exponentially large in the size of the input. Instead, we are given an initial source node and a circuit. The circuit takes a node name as input and in polynomial time returns the *next* node (the other end of the outgoing edge from the input node) and the *previous* node (the other end of the incoming edge into the input node). If the input node is a source (or sink), null is returned as the previous (or next) node. The problem for END OF THE LINE is to find a sink or a source other than the initial source.

Throughout this paper, we use PROBLEM A $\leq_P$ PROBLEM B to mean "There exists a polynomial-time reduction from finding a stable point in PROBLEM A to finding a stable point in PROBLEM B."

3. Preference games. In this section, we define a very simple game, the *preference game*. Each player has a preference list across the set of players and must assign weight to each player. No player may put more weight on another player than that player puts on itself. A best response for a player occurs when that player cannot move weight from a lower preference player to a higher preference player. We show in section 3.2 that when preferences are symmetric, it is very easy to find an equilibrium in which all weights are either 0 or 1. However, in section 3.3 we show that the set of equilibria in general preference games may not be convex, implying that we cannot hope to find an equilibrium using convex programming, and in section 3.4, we show that finding an equilibrium in general preference games is **PPAD**-hard. In section 3.5, we define an ϵ-approximate equilibrium for the preference game and extend our **PPAD**-hardness result to approximate equilibria. Our notion of approximation carries through all of the reductions in later sections, so we prove that there are no fully polynomial-time approximation schemes (unless **PPAD** is in **FP**) for computing stable points in any of the problems discussed in this paper. Finally, in section 3.6, we define the *degree* of a preference game and show that any preference game can be reduced to a preference game with constant degree.

3.1. Preference games. In a preference game with a set S of players, each player's strategy set is S. Each player $i \in S$ has a preference relation $\succeq_i$ among the strategies.[1] For strategies j and k, $j \succeq_i k$ indicates that player i prefers j at least as much as k. We say that $j \succ_i k$ if $j \succeq_i k$ is true but $k \succeq_i j$ is not true. When it is clear from the context that we are talking about the preferences for player i, we write $j \succeq k$ instead of $j \succeq_i k$.

The preference relation for a player i can be said to define a *preference list* for player i, which is a linear ordering of the strategies such that j is before k only if $j \succeq_i k$. Note that preference lists are not necessarily unique, since ties are allowed.

Each player i chooses a *weight distribution*, which is an assignment $w_i : S \to [0, 1]$ satisfying two conditions: (a) the weights add up to 1: $\sum_{j \in S} w_i(j) = 1$; and (b) the weight placed by i on j is no more than the weight placed by j on j: $w_i(j) \le w_j(j)$ for all $i, j \in S$.

Given weight assignments w_i, w'_i, and w_{-i} such that (w_i, w_{-i}) and (w'_i, w_{-i}) are both feasible, we say w_i is *lexicographically at least* w'_i (with respect to w_{-i}) if for all $j \in S$, $\sum_{k \succeq_i j} w_i(k) \ge \sum_{k \succeq_i j} w'_i(k)$. We say that w_i is *lexicographically maximal* (implied: with respect to w_{-i}) if (w_i, w_{-i}) is feasible and w_i is lexicographically at least every assignment w'_i such that (w'_i, w_{-i}) is feasible. Note that if w_i is lexicographically maximal with respect to w_{-i}, then there is some j such that (a) $w_i(k) = w_k(k)$ for all k with $k \succ_i j$, and (b) $w_i(k) = 0$ for all k with $j \succ_i k$.

An equilibrium in a preference game is an assignment $w = \{w_i : i \in S\}$ such that w_i is lexicographically maximal with respect to w_{-i} for all $i \in S$.

Every preference game has an equilibrium, a fact which can be shown using standard fixed point theorems; we defer the proof to section 4, where we show the existence and PPAD-membership of a more general class of equilibria.

> PREFERENCE GAME: Given a set of players $[n]$, each with strategy set $[n]$, and a preference relation $\succeq_i$ among the strategies for each player i. Find a feasible weight assignment w such that for all i, w_i is lexicographically maximal with respect to w_{-i}.

[1] A preference relation is a binary relation that is transitive and complete.

3.2. Symmetric preference games. In a *symmetric preference game*, there exists an ordering of the players $\{1, \ldots, n\}$ such that we have the following symmetry in the preferences: if $i \leq j$, and if $i \succeq_j j$, then $j \succeq_i i$. In other words, if a player j is later in the order than i, and if j prefers i over itself, then the earlier player i also prefers j over itself.

This ordering can be easily found from a directed graph in which each node represents a player and an edge exists from player i to player j if and only if $j \succeq_i i$. In this case, if we remove all double edges, then the ordering is simply a topological ordering of the nodes in the graph.

In a sense, this notion of symmetry captures preferences based on some metric. For instance, recall the blogger example. Without symmetry, each blogger has a preference list for which other authors he or she would choose to reference rather than write new content, but there is no relationship between the preferences of different bloggers. In reality, it is likely that each blogger would refer to another blog if and only if the author's specialty and opinions closely matched his or her own. We can imagine a metric measuring the expertise of each blogger, such that two blogs are likely to reflect similar views if and only if the points in the metric are nearby. Each blogger has a personal threshold for how similar another blogger's expertise must be before using borrowed content. There is a natural symmetry: if two players are very close in the metric, they are both more likely to refer to each other's content, since this content is likely to be similar. If two players are distant, they are less likely to agree. The numeric threshold in the metric of *how close* another blogger must be before an author is willing to borrow content defines the ordering of the players.

THEOREM 3.1. *In any symmetric preference game, an equilibrium in which all weights are* 0 *or* 1 *can be found in polynomial time.*

Proof. If our preference rules obey this style of symmetry, we can use Algorithm 1 to find an equilibrium.

Algorithm 1. Finding an equilibrium in a symmetric preference game.

Sort the players into their symmetry order.
Set all weights to -1.
for $i = 1, \ldots, n$ **do**
  if $w_i(i) = -1$ **then**
    Assign $w_i(i) = 1$.
    for $j = i + 1, \ldots, n$ **do**
      Assign $w_i(j) = 0$.
      if $j \succeq_i i$ **then**
        Assign $w_j(j) = 0$.
for $i = 1, \ldots, n$ **do**
  if $w_i(i) = 0$ **then**
    Find the player j with $w_j(j) = 1$ that is highest in i's preference list.
    Assign $w_i(j) = 1$.
    Assign $w_i(k) = 0$ for all other $k \neq j$.

Since each player has weight 1 assigned to exactly one strategy, Algorithm 1 assigns a feasible set of weights. To show that Algorithm 1 finds an equilibrium, we must show that the results of the algorithm obey the following: (a) if $w_i(i) = 1$, then there is no j such that $j \succeq_i i$ with $w_j(j) = 1$, and (b) if $w_i(i) = 0$, then there is some j such that $j \succeq_i i$ with $w_j(j) = 1$.

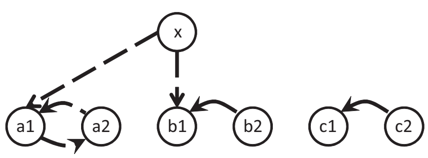
To show (a), consider the point in the algorithm at which $w_i(i)$ is set to 1. By this point, we have already looked through all j ahead of i in the ordering. Since $w_i(i)$ is still -1, for each j for which we assigned $w_j(j) = 1$, none had $i \succeq_j j$. By symmetry, this means that no j ahead of i in the ordering has $w_j(j) = 1$ and $j \succeq_i i$. Now, for all j following i in the ordering, if $j \succeq_i i$, then we assign $w_j(j) = 0$ immediately after we assign $w_i(i) = 1$.

To show (b), consider the point at which we assigned $w_i(i) = 0$. We had just assigned $w_j(j) = 1$ for some j ahead of i in the order. We found that $i \succeq_j j$, which by symmetry implies that $j \succeq_i i$, as required. ∎

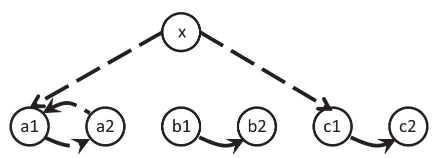
3.3. Nonconvexity. Although symmetric preference games have a simple equilibrium which can be found in polynomial time, general preference games are more complex. In this section, we show that the set of equilibria for a preference game may not be convex.

THEOREM 3.2. *There exists an instance of the preference game for which the set of equilibria is not convex.*

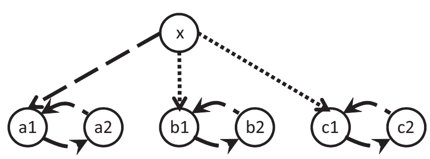
Proof. Consider the instance of the preference game in Figure 3.1. We have three sets of two players each, a_1, a_2, b_1, b_2, c_1, c_2, and one additional player, x. The preference lists for these nodes are a_1: (a_2, a_1); a_2: (a_1, a_2); b_1: (b_2, b_1); b_2: (b_1, b_2); c_1: (c_2, c_1); c_2: (c_1, c_2); x: (a_1, b_1, c_1, x). (Each list gives strategies in order from most preferred to least preferred.) Each list is enumerated only until the player itself; the rest of each list is irrelevant, since a player can assign as much weight as necessary to itself. We now show two equilibria whose linear combination is not an equilibrium. In equilibrium w (Figure 3.1(a)), $w_{a_1}(a_1) = \frac{1}{2}$, $w_{a_1}(a_2) = \frac{1}{2}$, $w_{a_2}(a_2) = \frac{1}{2}$, $w_{a_2}(a_1) = \frac{1}{2}$, $w_{b_1}(b_1) = 1$, $w_{b_2}(b_1) = 1$, $w_{c_1}(c_2) = 1$, $w_{c_2}(c_2) = 1$, $w_x(a_1) = \frac{1}{2}$, $w_x(b_1) = \frac{1}{2}$. In equilibrium w' (Figure 3.1(b)), $w'_{a_1}(a_1) = \frac{1}{2}$, $w'_{a_1}(a_2) = \frac{1}{2}$, $w'_{a_2}(a_2) = \frac{1}{2}$, $w'_{a_2}(a_1) = \frac{1}{2}$, $w'_{b_1}(b_2) = 1$, $w'_{b_2}(b_2) = 1$, $w'_{c_1}(c_1) = 1$, $w'_{c_2}(c_1) = 1$, $w'_x(a_1) = \frac{1}{2}$, $w'_x(c_1) = \frac{1}{2}$. It is easy to verify that w and w' are both equilibria, and in a solution $\lambda \cdot w + (1 - \lambda) \cdot w'$ (for any $\lambda > \frac{1}{4}$) (Figure 3.1(c) shows $\lambda = \frac{1}{2}$), player x would do better by moving



(a) The a players assign weights 1/2, 1/2, the b players both use b_1, the c players both use c_2.



(b) The a players assign weights $1/2 - 1/2$, the b players both use b_2, the c players both use c_1.



(c) Combining half of each equilibrium, x will assign 1/2 to a_1, 1/4 to each of b_1 and c_1. x could improve by assigning weight only to a_1 and b_1.

FIG. 3.1. *Example of an instance of the preference game for which the equilibrium set is not convex.*

more weight to its second preference. Therefore, the convex combination of w and w' is not an equilibrium. ∎

3.4. PPAD-hardness. We show that finding an equilibrium in preference games is **PPAD**-hard. We will follow the framework of [13], which shows that finding a Nash equilibrium in a degree-3 graphical game is **PPAD**-hard, using a reduction from the **PPAD**-complete problem 3-D BROUWER. In this problem, we are given a three-dimensional (3-D) unit cube in which each dimension is broken down into 2^{-n} segments—thereby dividing the cube into 2^{3n} cubelets. We are also given a circuit C that takes as input the three coordinates of the center of a cubelet (each as an n-bit number) and returns a 2-bit number that represents one of four 3-D vectors: $(1, 0, 0)$, $(0, 1, 0)$, $(0, 0, 1)$, and $(-1, -1, -1)$, represented by say 00, 01, 10, and 11, respectively. The circuit satisfies the following boundary conditions: $C(0, j, k) = 00$, $C(i, 0, k) = 01$, $C(i, j, 0) = 10$, and $C(2^n - 1, j, k) = C(i, 2^n - 1, k) = C(i, j, 2^n - 1) = 11$, with ties resolved arbitrarily. A solution to the 3-D BROUWER instance is a cubelet center such that the set of results obtained by running the circuit on each of the at most eight cubelets surrounding the vertex contains each of the four vectors at least once.

As in [13], we will construct a set of gadgets to simulate various arithmetic operators, logical operators, arithmetic comparisons, and other operators. We will then follow their framework to systematically combine these gadgets to simulate the input Boolean circuit and to encode the geometric condition of discrete fixed points in the 3-D BROUWER instance. In the preference game we construct, we specify the preference relation of any player P by an ordered list of a subset of the players, with the last element being P, also referred to as the "self" strategy. When we say that a player P *plays itself* with weight v, we mean that P assigns a weight of v to strategy P. We'll engineer the payoffs such that the game is in equilibrium only if the weights assigned by certain players to *themselves* successfully echo the inputs and outputs of eight copies of the circuit that surround a solution vertex of the 3-D BROUWER instance.

For this reduction, we require the following sets of players:

1. One player for each of the three dimensions (the *coordinate players*). If the graph is an equilibrium, each coordinate player plays itself with weight equal to its coordinate of the 3-D BROUWER solution vertex.
2. One player for each of the bits of each of the three coordinates (the *bit players*). In order to force these players to correctly represent the bits, we need some additional players. Assuming we've correctly calculated the first $i - 1$ bits of coordinate x (call them $x_0, \dots, x_{i-1}$), we can create the ith bit as follows. One player will play itself with weight $p_i = x - \sum_{j=0}^{i-1} \frac{x_j}{2^j}$. The bit player will play itself with weight equal to the ith bit. If $p_i \geq \frac{1}{2^i}$, then this bit should be 1. Otherwise, it should be 0. Therefore, in order to properly extract the bits, we create the following four types of players:
   (a) HALF player: in any equilibrium in which a given player plays itself with weight a, the HALF player will play itself with weight $\frac{a}{2}$.
   (b) DIFF player: in any equilibrium in which two given players play themselves with weights a and b, the DIFF player will play itself with weight $a - b$.
   (c) VALUE player: in any equilibrium, the VALUE player plays itself with weight $\frac{1}{2}$. This can be easily created by combining a player whose first preference is itself with a HALF player.

(d) LESS player: in any equilibrium in which two given players play themselves with weights a and b, respectively, the LESS player plays itself with weight 1 if and only if $a \geq b$, and plays itself with weight 0 otherwise. (Actually, the LESS player we create will be inaccurate if a and b are very close, which we discuss further below.)

3. One player simulates each type of gate used in the circuit of the 3-D Brouwer instance. For this, we create three more types of players.
   (e) AND player: in any equilibrium in which two given players play themselves with weights a and $b \in \{0, 1\}$, the AND player will play itself with weight $a \wedge b$.
   (f) OR player: in any equilibrium in which two given players play themselves with weights a and $b \in \{0, 1\}$, the OR player will play itself with weight $a \vee b$.
   (g) NOT player: in any equilibrium in which a given player plays itself with weight $a \in \{0, 1\}$, the NOT player will play itself with weight $\neg a$.
4. Finally, we need to ensure that the graph is in equilibrium if and only if all four vectors are represented in the results of the eight circuits. As in [13], we will represent the output of each circuit using 6 bits, one each for $+x, -x, +y, -y, +z, -z$. Now, the four possible result vectors are represented as 100000, 001000, 000010, and 010101. We can use these circuit results with only two additional types of players to feed back into the original coordinate players. First, we will create an OR player for each of the 6 bits (over the eight vertices), which yields a result of six 1's if and only if this is a solution vertex. Therefore, an AND player for each coordinate will all return 1 if and only if this is a solution vertex; at least one of the coordinates will be 0 otherwise. We can turn this around using a NOT player for each coordinate, so that we get all 0's if and only if this is a solution vertex. Finally, we need the last two new player types, which we'll use to add these results back to a copy of the original coordinates (the result will be the original coordinate player).
   (h) COPY player: in any equilibrium in which a given player plays itself with weight a, the COPY player will also play itself with weight a.
   (i) SUM player: in any equilibrium in which two given players play themselves with weights a and b, the SUM player will play itself with weight $\min(a + b, 1)$.

   Let the addition of the circuit outputs, each consisting of 6 bits, be represented as $\Delta = (\Delta x^+, \Delta x^-, \Delta y^+, \Delta_y^-, \Delta_z^+, \Delta_z^-)$. We add the results back to the coordinate players by subtracting Δx^- from Δx^+ (resp., Δy^- from Δy^+ and Δz^- from Δz^+) using a DIFF player and then using a SUM player to add the result to a copy of the x-coordinate (resp., y-coordinate and z-coordinate) player that was created using a COPY player.

   If the coordinates represent a solution vertex to the 3-D Brouwer instance, then all the values we've added back in will be zero; thus the coordinate players cannot do better by changing their strategies. On the other hand, if the coordinates do not form a solution vertex, then we can argue that players will not be in equilibrium. If the point (x, y, z) (given by the strategies of the coordinate players) is completely in the interior of the cube, i.e., not located in any of the boundary cubelets, then any nonzero coordinate in Δ changes the player's strategies, hence making it a nonequilibrium. For points on the boundary, we invoke the boundary condition specified for the coloring. There are several cases; we consider two here, and others follow similarly.

If (x, y, z) is in a cubelet that is adjacent to the $x = 0$ face but not the $y = 0$ or $z = 0$ face, then at least one of the neighboring result vectors is 100000 (given the boundary BROUWER condition for the $x = 0$ face). Since all four desired vectors are not present, one of the other three vectors is missing. If 010101 is missing, then the x-coordinate player is not in equilibrium; otherwise, one of the 001000 or 000010 vectors is missing, and the corresponding coordinate player is not in equilibrium. Considering another case, if (x, y, z) is in a cubelet that is adjacent to the $x = 1$ and $y = 0$ faces but not the $z = 0$ face, then at least two of the neighboring result vectors are 010101 and 001000. Again, since all four desired vectors are not present, one of the other two vectors is missing. If 100000 is missing, then the x-coordinate player is not in equilibrium; otherwise, the z-player is not in equilibrium. The other boundary cases follow similarly. We omit further details here since, as we discuss below, the addition of the circuit results and feedback need, in fact, to be further refined to account for the inaccuracies in the LESS player that cause problems with points at the boundary of the cube.

We now describe how to create the new types of players (gadgets) required for the reduction. For each of these gadget definitions, we assume that we are given a preference game in which v_1 (resp., v_2) denotes the weight of player X (resp., player Y) in any equilibrium. For the first three gadgets, we assume $v_1, v_2 \in \{0, 1\}$. For the rest of the gadgets, we assume $v_1, v_2 \in [0, 1]$.

$\mathbf{OR}(X, Y)$. We can add a new node $R = \mathrm{OR}(X, Y)$ that will play itself with weight $v_1 \vee v_2$ in any equilibrium. Create a node R_1 with preference list (X, Y, R_1). Let node R's preference list be (R_1, R). Now, if v_1 and/or v_2 is 1, then R_1 will play R_1 with weight 0, so R will play itself with weight 1. If both v_1 and v_2 are 0, then R_1 will play itself with weight 1, so R will play R_1 with weight 1 and R with weight 0.

$\mathbf{NOT}(X)$. We can add a new node $N = \mathrm{NOT}(X)$ that will play itself with weight $\neg v_1$ in any equilibrium. Let node N's preference list be (X, N). Clearly, N will play X as much as v_1 and will play N with the remainder.

$\mathbf{AND}(X, Y)$. We can add a new node $A = \mathrm{AND}(X, Y)$ that will play itself with weight $v_1 \wedge v_2$ in any equilibrium. We assemble the OR and NOT gadgets as follows: $\mathrm{NOT}(\mathrm{OR}(\mathrm{NOT}(X), \mathrm{NOT}(Y)))$.

$\mathbf{SUM}(X, Y)$. We can add a new node $S = \mathrm{SUM}(X, Y)$ that will play itself with weight $\min(1, v_1 + v_2)$ in any equilibrium. Create a node S_1 with preference list (X, Y, S_1). Let node S's preference list be (S_1, S). Now, clearly node S_1 will play S_1 with weight $\max(0, 1 - v_1 - v_2)$, and node S will play S_1 with the same weight. So node S will play itself with weight $1 - \max(0, 1 - v_1 - v_2)$. In other words, if $v_1 + v_2 \geq 1$, then S will play itself with weight 1. Otherwise, S will play itself with weight $1 - 1 + v_1 + v_2 = v_1 + v_2$, as desired. Note that $\mathrm{SUM}(X, Y)$ implements $\mathrm{OR}(X, Y)$ when $X, Y \in \{0, 1\}$.

$\mathbf{DIFF}(X, Y)$. We can add a new node $D = \mathrm{DIFF}(X, Y)$ that will play itself with weight $v_1 - v_2$ if $v_1 > v_2$, or 0 otherwise in any equilibrium. Create a node D_1 with preference list (X, D_1). D_1 will play itself with weight $1 - v_1$. Now set the preference list for D to (D_1, Y, D). D will play itself with weight $\min(0, 1 - (1 - v_1) - v_2) = \min(0, v_1 - v_2)$, as desired.

$\mathbf{COPY}(X)$. We can add a new node $C = \mathrm{COPY}(X)$ that will play itself with weight v_1 in any equilibrium. Create a node C_1 with preference list (X, C_1). C_1 will play itself with weight $1 - v_1$. Set the preference list for node C to (C_1, C). C will play C_1 with weight $1 - v_1$, leaving weight v_1 on C.

DOUBLE(X). We can add a new node $M = \text{DOUBLE}(X)$ that will play itself with weight $\min(1, v_1 * 2)$ in any equilibrium. Create player $M_1 = \text{COPY}(X)$ and set M as $\text{SUM}(X, M_1)$.

LESS(X, Y). Given ϵ_l ($0 < \epsilon_l \leq \frac{1}{2}$), we can add a new node $L = \text{LESS}(X, Y)$ to the game that in any equilibrium will play only itself if $v_1 - v_2 \geq \epsilon_l$, and will play L_1 (for a new node L_1) if $v_1 \leq v_2$. First create $D = \text{DIFF}(X, Y)$. Then create $M_1 = \text{DOUBLE}(D)$. For $i = 1$ to $-\log \epsilon_l$, create player $M_{i+1} = \text{DOUBLE}(M_i)$. Call the last DOUBLE player node L, and the extra player for the sum player of the last DOUBLE player node L_1. If $v_1 \leq v_2$, the DIFF player will return 0, so player L will play the result of multiplying 0 by 2 many times, or 0. If $v_1 - v_2 \geq \epsilon_l$, player L will play the max of 1 and $(v_1 - v_2) * 2^{-\log \epsilon_l} = (v_1 - v_2) * \frac{1}{\epsilon_l} \geq \frac{\epsilon_l}{\epsilon_l} = 1$.

HALF(X). We can add a new node $H = \text{HALF}(X)$ that will play itself with weight $v_1/2$ in any equilibrium. Create a node H_1 with preference list (X, H_1). H_1 will play itself with weight $1 - v_1$. Then create two more nodes: H_2 and H_3. Node H_2 has preference list (H_1, H_3, H_2). Node H_3 has preference list (H_1, H, H_3). Set the preference list for node H to be (H_1, H_2, H). Each of H, H_2, and H_3 will use its first choice with weight $1 - v_1$, leaving v_1 for its other two choices. Then, we have $w_H(H) + w_H(H_2) = v_1$, $w_{H_2}(H_2) + w_{H_2}(H_3) = v_1$, and $w_{H_3}(H_3) + w_{H_3}(H) = v_1$. In any equilibrium, it must be true that $w_H(H_2) = w_{H_2}(H_2)$, $w_{H_2}(H_3) = w_{H_3}(H_3)$, and $w_{H_3}(H) = w_H(H)$. Solving this gives $w_H(H) = w_H(H_2) = w_{H_2}(H_2) = w_{H_2}(H_3) = w_{H_3}(H_3) = w_{H_3}(H) = \frac{v_1}{2}$.

As in [13], our LESS player plays the specified action (itself, in our case) with weight 1 if $v_1 \geq v_2 + \epsilon_l$, and plays itself with weight 0 if $v_1 \leq v_2$, but will play some unspecified fraction on itself if $v_2 < v_1 < v_2 + \epsilon_l$. We use the LESS player to extract the bits representing the coordinates of a cubelet to be passed into the circuit. This procedure is identical to that of [13]. Let X denote the x-coordinate player, and let $X_1 = \text{COPY}(X)$. For i from 1 through n, we create players $B_i = \text{LESS}(2^{-i}, X_i)$ and $X_{i+1} = \text{DIFF}(X_i, \text{HALF}^i(B_i))$, where HALF^i indicates applying the HALF gadget i times. It can be shown that as long as x is not too close to a multiple of 2^{-n}, we will extract its n bits correctly. If this is not the case, however, we will not properly extract the bits, and our circuit simulation may return an arbitrary value. We resolve this problem using the same average technique as in [13]: we compute the circuit for a large constant number of points surrounding the vertex and take the average of the resulting vectors. We separate out the positive and negative components of the result vectors, as before, to obtain an average vector $\Delta = (\Delta x^+, \Delta x^-, \Delta y^+, \Delta_y^-, \Delta_z^+, \Delta_z^-)$. We then scale the vector down, using several invocations of the HALF gadget, so that the magnitude is sufficiently smaller than the side length of a cubelet. For each dimension, we next add that dimension of the scaled vector to a copy of the appropriate coordinate player, by adding the positive component first and then subtracting the negative component, to obtain the original coordinate player. Since these details are identical to those of [13, Lemma 21], we omit them.

Based on the above gadgets and the framework from [13], we get the following.

THEOREM 3.3. 3-D BROUWER $\leq_P$ PREFERENCE GAME.

We note that in the construction used in the proof of this theorem, the preference relation defined in each gadget can, in fact, be replaced by a linear relation so that each node has a strict preference order among the nodes in the game. This is simply because all of the preference lists defined in the gadgets form a linear order and the ordering among nodes not in the preference list of any node is immaterial and hence can be set arbitrarily. Thus, Theorem 3.3 in fact holds for preference games with linear preference relations for each node.

3.5. Approximate equilibria. Given the hardness of finding exact equilibria in preference games, a natural next question is whether it is easier to find approximate equilibria. We define an ϵ-equilibrium of a k-player preference game to be a set of weight distributions $w_1, \ldots, w_k$ that satisfy the following conditions for every player i: (a) $\sum_j w_i(j) = 1$; (b) for each j, $w_i(j) \leq w_j(j) + \epsilon$; and (c) for each j, either $\sum_{\ell:\ell \succeq j} w_i(\ell) \geq 1 - \epsilon$ or $|w_i(j) - w_j(j)| \leq \epsilon$. In other words, the weight assigned by a player i on another player j is at most ϵ more than the weight assigned by j on itself, and for any i and j, either i plays a total weight of at least $1-\epsilon$ on players it prefers at least as much as j, or the weight assigned by i on j differs from that assigned by j to itself by at most ϵ. The problem of finding an ϵ-equilibrium is ϵ-Approximate Preference Game.

Theorem 3.4. Brouwer $\leq_P$ ϵ-Approximate Preference Game. *Thus, it is* **PPAD**-*hard to find an ϵ-equilibrium for preference games for ϵ inverse polynomial in n.*

Proof. Our proof follows the framework of [8, 9] for proving the hardness of approximating Nash equilibria in 2-player games. This framework starts with a high-dimensional discrete fixed point problem, Brouwer, which is also **PPAD**-complete. The input to Brouwer is a Boolean circuit that assigns a color from $\{1, \ldots, n, n+1\}$ to each interior node of an n-dimensional grid $\{0, 1, \ldots, 8\}^n$. This grid has about 2^{3n} cells, each of which is an n-dimensional hypercube. The discrete fixed point is defined to be a panchromatic simplex inside a hypercube. This framework of [8, 9] uses a new geometric condition for discrete fixed points, which requires that the average of n^3 sampled points in the interior of the targeted panchromatic simplex is inverse-polynomially close to the zero vector. The rest of the proof follows the framework of [13].

Our broad definition of an ϵ-equilibrium poses additional technical challenges which did not occur in the reductions of [8, 9]. In particular, in the presence of errors, our Boolean gadgets only approximately simulate the Boolean operations, while in previous reductions, the Boolean gadgets are precise. We prevent magnification of errors in the Boolean simulation by strategically adding a LESS gadget to correct errors after each logic step.

We focus on bounding the errors for the gadgets of Theorem 3.3 and the addition of the extra LESS gadgets. Other details closely match those of [8, 9, 13].

Let ϵ_l (the measure of the fragility of our LESS gadget) be a real number such that $\epsilon \leq \epsilon_l^3$. Then, we have the following error bounds.

Lemma 3.5. *Assuming node X plays itself with weight v_1', $v_1 - \epsilon_l \leq v_1' \leq v_1 + \epsilon_l$, and node Y plays itself with weight v_2', $v_2 - \epsilon_l \leq v_2' \leq v_2 + \epsilon_l$, each of the Boolean gadgets plays itself within $\pm(2\epsilon_l + 6\epsilon)$ of the correct value for the correct v_1 and v_2 inputs.*

Proof. OR: if v_1 and/or v_2 is 1, then v_1' and/or v_2' is at least $1 - \epsilon_l$, and node R_1 will play R_1 with weight at most $\epsilon_l + \epsilon$, so R will play R with weight at least $1 - \epsilon_l - 2\epsilon$. If both v_1 and v_2 are 0, then v_1' and v_2' are at most ϵ_l, and node R_1 will play R_1 with weight at least $1 - 2\epsilon_l - 2\epsilon$, so R will play R with weight at most $2\epsilon_l + 3\epsilon$. NOT: if $v_1 = 1$, then v_1' is at least $1 - \epsilon_l$, and node N will play itself with weight at most $\epsilon_l + \epsilon$. If $v_1 = 0$, then v_1' is at most ϵ_l, and node N will play N with weight at least $1 - \epsilon_l - \epsilon$. AND: The AND gadget concatenates other new players to get $\neg(\neg v_1 \vee \neg v_2)$. Each NOT may add at most one additional ϵ error to the given value, and the OR may add up to 3ϵ error (on top of the sum of the errors from both inputs). So the AND player will return a value within an additive $2\epsilon_l + 6\epsilon$ of the correct 0 or 1 answer. ∎

LEMMA 3.6. *Each of the arithmetic gadgets plays itself within* $\pm 5\epsilon$ *of the correct value for the input it is given.*

Proof. SUM: node S_1 will play S_1 with weight $w(S_1T) \in [\max(0, 1 - v_1' - v_2' - 2\epsilon), \max(0, 1 - v_1' - v_2' + 2\epsilon)]$. So node S will play S with weight $w_S(S) \in [v_1' + v_2' - 3\epsilon, v_1' + v_2' + 3\epsilon]$ unless $w_{S_1}(S_1) = 0$, which means $v_1' + v_2' \geq 1 - 2\epsilon$. In this case, node S will play S with weight at least $1 - \epsilon$. DIFF: node D_1 will play D_1T with weight $w_{D_1}(D_1) \in \max(0, 1 - v_1' - \epsilon), \max(0, 1 - v_1' + \epsilon)]$. Node D will play D with weight $w_D(D) \in [\max(0, v_1' - v_2' - 3\epsilon), \max(0, v_1' - v_2' + 3\epsilon)]$ unless $w_{D_1}(D_1) = 0$, which means $v_1' \geq 1 - \epsilon$. In this case, node D will play D with weight at least $1 - v_2' - 2\epsilon$ and at most $1 - v_2' + \epsilon$ (not 2ϵ because we cannot underfill the strategy with weight 0). COPY: node C_1 will play C_1 with weight at least $1 - v_1' - \epsilon$ and at most $1 - v_1' + \epsilon$. Node C will play C with weight at least $v_1' - 2\epsilon$ and at most $v_1' + 2\epsilon$. HALF: node H_1 will play H_1 with weight $w_{H_1}(H_1) \in [1 - v_1' - \epsilon, 1 - v_1' + \epsilon]$, and each other player will play its second and third preferences with total weight between $1 - w_{H_1}(H_1) - \epsilon$ and $1 - w_{H_1}(H_1) + \epsilon$. Every other player will play itself half of this amount plus or minus 3ϵ (this is easy to verify by writing the system of inequalities and checking the extreme points). Therefore, node H plays H with weight at least $\frac{v_1'}{2} - 4\epsilon$ and at most $\frac{v_1'}{2} + 4\epsilon$. DOUBLE: the DOUBLE gadget consists of a copy player, which adds at most 2ϵ error, and a sum player, which adds at most 3ϵ error on top of the sum of the errors in the two inputs. Therefore, node M plays M with weight at least $2v_1' - 5\epsilon$ and at most $2v_1' + 5\epsilon$. ∎

LEMMA 3.7. *The* LESS *player will play itself with weight* $< \epsilon_l$ *if it is given* v_1', v_2' *such that* $v_1' \leq v_2'$*, and with weight* $> 1 - \epsilon_l$ *if* $v_1' - v_2' \geq \epsilon_l$.

Proof. LESS: the LESS gadget inherits its susceptibility to error from its initial DIFF player (which was, in the exact equilibrium case, nonzero if and only if $v_1 < v_2$). For the case where $v_1 < v_2$, we can account for the errors of the DOUBLE players (used to repeatedly amplify the difference) simply by adding extra iterations of DOUBLE. Since we stipulated that $\epsilon \leq \epsilon_l^3$, a value that started $\leq 5\epsilon$ will remain $< \epsilon_l$, even after doubling enough times to push a value $\geq \epsilon_l$ to a value over 1 (including extra multiplications to account for the DOUBLE errors). Therefore, the LESS player will play itself with weight less than ϵ_l if $v_1' \leq v_2'$, and with weight greater than $1 - \epsilon_l$ if $v_1' - v_2' \geq \epsilon_l$. ∎

LEMMA 3.8. *By using a* LESS *gadget after each Boolean logic gadget, we can ensure that the output from each gate is at most* ϵ_l *away from the correct output.*

Proof. After a single gate (if the inputs are within additive ϵ_l of the correct 0 or 1 inputs), a player will play itself at least $1 - 2\epsilon_l - 6\epsilon$ if the correct answer is 1, and at most $2\epsilon_l + 6\epsilon$ if the correct answer is 0 (based on the analysis in the proof of Lemma 3.5). Call this player OUTPUT and the value it plays itself v. Then, we need only add a player CONSTANT-HALF that plays itself with weight close to $\frac{1}{2}$, and a LESS player, CORRECTION = LESS(OUTPUT, CONSTANT-HALF).

CONSTANT-HALF can be made up of a player that plays itself with weight at least $1 - \epsilon$ and at most 1 (its first preference is for itself) and a HALF player, who by Lemma 3.6 will play itself with weight at least $\frac{1-\epsilon}{2} - 5\epsilon$ and at most $\frac{1}{2} + 5\epsilon$.

We know that if the correct answer was 0, then $v \leq 2\epsilon_l + 6\epsilon < \frac{1-\epsilon}{2} - 5\epsilon$, so

CORRECTION will play itself with weight $< \epsilon_l$ (by Lemma 3.7), and if the correct answer was 1, then $v \geq 1 - 2\epsilon_l - 6\epsilon > \frac{1}{2} + 5\epsilon + \epsilon_l$, so CORRECTION will play itself with weight $> 1 - \epsilon_l$ (again by Lemma 3.7). ∎

After the corrections, we're left with the following possible errors due to the ϵ-approximation. We have small errors in the bit extraction, which are no larger than the parallel errors in [13] (they verify that these small error values will not affect the final result). We also have small errors (at most ϵ_l) coming out of the circuit. As in [8, 9], we will repeat the circuit a polynomial number of times and take the average in order to override any errors from the LESS gadgets in the bit extraction.

Taking an average of two results requires three steps: first we divide each "bit" in half (we cannot take the average of the entire values because we have a max value of 1 for any player, so the average of two 1's would come out to $\frac{1}{2}$). Here, we may pick up 4ϵ of error for each of the two results. Then, we sum the two. The total error so far is at most 11ϵ. Finally, we take half of the sum, which also divides the error in half but may add up to an additional 4ϵ of error, for a total additional error of at most 9.5ϵ from taking the average of two results.

We can add LESS gadgets periodically during the averaging and during the final OR, AND, and NOT of the results to keep our total errors under ϵ_l. In other words, if this is a solution vertex for Brouwer, then we will have six players, each playing at most ϵ_l. If this is not a solution vertex, then at least one of the six players will play at least $1 - \epsilon_l$.

Suppose we have an ϵ-equilibrium in this game, and the x-coordinate player is playing value x. This is a SUM player, and the extra player from the SUM gadget must be playing between $1 - x - \epsilon$ and $1 - x + \epsilon$. Therefore, the sum of the two values it is adding (a copy of the coordinate player and the feedback NOT player) must be between $x - 3\epsilon$ (if this player overfills each of its top strategies by ϵ) and $x + 3\epsilon$ (if this player underfills each of its top strategies by ϵ). We know that the copy player must be playing the same value as the coordinate player to within 2ϵ (between $x - 2\epsilon$ and $x + 2\epsilon$). Adding this range to a number $\geq 1 - \epsilon_l$ cannot possibly give something in the range $[x - 3\epsilon, x + 3\epsilon]$, so the feedback player must be playing a value at most ϵ_l on itself (since we know the feedback player will play either a value $\leq \epsilon_l$ or a value $\geq 1 - \epsilon_l$), and the correct feedback must be 0, so this is a valid fixed point. ∎

3.6. Constant degree preference games. For a given preference game, define $\mathrm{in}(v)$ (resp., $\mathrm{out}(v)$) of a player v to be the set $\{u : v \succ_u u\}$ (resp., $\{u : v \prec_v u\}$). Thus, $\mathrm{in}(v)$ is the set of players that v prefers strictly more than itself, while $\mathrm{out}(v)$ is the set of nodes that prefer v strictly more than themselves. We define the in-degree (resp., out-degree) of a player v to be $|\mathrm{in}(v)|$ (resp., $|\mathrm{out}(v)|$). The degree of the player is defined to be the sum of the in-degree and the out-degree of the player. The in-degree (resp., out-degree, degree) of the preference game is defined to be the maximum, over all nodes, of the in-degree (resp., out-degree, degree) of the node. Note that this is the same as the degree in a directed graph in which each player is represented by a node, and an edge from u to v means that u prefers v over itself. Degree d Preference Game is the problem of finding an equilibrium in a preference game with constant degree d.

We note that the players defined in section 3.4 all have out-degree at most 2. There is no implicit constant bound on the in-degree, but by adding COPY gadgets (which have out-degree 1) one can guarantee that the in-degree is at most 2. Furthermore, since COPY gadgets have out-degree 1, we can ensure that the overall degree of the preference game is at most 3. We formalize this argument in the following theorem, which presents a more direct reduction from general preference games to constant-degree preference games. This automatically implies that it is **PPAD**-hard to find an equilibrium even in a preference game with degree 3. We will use this fact in later sections, where we show **PPAD**-hardness of several other problems via reductions from constant-degree preference games.

THEOREM 3.9. PREFERENCE GAME $\leq_P$ DEGREE 3 PREFERENCE GAME. *In fact, there is a polynomial-time reduction from general preference games to preference games with degree* 3*, out-degree* 2*, and in-degree* 2*.*

Proof. Given a preference game over player set $[n] = \{1, \ldots, n\}$, with the sum of the lengths of the preference lists equal to m, assume that each player exists in the preference list (ahead of "self") for at most m' other players. We present our reduction in two steps: first, a reduction to preference games with out-degree at most 2, and then to preference games with degree at most 3, out-degree at most 2, and in-degree at most 2.

Reducing to a preference game with out-degree 2 *and* $O(m+n)$ *players.* Suppose player i in the original game has preference list $j_1, j_2, \ldots, j_k$. Let $d = \lceil \frac{k}{2} \rceil$. Create $2d$ new players, split into two sets: $I = \{i_1, \ldots, i_d\}$, $I' = \{i'_1, \ldots, i'_d\}$. For ease of notation, we will also refer to player i_d as i^*, since this is the player that will play itself the same amount that the original player i should play itself.

Set the preference list for the new player i_1 to j^*_1, j^*_2, i_1. For new player i_h ($h > 1$), set the preference list to $i'_{h-1}, j^*_{2h-1}, j^*_{2h}, i_h$. For each new player i'_h ($h \geq 1$), set the preference list to i_h, i'_h.

We now show that every equilibrium in the original preference game maps, in polynomial time, to an equilibrium in the new out-degree-2 preference game, and vice versa. For the purpose of the reduction, it is sufficient to establish the latter mapping (from an equilibrium in the new game to an equilibrium in the original game); we present both directions to show the tight connection between the two problems.

Equilibrium in the original preference game maps to an equilibrium in the new preference game. The map will be as follows: suppose we are given weights $w(i,j)$ for the original game, where $w(i,j)$ is the weight player i puts on player j. We will set the weights w^* in the new preference game as follows. Again, assume the preference list for player i in the original game is $j_1, j_2, \ldots, j_k$.

- $w^*(i_h, j^*) = w(i,j)$ for all j^* in the preference list of i_h.
- $w^*(i_h, i'_{h-1}) = \sum_{l=1}^{2(h-1)} w(i, j_l)$.
- $w^*(i_h, i_h) = 1 - \sum_{l=1}^{2h} w(i, j_l)$.
- $w^*(i'_h, i_h) = 1 - \sum_{l=1}^{2h} w(i, j_l)$.
- $w^*(i'_h, i'_h) = \sum_{l=1}^{2h} w(i, j_l)$.

Note that

$$\begin{aligned}
w^*(i^*, i^*) &= w^*(i_d, i_d) \text{ (by definition of } i^*) \\
&= 1 - \sum_{l=1}^{2d} w(i, j_l) \text{ (from map above)} \\
&= 1 - \sum_{l=1}^{\lceil \frac{k}{2} \rceil 2} w(i, j_l) \text{ (by definition of } d) \\
&= 1 - \sum_{l=1}^{k} w(i, j_l) \text{ (we can ignore the } \lceil\rceil \text{ since the pref list stops after } k \text{ items)} \\
&= w(i, i).
\end{aligned}$$

In order to verify that this is an equilibrium in the new game, we must check the following:

1. $w^*(i,j) \le w^*(j,j)$ for all i,j.
2. $w^*(i,i) + \sum_{j \neq i} w^*(i,j) = 1$ for all i.
3. If $w^*(i,j) > 0$, and if i prefers a over j, then $w^*(i,a) = w^*(a,a)$.

All three of these are trivial for players in I', so we will verify the conditions for players in I. First consider condition 1 for each weight placed by a player in set I.

- $w^*(i_h, a^*) = 0$ unless a^* is in the preference list for i_h. If a^* is in the preference list, then $a^* = a'_p$ for some player a from the original game with p being $\lceil$length of a's preference list/2$\rceil$, and $w^*(a^*, a^*) = w(a,a)$. By the map above, $w^*(i_h, a^*) = w(i,a)$. Since $w(i,a) \le w(a,a)$, $w^*(i_h, a^*)$ obeys condition 1.
- $w^*(i_h, i'_{h-1}) = \sum_{l=1}^{2(h-1)} w(i, j_l)$. But we know from the map that $w^*(i'_{h-1}, i'_{h-1}) = \sum_{l=1}^{2(h-1)} w(i, j_l)$, so $w^*(i_h, i'_{h-1})$ also obeys condition 1.

Next, check condition 2 for each player in set I. The total weight placed by player i_h is

$$\begin{aligned}
&w^*(i_h, i_h) + w^*(i_h, i'_{h-1}) + \sum_{j^* \text{ in the pref list of } i_h} w^*(i_h, j^*) \\
&= 1 - \sum_{l=1}^{2h} w(i, j_l) + \sum_{l=1}^{2(h-1)} w(i, j_l) + \sum_{j^* \text{ in the pref list of } i_h} w(i,j) \\
&= 1 - (w(i, j_{2h-1}) + w(i, j_{2h})) + (w(i, j_{2h-1}) + w(i, j_{2h})) \\
&= 1.
\end{aligned}$$

Finally, check condition 3. From above, we know that $w^*(i_h, i'_{h-1}) = w^*(i'_{h-1}, i'_{h-1})$, so the first element in each preference list in the new game (the first preference of i_h is for i'_{h-1}) will always obey $w^*(i,a) = w^*(a,a)$. Also from above, $w^*(j^*, j^*) = w(j,j)$ and $w^*(i_h, j^*) = w(i,j)$. Therefore, if any lower preference disobeys $w^*(i,a) = w^*(a,a)$, then it must also be true that $w(i,a) \neq w(a,a)$. Since we assumed the w values were an equilibrium in the original game, this must mean that there is no b preferred less than a with $w(i,b) > 0$, so for all b^* preferred less than a^*, $w^*(i_h, b^*) = 0$.

Equilibrium in the new preference game maps to an equilibrium in the original preference game. This map is simple. Given weights w^* in the new preference game, create weights w in the original preference game as follows:

- $w(i,j) = \max_h w^*(i_h, j^*)$.
- $w(i,i) = w^*(i^*, i^*)$.

The max in the first rule is a notational shortcut, since only one of the i_h players will have any preference for j^*, and therefore at most one of the i_h players will have $w^*(i_h, j^*) > 0$.

As before, we need to show the following to verify that this is an equilibrium in the original game:

1. $w(i,j) \le w(j,j)$ for all i,j.
2. $w(i,i) + \sum_{j \neq i} w(i,j) = 1$ for all i.
3. If $w(i,j) > 0$, and if i prefers a over j, then $w(i,a) = w(a,a)$.

To show condition 1, consider players i and j. Let $i_h =$ the player in the new game that has j^* in its preference list. Now, $w(i,j) = w^*(i_h, j^*)$ and $w(j,j) = w^*(j^*, j^*)$. Since w^* was feasible, we know that $w^*(i_j, j^*) \le w^*(j^*, j^*)$, as desired.

Next, to show condition 2, consider player i. $w(i,i) + \sum_{x=1}^{k} w(i, j_x) = w^*(i^*, i^*) + \sum_{x=1}^{k} \max_h w^*(i_h, j_x^*)$. Let's compute $w^*(i^*, i^*)$ in the new preference game. Recall that the preference list for player i_h is i'_{h-1}, j^*_{2h-1}, j^*_{2h}, i_h, and, specifically, the preference list for i^* $(= i_d)$ is i'_{d-1}, j^*_{2d-1}, j^*_{2d}, i_d. Thus,

$$
\begin{aligned}
w^*(i^*, i^*) &= 1 - w^*(i_d, i'_{d-1}) - \sum_{x=2d-1}^{k} w^*(i_d, j^*_x) \\
&= 1 - w^*(i'_{d-1}, i'_{d-1}) - \sum_{x=2d-1}^{k} \max_h w^*(i_h, j^*_x) \\
&= 1 - [1 - w^*(i'_{d-1}, i_{d-1})] - \sum_{x=2d-1}^{k} \max_h w^*(i_h, j^*_x) \\
&= w^*(i'_{d-1}, i_{d-1}) - \sum_{x=2d-1}^{k} \max_h w^*(i_h, j^*_x) \\
&= w^*(i_{d-1}, i_{d-1}) - \sum_{x=2d-1}^{k} \max_h w^*(i_h, j^*_x) \\
&= 1 - \left[w^*(i_{d-1}, i'_{d-2}) + \sum_{x=2d-3}^{2(d-1)} w^*(i_{d-1}, j^*_x) \right] - \sum_{x=2d-1}^{k} \max_h w^*(i_h, j^*_x) \\
&= 1 - w^*(i'_{d-2}, i'_{d-2}) - \sum_{x=2d-3}^{2(d-1)} \max_h w^*(i_h, j^*_x) - \sum_{x=2d-1}^{k} \max_h w^*(i_h, j^*_x) \\
&= 1 - w^*(i'_{d-2}, i'_{d-2}) - \sum_{x=2d-3}^{2c} \max_h w^*(i_h, j^*_x) \\
&= \cdots \\
&= 1 - w^*(i'_1, i'_1) - \sum_{x=c+1}^{k} \max_h w^*(i_h, j^*_x) \\
&= 1 - [1 - w^*(i'_1, i_1)] - \sum_{x=3}^{k} \max_h w^*(i_h, j^*_x) \\
&= w^*(i_1, i_1) - \sum_{x=3}^{k} \max_h w^*(i_h, j^*_x) \\
&= 1 - \sum_{x=1}^{2} w^*(i_1, j^*_x) - \sum_{x=3}^{k} \max_h w^*(i_h, j^*_x) \\
&= 1 - \sum_{x=1}^{2} \max_h w^*(i_h, j^*_x) - \sum_{x=3}^{k} \max_h w^*(i_h, j^*_x) \\
&= 1 - \sum_{x=1}^{k} \max_h w^*(i_h, j^*_x).
\end{aligned}
$$

Putting this back into our sum for player i, we get

$$
w(i, i) + \sum_{x=1}^{k} w(i, j_x) = w^*(i^*, i^*) + \sum_{x=1}^{k} \max_h w^*(i_h, j^*_x)
$$

$$= 1 - \sum_{x=1}^{k} \max_h w^*(i_h, j_x^*) + \sum_{x=1}^{k} \max_h w^*(i_h, j_x^*) = 1.$$

So condition 2 holds.

Finally, we need to verify that if $w(i,j) > 0$, and if i prefers a over j, then $w(i,a) = w(a,a)$. If $w(i,j) > 0$, then $\max_h w^*(i_h, j^*) > 0$. Let $h =$ the h that satisfies $\max_h w^*(i_h, j^*)$. Now, if i prefers a over j, then either (Case 1) i_h prefers a^* over j^* or (Case 2) there is some $b < h$ such that i_b has preference for a^*. Start with Case 1. Since we know that w^* is an equilibrium for the new preference game, it must be true that $w^*(i_h, a^*) = w^*(a^*, a^*)$, so $w(i,a) = w(a,a)$, as desired.

For Case 2, since $w^*(i_h, j^*) > 0$, we know that for all $b < h$, $w^*(i'_b, i'_b) < 1$. (Otherwise, for all $c > b$, i_c would put weight 1 on i'_{c-1}, leaving no weight left for itself, so i'_c would also be 1. Therefore, i_h would put weight 1 on i_{h-1}, leaving no weight for j^*.) Therefore, for the b with preference for a^*, $w^*(i'_b, i'_b) < 1 \Rightarrow w^*(i'_b, i_b) > 0 \Rightarrow w^*(i_b, i_b) > 0$. Therefore, for all c^* in the preference list for i_b (including a^*), $w^*(i_b, c^*) = w^*(c^*, c^*)$. So $w^*(i_b, a^*) = w^*(a^*, a^*)$, and $w(i,a) = w^*(i_b, a^*) = w^*(a^*, a^*) = w(a,a)$, as desired.

Reducing to a preference game with degree at most 3, *in-degree at most* 2, *and out-degree at most* 2, *and* $O(m'n)$ *players.* Following the preceding reduction, we have a preference game in which each player has out-degree at most 2. We now modify the game so that every node has in-degree at most 2, out-degree at most 2, and degree at most 3. Suppose we have a player i with in-degree at least 2; i.e., i exists in the preference lists of m' other players for $m' \geq 2$: $j_1, j_2, \ldots, j_{m'}$. We will add extra players $i'_1, i_1, i'_2, i_2, \ldots, i'_{m'}, i_{m'}$. The preference lists for these new players will be as follows: i'_1 has list (i, i'_1); for all $k > 1$, i'_k has list (i_{k-1}, i'_k); for all k, i_k has list (i'_k, i_k). If i plays itself with weight v, then i'_k will play itself with weight $1 - v$ and i_k will play itself with weight v. Each of these new players has in-degree 1 and out-degree 1. Now we can replace i with i_k in the preference list for j_k so that i now has in-degree 1 and each i_k has in-degree 2. Note that each i_k has out-degree at most 1, for a total degree of at most 3. Furthermore, this construction does not affect the degree of any j_k. It is easy to see that all the equilibria remain exactly the same. Repeating this step for every node with in-degree at least 2, we obtain the desired preference game. ∎

4. Personalized equilibria.

In this section, we introduce a new notion of equilibrium for matrix games, in which a player may individually match its strategies to its opponents' strategies without obeying a product distribution. Since this equilibrium allows different players to simultaneously choose different matchings across the strategies, we call this a *personalized equilibrium*. In section 4.2, we characterize the set of all personalized equilibria in a k-player game. In section 4.3, we show that finding a personalized equilibrium is **PPAD**-complete.

Suppose we are given a k-player matrix game between players $1, \ldots, k$. Each player i has strategy set S_i. We are also given a utility function for each i specified by $u_i : E \to \mathbb{R}$, where $E = \prod_j S_j$. Now, given probability distributions $p_j(S_j)$ for each $j \neq i$, a best response for player i (when using traditional Nash payoffs) is defined by the $p_i(S_i)$ that satisfies the following, where w is a weight function over $e \in E$:

$$\max \sum_{e \in E} w(e) u_i(e) \quad \text{such that}$$

$$
\begin{aligned}
w(e) &= \prod_{s \in e \cap S_j} p_j(s) \quad \forall e \in E, \\
\sum_{s \in S_i} p_i(s) &= 1, \\
w(e) &\geq 0 \quad \forall e \in E.
\end{aligned}
$$

The correlator in a correlated equilibrium [3] relaxes the requirement that w be a product distribution; however, w does satisfy, among other conditions, the projection constraint $\sum_{e:s\in e} w(e) = p_j(s)$ for all $s \in S_j, 1 \leq j \leq k$. For a personalized equilibrium, we further relax this by allowing each player to define its own weight function, w_i, so that in the best response of player i, $p_i(s)$ (and $w_i(e)$) satisfy the following:

$$
\begin{aligned}
&\max \sum_{e \in E} w_i(e) u_i(e) \quad \text{such that} \\
&\sum_{e:s\in e} w_i(e) = p_j(s), \quad s \in S_j,\ 1 \leq j \leq k, \\
&\sum_{s \in S_i} p_i(s) = 1, \\
&w_i(e) \geq 0, \quad e \in E.
\end{aligned}
$$

We can view a matrix game as a hypergraph with nodes $V = \cup_j S_j$ and edges $E = \prod_j S_j$. Then, if we interpret the $p_j(s)$ values as capacities on the nodes and the utility function for player i as weights on the edges from the perspective of player i, a personalized equilibrium is simultaneously a maximum-weight fractional hypergraph matching for each player. In the context of the clothing business example given in section 1, we have two players, a "shirt" player s with two strategies (wool/cotton) and a "pant" player p with two strategies (striped/solid). The probability variables have the natural meaning; for instance, p_s(wool) is the fraction of shirts produced by the shirt player that are wool. Each player's edges are the complete outfits. Finally, the weights give the proportion of a certain kind of outfit being created; for example, w_s(wool-striped) is the probability that an outfit received by the shirt player consists of a wool shirt and striped pant. A personalized equilibrium determines the proportions of wool/cotton shirts and striped/solid pants that each player produces so that the proportions of outfits they receive maximize their individual payoffs, subject to the constraint imposed by the other player.

The description of the game above is exponential in the number of players since we require that every edge connects one strategy of each player. To allow for more succinct descriptions, we generalize the game as follows. For each player i, we introduce a hypergraph with nodes $V = \cup_j S_j$ and edges E_i. The set E_i is required to satisfy two conditions (that are satisfied by E): (i) for each e in E_i and player j, e contains at most one element of S_j; (ii) there do not exist distinct e and e' in E_i such that $e \subset e'$. In the game, player i places a weight $w_i(e)$ on each edge in E_i. A player must still place a total of weight 1 on all her edges, and all weights must be nonnegative. Since the edges of E_i may not connect all players, however, we relax the projection constraint to $\sum_{e:s\in e} w_i(e) \leq p_j(s)$. Thus, the collection of weights $w_i(e)$, $e \in E_i$, and probability distributions $p_i(s)$, $s \in S_i$, over all players i, forms a personalized equilibrium if for each i, $w_i(e)$ and $p_i(s)$ maximize $\sum_{e\in E_i} w_i(e)u_i(e)$ subject to constraints captured by the following LP P_i determined by given $p_j(s)$ values (for all

$j \neq i$ and $s \in S_j$):

$$\begin{aligned} \sum_{e:s\in e} w_i(e) &\leq p_j(s) \quad \forall s \in S_j,\ \forall j \neq i, \\ \sum_{e:s\in e} w_i(e) &= p_i(s) \quad \forall s \in S_i, \\ \sum_{s\in S_i} p_i(s) &= 1, \\ w_i(e) &\geq 0 \quad \forall e \in E_i. \end{aligned} \tag{4.1}$$

For the notion of personalized equilibrium to be meaningful, we need to place one technical condition on the game. We say that a multiplayer game is *well behaved* if, for each player i, for any given set of $p_j(s)$ values (for all $j \neq i$ and $s \in S_j$) such that $\sum_{s\in S_j} p_j(s) = 1$ for all $j \neq i$, LP P_i is feasible. As we show below, matrix games where, for every player i, $E_i = E = \prod_j S_j$, as well as a natural graphical variant, are well behaved.

> PERSONALIZED EQUILIBRIUM: Given a well-behaved game with players $1, \dots, k$, strategy set S_i, edge set E_i, and utility function $u_i : E_i \to \mathbb{R}$ for each player i, find a probability distribution $p_i : S_i \to \mathbb{R}$ and a weight assignment $w_i : E_i \to \mathbb{R}$ for each player i that obeys the constraints of LP (4.1) and maximizes $\sum_{e\in E_i} w_i(e)u_i(e)$.

We define PERSONALIZED EQUILIBRIUM as the problem of finding a personalized equilibrium in a well-behaved game. Just as mixed Nash equilibria exist for every matrix game, we show that every well-behaved game thus defined has a personalized equilibrium.

THEOREM 4.1. *For every multiplayer well-behaved game, a personalized equilibrium always exists.*

Proof. Given the matrix game G, we construct the k-player game $\mathcal{G}$ in which the ith player's strategy space is the set of all probability distribution functions over S_i and the payoff is given by the personalized payoff function defined above. We can view the strategy space as the set of probability distribution functions over S_i instead of weight assignments to E_i since a weight assignment uniquely defines a probability distribution function, and since the payoffs and responses of the other players depend only on the $p_i(s)$ values, not on the $w_i(e)$ values. Then a personalized equilibrium of G is equivalent to a Nash equilibrium of $\mathcal{G}$. By [37, Proposition 20.3], a game has a pure Nash equilibrium if the strategy space of each player is a compact, nonempty, convex space, and the payoff function of each player is continuous on the strategy space of all players and quasi-concave in the strategy space of the player. The set of probability distributions over S_i that satisfies the LP (4.1) is clearly nonempty, convex, and compact. (The nonemptiness follows from the well-behavedness condition.) Furthermore, given probability distributions p_i over S_i, $1 \leq i \leq k$, the payoff for any player i is simply the solution to the following LP with variables $w_i(e)$, over $e \in E_i$:

$$\begin{aligned} \max \sum_{e\in E_i} w_i(e)u_i(e) \quad &\text{such that} \\ \sum_{e\in E_i:s\in e} w_i(e) \leq p_j(s), \quad &s \in S_j,\ 1 \leq j \leq k, \end{aligned}$$

$$\sum_{e \in E_i} w_i(e) = 1,$$
$$w_i(e) \geq 0, \quad e \in E.$$

It is easy to see that the payoff function is both continuous in the probability distributions of all players and quasi-concave in the strategy space of player i, thus completing the proof of the theorem. □

We define k-Personalized Equilibrium as a matrix game with k players, where for every player i, $E_i = E = \prod_j S_j$. Note that the traditional definition of a graphical game [24] may be used in this setting with a smaller set of edges. As for preference games, we define the out-degree (resp., in-degree) of a player i as $|N_i|$ (resp., the number of j such that $i \in N_i$). The out-degree (resp., in-degree) of the game is the maximum out-degree (resp., in-degree) of a player in the game. In d-Graphical Personalized Equilibrium, each player i has a neighborhood N_i of at most d other players, and all edges defined for player i are in $\prod_{j \in N_i} S_j$.

Corollary 4.2. *Every k-Personalized Equilibrium and d-Graphical Personalized Equilibrium instance has a personalized equilibrium.*

Proof. It is sufficient to show that both games are well behaved. In each kind of game, E_i equals $\prod_{j \in N_i} S_j$ for some set N_i of players; in the matrix game, N_i is in fact the set of all players. Consider LP (4.1) for a given set of $p_j(s)$ values (for all $j \neq i$ and $s \in S_j$) such that $\sum_{s \in S_j} p_j(s) = 1$ for all $j \neq i$. We present a procedure for finding a feasible solution to the program. We always maintain $p_i(s) = \sum_{e:s \in e} w_i(e)$ for all $s \in S_i$ and $\sum_{s \in S_i} p_i(s) = \sum_{e \in E_i} w_i(e) \leq 1$. We initially set $w_i(e) = 0$ for $e \in E_i$. Throughout the procedure, we ensure that the first two constraints and the nonnegativity constraint of (4.1) are always maintained.

We repeat the following steps: (a) identify for each player $j \neq i$ a strategy s_j such that $\sum_{e:s_j \in e} w_i(e) < p_j(s_j)$; if there exists a player j for which the preceding condition does not hold, then we have $\sum_{s \in S_j} \sum_{e:s_j \in e} w_i(e) = \sum_{s \in S_j} p_j = 1$, and we are done; (b) increase $w_i(e)$ for $e = (s_i, s_{-j})$, where s_i is an arbitrary strategy of i, until one of the inequality constraints becomes tight. Clearly, at the end of the process, we have a feasible solution to the desired program, establishing that the two kinds of games are well behaved. Theorem 4.1 guarantees the existence of personalized equilibria. □

Finally, we define ϵ-Approximate Personalized Equilibrium as the problem of finding a set of weight assignments ($w_i(e) \geq 0$ is the weight assigned by player i to edge e) such that (a) for every player i, $\sum_e w_i(e) = 1$; (b) for each player pair i and j, and for each strategy $s \in S_j$, $\sum_{e:s \in e} w_i(e) \leq \sum_{e:s \in e} w_j(e) + \epsilon$; and (c) for any player i, for any best response feasible weight assignment w_i^* against w_{-i}, $\sum_e w_i^*(e) u_i(e) - \sum_e w_i(e) u_i(e) \leq \epsilon$. As before, we require that the game be well behaved. Condition (b) requires the solution to be approximately feasible, while condition (c) requires that each player use an approximate best response. Note that we require the best response assignment w_i^* to be feasible; that is, for each player $j \neq i$, and for each strategy $s \in S_j$, $\sum_{e:s \in e} w_i^*(e) \leq \sum_{e:s \in e} w_j(e)$.

4.1. Characterizing personalized equilibria in 2-player games. We can simplify the definition of personalized equilibria when discussing 2-player games. Consider a matrix game (R, C) between two players ROW and COLUMN, in which player ROW has strategies $r_1, r_2, \ldots, r_m$ and player COLUMN has strategies $c_1, c_2, \ldots, c_n$. $R \in \mathbb{R}^{m \times n}$ is the payoff matrix of ROW, and $C \in \mathbb{R}^{m \times n}$ is the payoff matrix of COLUMN.

Like a standard bimatrix game, if player ROW selects r_i and player COLUMN selects c_j, the payoff to ROW is $R[i,j]$ and the payoff to COLUMN is $C[i,j]$. Suppose ROW selects a distribution x among the strategies $\{r_1, r_2, \ldots, r_m\}$, and COLUMN selects a distribution y among $\{c_1, c_2, \ldots, c_n\}$. Unlike payoffs defined for mixed strategies, in which the payoff to ROW is $\sum_{i,j} x[i]y[j]R[i,j]$ and the payoff to COLUMN is $\sum_{i,j} x[i]y[j]C[i,j]$, we define the payoffs using flows. The payoffs to ROW and COLUMN are

$$\text{Payoff (ROW)} = \max_{u_{i,j}} \sum_{i,j} u_{i,j} R[i,j] \quad \text{subject to } \sum_j u_{i,j} = x[i] \quad \forall i \quad \text{and} \quad \sum_i u_{i,j} = y[j] \quad \forall j, \tag{4.2}$$

$$\text{Payoff (COLUMN)} = \max_{v_{i,j}} \sum_{i,j} v_{i,j} C[i,j] \quad \text{subject to } \sum_j v_{i,j} = x[i] \quad \forall i \quad \text{and} \quad \sum_i v_{i,j} = y[j] \quad \forall j. \tag{4.3}$$

In other words, Payoff (ROW) is the cost of a 1-unit minimum-cost flow from source r to destination c in the directed graph $G_R = (V, E_R)$, with

$$V = \{r, c, r_1, r_2, \ldots, r_m, c_1, c_2, \ldots, c_n\},$$
$$E_R = \{(r \to r_i)\ \forall i\} \cup \{(r_i \to c_j)\ \forall i,j\} \cup \{(c_j \to c)\ \forall j\},$$

where the capacity of edge $(r \to r_i)$ is $x[i]$, the capacity of edge $(c_j \to c)$ is $y[j]$, and the capacity of all other edges is $+\infty$. The cost of edge $(r_i \to c_j)$ is $-R[i,j]$, and the cost of all other edges is 0. We note that for any distributions x and y, a unit-flow from r to c always exists, so the above payoff function is well defined.

Similarly, Payoff (COLUMN) is the cost of a 1-unit minimum-cost flow from source c to destination r in the directed graph $G_C = (V, E_C)$, with

$$E_C = \{(c \to c_j)\ \forall j\} \cup \{(c_j \to r_i)\ \forall i,j\} \cup \{(r_i \to r)\ \forall i\},$$

where the capacity of edge $(c \to c_j)$ is $y[j]$, the capacity of edge $(r_i \to r)$ is $x[i]$, and the capacity of all other edges is $+\infty$. The cost of edge $(c_j \to r_i)$ is $-C[i,j]$, and the cost of all other edges is 0.

It is not hard to show that the set of all 2-player personalized equilibria is convex. In fact, we can give a stronger characterization, which will lead to a polynomial-time algorithm.

THEOREM 4.3. *A 2-player personalized equilibrium can always be found in polynomial time.*

Proof. Consider a bipartite graph $G' = (A, B, E')$ with A being the set $\{r_1, \ldots, r_n\}$, B being the set $\{c_1, \ldots, c_m\}$, and the edge set E' being defined as follows: $(r_i \to c_j) \in E_R$ is in E' if and only if $R[i,j] \geq R[i',j]$ for all i', and $(c_j \to r_i) \in E_C$ is in E' if and only if $C[i,j] \geq C[i,j']$ for all j'.

Any directed cycle in G' corresponds to a personalized equilibrium. Consider any cycle $\{r_{i1}, c_{j1}, r_{i2}, c_{j2}, \ldots, r_{il}, c_{il}\}$ in G', each node played with weight $\frac{1}{l}$. Player ROW can match each of its strategies r_{ik} with player COLUMN's strategy c_{jk}. Since this is a best response for player ROW, ROW cannot do better by changing to another strategy. Similarly, player COLUMN can match each of its strategies c_{jk} with player ROW's strategy $r_{i(k+1)}$ for $k < l$, and c_{jl} can be matched with r_{i1}.

Every personalized equilibrium is a linear combination of cycles in G'. Any personalized equilibrium corresponds to an assignment σ of weights to each edge of the bipartite graph G': player ROW assigns weights to the edges out of A, and player COLUMN assigns weights to the edges out of B; from the construction of G', it follows that any solution that assigns weights to edges outside E' is not a personalized equilibrium. Furthermore, by definition of an equilibrium, we obtain that for each node the sum of the weights on the edges coming into a node equals the sum of the weights of the edges going out of the node; we refer to this property as *weight symmetry*.

Let C be any cycle in G' such that all edges of the cycle have positive weight. Such a cycle can be found by starting from any node and repeatedly going out on an edge with positive weight until we loop back to an already visited node; this process is guaranteed to loop due to weight symmetry. Let $\alpha(C)$ denote the weight of the minimum-weight edge in C. We reduce the weight of each edge in C by $\alpha(C)$ to obtain a new weight assignment σ' that also has weight symmetry. Repeating this process until we reach an empty weight assignment gives us a set S of cycles such that the personalized equilibrium is a linear combination of all cycles in S with the coefficient of C being $\alpha(C)$. ∎

4.2. Characterizing personalized equilibria in k-player games. We have shown that the set of all personalized equilibria for a 2-player game is just the set of all linear combinations of cycles in an appropriately defined graph, which is easy to compute in polynomial time. However, for k-player games ($k > 3$), we will give a reduction from finding an equilibrium in a preference game to finding a personalized equilibrium in a k-player game (for $k > 3$), thereby showing that finding personalized equilibria is **PPAD**-hard. Nevertheless, we are able to give a concise characterization of the set of all personalized equilibria for arbitrary multiplayer games.

THEOREM 4.4 (personalized equilibrium characterization). *The following program* (4.4) *represents the set of all exact personalized equilibria. The variables are $w_i(e)$, the weight placed by player i on edge e for all $e \in E_i$.*

$$\begin{aligned}
&\sum_{e\in E_i:s\in e} w_i(e) \le \sum_{e\in E_j:s\in e} w_j(e), \quad s\in S_j,\ 1\le j,i\le k,\\
&\sum_{e\in E_i} w_i(e) = 1, \quad 1\le i\le k,\\
&w_i(e)\ge 0, \quad 1\le i\le k,\ e\in E_i,\\
&\min_{e\in F} w_i(e) = 0 \quad \forall \textit{ players } i \textit{ and subsets } F\subseteq E_i\\
&\qquad\qquad \textit{such that LP } (4.5) \textit{ is feasible.}
\end{aligned} \tag{4.4}$$

The following LP (4.5) *is defined for each player i and $F \subseteq E_i$ (referred to as an* improvement set*). The variables are $\delta(e)$ for each edge $e \in E_i$.*

$$\begin{aligned}
&\sum_{e\in E_i} \delta(e)u_i(e) > 0,\\
&\sum_{e\in E_i:s\in e} \delta(e) = 0, \quad s\in S_j,\ 1\le j\le k,\ j\ne i,\\
&\delta(e) < 0 \quad (e\in F),\\
&\delta(e) \ge 0 \quad (e\notin F).
\end{aligned} \tag{4.5}$$

Before formally proving this theorem, we will start with some intuition about why this characterizes all equilibria. The first two constraints of program (4.4) specify a feasible weight assignment, and the first two constraints of LP (4.5) specify feasible "weight changes" that would increase the payoff for player i. How do we know that checking this for all subsets of edges is enough to find any possible improvement, and how does the last constraint of program (4.4) ensure that no improvement is possible? We can think of the δ values found in any solution to LP (4.5) as an "improvement direction." This is a vector that is orthogonal to the vector of all 1's and has a positive dot product with the utilities of i. In other words, if player i were to move weight in this direction, its payoff would improve. Of course, there may be a continuum of such improvement directions. However, there are most an exponential number of negative supports, or "improvement sets." These are exactly the F values for which LP (4.5) is feasible. Given an improvement set, the associated player can get a higher payoff by removing weight from all of those edges and adding them instead to edges with positive δ value. This improvement will be possible unless the player does not have weight on this entire improvement set; that is, unless $\min_{e \in F} w_i(e) = 0$.

Proof. A solution to the program is an exact personalized equilibrium. Assume we have a solution to (4.4) that is not a personalized equilibrium. The first two constraints ensure that our solution is a feasible weight assignment for the game. Therefore, there must be some player i who is not playing a best response. Take some better response, in which player i plays weights $w_i^*(e)$, and let $\delta(e) = w_i^*(e) - w_i(e)$.

Let F be the subset of E_i such that $\delta(e) < 0$ (that is, player i puts more weight on each edge in F in the original response than in the best response). Since w_i^* has a strictly higher total utility for player i than w_i has, we know that $\sum_{e \in E_i} w_i^*(e)u_i(e) > \sum_{e \in E_i} w_i(e)u_i(e)$, which implies that $\sum_{e \in E_i} \delta(e)u_i(e) > 0$. Since both w_i and w_i^* were feasible weights, it must be true for any strategy s that $\sum_{e:s \in e} w_i(e) = \sum_{e:s \in e} w_i^*(e) \Rightarrow \sum_{e:s \in e} \delta(e) = 0$.

By our definition of F, $\delta(e) < 0$ for all $e \in F$ and $\delta(e) \geq 0$ for all $e \notin F$. Therefore, F and i obey all the constraints of LP (4.5), so since $w_i(e)$ obeyed program (4.4), we know that $\min_{e \in F} w_i(e) = 0$. Thus, there exists some edge $f \in F$ such that $w_i(f) = 0$. But then $0 > \delta(f) = w_i^*(f) - w_i(f) = w_i^*(f)$, contradicting the fact that w_i^* was a feasible best response for player i.

Any personalized equilibrium is a solution to the program. Assume we have a personalized equilibrium that does not satisfy some constraint of the program. Let $w_i(e)$ be the weight placed by player i on edge e in this equilibrium. The first three constraints are the definition of a feasible weight assignment. Therefore, assume this equilibrium does not satisfy the min constraint for some player i and some subset $F \subseteq E_i$ for which LP (4.5) is feasible.

Consider a solution δ for LP (4.5) for this i and F. Let $M = \frac{\min_{e \in F} |w_i(e)|}{\max_{e \in F} |\delta(e)|}$, and let $\delta'(e) = \delta(e) \cdot M$. M is a well-defined positive number since (1) the min constraint was not satisfied, and (2) LP (4.5) specifies that $\delta(e) < 0$ for all $e \in F$. We know that F is nonempty because the first constraint implies there is some $\delta > 0$, and combining this with the second constraint implies there is also some $\delta < 0$. Furthermore, for all e with $\delta(e) < 0$ (i.e., for all $e \in F$), $|\delta'(e)| \leq w_i(e)$. Now, consider the alternate assignment for player i specified by $w_i^*(e) = w_i(e) + \delta'(e)$. By the second constraint of LP 4.5 and the fact that $|\delta'(e)| \leq w_i(e)$ for all e with $\delta(e) < 0$, this is still a valid weight assignment. By the first constraint of LP (4.5), this gives a strictly higher total utility for player i. Therefore, weights $w_i(e)$ did not give a best response for player i, so we did not have a personalized equilibrium, contradicting our assumption and completing the proof. ∎

COROLLARY 4.5. *For any matrix game with all rational payoffs, there exists a personalized equilibrium in which the probability assigned by each player to each strategy is a rational number.*

Proof. In Theorem 4.4, we showed that any personalized equilibrium is a solution to an LP plus additional min constraints, in which all coefficients are rational. By Theorem 4.1, this program has at least one solution. Now, we can rewrite this as a union of many LPs as follows. Let $F_1, \ldots, F_\alpha$ be the set of all improvement sets. We can write $\prod_{i=1}^{\alpha} |F_i|$ LPs, each consisting of the first three constraints from program (4.4) as well as the α constraints $[w_1(e_1) = 0$ for some $e_1 \in F_1]$, $[w_2(e_2) = 0$ for some $e_2 \in F_2]$, …, $[w_\alpha(e_\alpha) = 0$ for some e_α in $F_\alpha]$. We can create one LP for each such combination of one edge from each improvement set, or $\prod_{i=1}^{\alpha} |F_i|$ LPs. Since the union of these LPs is exactly the same as the program in Theorem 4.4, and since (by Theorem 4.1) the program in Theorem 4.4 has at least one solution, we know that at least one of these LPs has a solution. Any feasible LP with rational coefficients will have a rational solution. Therefore, there will be a personalized equilibrium with all rational weights. ∎

4.3. Finding personalized equilibria is PPAD-complete. This section contains four reductions. First, we reduce DEGREE d PREFERENCE GAME to d-GRAPHICAL PERSONALIZED EQUILIBRIUM. We next reduce 3-GRAPHICAL PERSONALIZED EQUILIBRIUM to 4-PERSONALIZED EQUILIBRIUM. It can be easily verified that the same reductions can be used to show ϵ-APPROXIMATE PREFERENCE GAME $\leq_P$ ϵ-APPROXIMATE PERSONALIZED EQUILIBRIUM. These reductions together show that finding an ϵ-approximate personalized equilibrium in both graphical games and 4-player games is **PPAD**-hard.

THEOREM 4.6. PREFERENCE GAME $\leq_P$ PERSONALIZED EQUILIBRIUM.

Proof. Given: a preference game over player set $[n]$, with the preference lists specified as a set of values Q_{ij} for all $i, j \in [n]$, where Q_{ij} = the number of players k such that $j \succeq_i k \succeq_i i$.

Define a game as follows, in which we will find a personalized equilibrium:

- The set of players = $\{p_1, \ldots, p_n\}$.
- S_i (the set of strategies for player p_i) = $\{s_{ij} : Q_{ij} > 0\}$.
- H_i = the set of hyperedges for player p_i = $\{\{s_{ij}, s_{jj}\}$ for all $s_{ij} \in S_i$, $j \neq i\}$ $\cup \{s_{ii}\}$.
- $u_i(\{s_{ij}, s_{jj}\})$ (the payoff to player i for this hyperedge) = Q_{ij}.
- $u_i(\{s_{ii}\}) = Q_{ii} \geq 1$.

Notice that the degree of the game is preserved, and the number of edges defined is at most n times the degree. Furthermore, since every player i has a hyperedge consisting of the singleton set $\{s_{ii}\}$, a solution to LP (4.1) always exists, making the game well behaved.

We now show that every personalized equilibrium maps, in polynomial time, to an equilibrium in the preference game, and vice versa. For the purpose of the reduction, it is sufficient to establish the former mapping (from a personalized equilibrium to an equilibrium in the preference game); we present both directions to show the tight connection between the two problems.

A personalized equilibrium maps to an equilibrium in the preference game. The map will be as follows: suppose we are given weights x_{ij} for each player i and edge $\{s_{ij}, s_{jj}\}$, and x_{ii} for player i and hyperedge $\{s_{ii}\}$. These weights form a personalized equilibrium. We will set weights $w_{ij} = x_{ij}$ in the preference game.

To show that this is an equilibrium in the preference game, we must show the following:

- For all i, $\sum_j w_{ij} = 1$.
  $\sum_j w_{ij} = \sum_j x_{ij} = 1$, since this is a valid solution to the personalized game.
- For all i, j, $w_{ij} \leq w_{jj}$.
  $w_{ij} = x_{ij} \leq x_{jj}$ (by the projection constraint for personalized equilibria), and $x_{jj} = w_{jj}$.
- w is a lexicographically maximal weight assignment.
  Suppose this is not true. Then there exists another weight assignment w' that is lexicographically larger than w. Let w' be the lexicographically maximal such assignment. Thus, there exist i, j such that $\sum_{k:Q_{ik} \geq Q_{ij}} w'_{ij} > \sum_{k:Q_{ik} \geq Q_{ij}} w_{ij}$. For this i, take a j with the largest Q_{ij} that meets this condition. By our definition of j and the fact that w' is lexicographically maximal, we know that for all j' with $Q_{ij'} > Q_{ij}$, $\sum_{k:Q_{ik} \geq Q_{ij'}} w'_{ij} = \sum_{k:Q_{ik} \geq Q_{ij'}} w_{ij}$. Let $\delta = \sum_{k:Q_{ik} \geq Q_{ij}} w'_{ij} - \sum_{k:Q_{ik} \geq Q_{ij}} w_{ij} > 0$. Now consider the payoff to player i in the personalized game:

$$
\begin{aligned}
\text{Payoff to } i &= \sum_l x_{il} Q_{il} \\
&= \sum_{l:Q_{il} > Q_{ij}} x_{il} Q_{il} + \sum_{l:Q_{il} = Q_{ij}} x_{il} Q_{il} + \sum_{l:Q_{il} < Q_{ij}} x_{il} Q_{il} \\
&= \sum_{l:Q_{il} > Q_{ij}} w_{il} Q_{il} + Q_{ij} \sum_{l:Q_{il} = Q_{ij}} w_{il} + \sum_{l:Q_{il} < Q_{ij}} w_{il} Q_{il} \\
&= \sum_{l:Q_{il} > Q_{ij}} w'_{il} Q_{il} + Q_{ij} \sum_{l:Q_{il} = Q_{ij}} w_{il} + \sum_{l:Q_{il} < Q_{ij}} w_{il} Q_{il} \\
&= \sum_{l:Q_{il} > Q_{ij}} w'_{il} Q_{il} + Q_{ij} \left(\left(\sum_{l:Q_{il} = Q_{ij}} w'_{il} \right) - \delta \right) + \sum_{l:Q_{il} < Q_{ij}} w_{il} Q_{il} \\
&\leq \sum_{l:Q_{il} > Q_{ij}} w'_{il} Q_{il} + Q_{ij} \sum_{l:Q_{il} = Q_{ij}} w'_{il} - \delta Q_{ij} \\
&\quad + \sum_{l:Q_{il} < Q_{ij}} w'_{il} Q_{il} + \delta (Q_{ij} - 1) \\
&< \sum_{l:Q_{il} > Q_{ij}} w'_{il} Q_{il} + Q_{ij} \sum_{l:Q_{il} = Q_{ij}} w'_{il} - \delta (Q_{ij} - 1) \\
&\quad + \sum_{l:Q_{il} < Q_{ij}} w'_{il} Q_{il} + \delta (Q_{ij} - 1) \\
&= \sum_{l:Q_{il} > Q_{ij}} w'_{il} Q_{il} + Q_{ij} \sum_{l:Q_{il} = Q_{ij}} w'_{il} + \sum_{l:Q_{il} < Q_{ij}} w'_{il} Q_{il} \\
&= \sum_l w'_{il} Q_{il}.
\end{aligned}
$$

  So player i would do strictly better by playing $x = w'$, leading to a contradiction.

An equilibrium in the preference game maps to a personalized equilibrium. Suppose that we are given weights w_{ij} forming an equilibrium in the preference game.

We will set weights in the personalized game as follows. $x_{ih} = w_{ij}$ for player i and edge $h = \{s_{ij}, s_{jj}\}$. $x_{ih} = w_{ii}$ for player i and edge $h = \{s_{ii}\}$.

To show this is a personalized equilibrium, we must show the following:

- For all i, $\sum_h x_{ih} = 1$.
  $\sum_h x_{ih} = \sum_j w_{ij} = 1$, since this is a valid weight assignment in the preference game.
- For all $i, s_{jk} \in S_j$, $\sum_{h:s_{jk}\in h} x_{ih} \le \sum_{h:s_{jk}\in h} x_{jh}$.
  If $j \neq k$, $\sum_{h:s_{jk}\in h} x_{ih} = 0 \le \sum_{h:s_{jk}\in h} x_{ij}$. If $j = k$, $\sum_{h:s_{jj}\in h} x_{ih} = x_{ih'}$, where $h' = \{s_{ij}, s_{jj}\} = w_{ij} \le w_{jj} = x_{jh''}$, where $h'' = \{s_{jj}\} = \sum_{h:s_{jj}\in h} x_{jh}$.
- x is a best response in the personalized game for all players i.
  Consider any other weight function x' for the personalized game. Since there is a one-to-one mapping from defined edges to i, j pairs in the preference game (including $i = j$), we can define a new weight function w' in the preference game using the same rules as defined in the first half of this proof ($w'_{ij} = x'_{ij}$). We know that w is lexicographically maximal for the preference game. Using the same reasoning as above, we get the following:

$$\begin{aligned}
\text{Payoff to } i \text{ playing } x' &= \sum_l x'_{il} Q_{il} \\
&= \sum_{l:Q_{il}>Q_{ij}} x'_{il} Q_{il} + \sum_{l:Q_{il}=Q_{ij}} x'_{il} Q_{il} + \sum_{l:Q_{il}<Q_{ij}} x'_{il} Q_{il} \\
&= \sum_{l:Q_{il}>Q_{ij}} w'_{il} Q_{il} + Q_{ij} \sum_{l:Q_{il}=Q_{ij}} w'_{il} + \sum_{l:Q_{il}<Q_{ij}} w'_{il} Q_{il} \\
&< \sum_{l:Q_{il}>Q_{ij}} w_{il} Q_{il} + Q_{ij} \sum_{l:Q_{il}=Q_{ij}} w_{il} + \sum_{l:Q_{il}<Q_{ij}} w_{il} Q_{il} \\
&= \sum_l w_{il} Q_{il} \\
&= \text{Payoff to } i \text{ playing } x. \qquad \square
\end{aligned}$$

COROLLARY 4.7. *It is* **PPAD**-*hard to find a personalized equilibrium, even in a graphical game with out-degree and in-degree at most* 2 *and degree at most* 3.

Proof. The reduction in the proof of Theorem 4.6 preserves the number of players. Furthermore, the graphical game inherits the degree, the out-degree, and the in-degree of the given preference game. The desired claim follows since by Theorem 3.9, it is **PPAD**-hard to find an equilibrium in a preference game with out-degree and in-degree at most 2 and degree at most 3. □

THEOREM 4.8. *There is a polynomial-time reduction from* 3-GRAPHICAL PERSONALIZED EQUILIBRIUM *games with in-degree and out-degree at most* 2 *to* 4-PERSONALIZED EQUILIBRIUM.

Proof. Suppose we are given a graphical game $G = (V, E)$ with degree 3, out-degree 2, and in-degree 2, for which we want to find a personalized equilibrium. We aim to obtain a 3-coloring of the nodes such that (a) for any node u, the out-neighbors of u and u all have different colors, and (b) for any node u, the in-neighbors of u and u all have different colors. This cannot be guaranteed for G, so we convert G to a new graph G' that has the desired property yet maintains the equilibria of G.

We now present graph G', together with the desired 3-coloring. We set G' to G initially and insert new nodes and insert/delete edges as follows. For each edge (u, v), we add two new nodes, r_{uv} and s_{uv}, and replace edge (u, v) by the edges (u, r_{uv}),

(r_{uv}, s_{uv}), and (s_{uv}, v). Node s_{uv} is a *copy* of v: it has the same strategies as v and will play exactly the same weights as v in an equilibrium. We can do this by setting the payoff to s_{uv} for agreeing with v to 1, and the payoffs for disagreeing with v to 0. Node r_{uv} is a copy of s_{uv}.

Let R and S denote the sets $\{r_{uv} : (u, v) \in E\}$ and $\{s_{uv} : (u, v) \in E\}$, respectively. Each node in R and S has in-degree and out-degree exactly 1. We assign color 0 to all nodes in V (that is, the original set of nodes in G). Since all edges out of V go to a node in R, no two nodes in V have an edge to the same node. Similarly, all edges into V come from a node in S, so no two nodes in V have an edge from the same node. So the color assignment to the nodes in V does not introduce any violations. If a node u has only one outgoing edge (u, r_{uv}), we assign a color of 1 to r_{uv}; if u has two outgoing edges (u, r_{uv}) and (u, r_{uw}), we arbitrarily assign color 1 to either r_{uv} or r_{uw} and color 2 to the other. We similarly color the nodes in S. If a node v has only one incoming edge (s_{uv}, v), we assign a color of 1 to s_{uv}; if v has two incoming edges (s_{uv}, v) and (s_{wv}, v), we arbitrarily color either s_{uv} or s_{wv} 1 and the other 2. Finally, for any two nodes r_{uv} and s_{uv} that have the same color, we introduce a new node $t_{u,v}$ that is a copy of s_{uv}, replace edge (r_{uv}, s_{uv}) by two edges (r_{uv}, t_{uv}) and (t_{uv}, s_{uv}), and modify r_{uv} so that it is a copy of t_{uv} (instead of s_{uv}). If r_{uv} and s_{uv} both have color 2, then we color t_{uv} 1; otherwise we color it 2. Let T denote the set of nodes of the form t_{uv}. Let the set $C = R \cup S \cup T$.

We now observe that both the conditions of the 3-coloring are met by the above assignment. Any node in V and its out-neighbors all get different colors. Any node in V and its in-neighbors all get different colors. Finally, any node in C has in-degree and out-degree at most 1 and also satisfies the property that the node has a different color from its out-neighbor (resp., in-neighbor), if it exists. Furthermore, by construction any equilibrium in the new graphical game maps to an equilibrium in the original graphical game, and vice versa.

We now create the 4-player personalized game as follows. We create one player per color. Each player takes each of the strategies for each node in that color. For ease of notation, we assume in the following that each of the original nodes had only two strategies, labeled 0 and 1. This can be easily extended to handle more strategies. We add dummy strategies as necessary so that each of the three players has the same number of strategies, and the number of strategies for each player is even. We also add a fourth player with half the number of strategies of any other player.

This gives us four players. Let the strategies for player 1 be $\{a_{10}, a_{11}, a_{20}, a_{21}, \ldots, a_{k0}, a_{k1}\}$. The strategies for player 2 are $\{b_{10}, b_{11}, \ldots, b_{k0}, b_{k1}\}$. The strategies for player 3 are $\{c_{10}, c_{11}, \ldots, c_{k0}, c_{k1}\}$. The strategies for player 4 are $\{d_1, d_2, \ldots, d_k\}$.

Next we will assign initial payoffs for each hyperedge and player; we will revise these later. Let $(a_{xi}, b_{yj}, c_{z\ell}, d_m)$ be a hyperedge. If x has y and z as out-neighbors in G', then for this hyperedge, we assign the initial payoff for player 1 as the same as that obtained by x when it plays strategy i against strategy j of y and strategy ℓ of z; otherwise, we assign a payoff of 0 for player 1. Analogously, we assign appropriate initial payoffs to players 2 and 3. All of player 4's payoffs start at 0. Let $p_i(w, x, y, z)$ = the payoff to player i if player 1 plays w, player 2 plays x, player 3 plays y, player 4 plays z. Now we want to add to these payoffs in order to ensure that each player plays each strategy pair equally.

Let M be strictly greater than the largest payoff so far. Now, change the following payoffs:

$p_1(a_{si}, x, y, d_s) += M$ (player 1 is playing either strategy from the node numbered s; player 4 is playing his sth strategy).

$p_2(w, b_{si}, y, d_s)+ = M$ (player 2 is playing either strategy from the node numbered s; player 4 is playing his sth strategy).

$p_3(w, x, c_{si}, d_s)+ = M$ (player 3 is playing either strategy from the node numbered s; player 4 is playing his sth strategy).

$p_4(a_{si}, x, y, d_{(s+1)})+ = M$ (player 1 is playing either strategy from the node numbered s; player 4 is playing strategy $s + 1 \bmod k$).

If $f_i(x) =$ the amount player i plays strategy x, then in any equilibrium we must have (for all s)

$$\begin{aligned}
f_1(a_{s0}) + f_1(a_{s1}) &= f_4(d_s),\\
f_2(b_{s0}) + f_2(b_{s1}) &= f_4(d_s),\\
f_3(c_{s0}) + f_3(c_{s1}) &= f_4(d_s),\\
f_4(d_s) &= f_1(a_{(s-1)0}) + f_1(a_{(s-1)1}) \quad \text{for } 1 < s \le k,\\
f_4(d_1) &= f_1(a_{k0}) + f_1(a_{k1}).
\end{aligned}$$

These equations imply that

$$\begin{aligned}
f_1(a_{s0}) + f_1(a_{s1}) &= f_1(a_{(s-1)0}) + f_1(a_{(s-1)1}) \quad \text{for } s > 0,\\
f_1(a_{00}) + f_1(a_{01}) &= f_1(a_{k0}) + f_1(a_{k1}),\\
f_2(b_{s0}) + f_2(b_{s1}) &= f_1(a_{s0}) + f_1(a_{s1}) \quad \forall\ s,\\
f_3(c_{s0}) + f_3(c_{s1}) &= f_1(a_{s0}) + f_1(a_{s1}) \quad \forall\ s.
\end{aligned}$$

It thus follows that each player plays each strategy pair equally. Let F denote the amount $f_1(a_{00}) + f_1(a_{01})$. Within a strategy pair (say a_{s0} and a_{s1} for player 1), the distribution among the two strategies ($f_1(a_{s0})$ and $f_1(a_{s1})$) in any equilibrium needs to be a best response in the graphical game for player s given appropriate strategies for its out-neighbor(s) (using color(s) 1 and/or 2). Thus, given a personalized equilibrium in which $f_i(x)$ denotes the amount player i plays strategy x, we obtain a personalized equilibrium for the graphical game by having the sth node of color 0 (resp., 1 and 2) play strategy $j \in \{0, 1\}$ with amount $f_0(a_{sj})/F$ (resp., $f_1(b_{sj})$ and $f_2(c_{sj})$). ∎

The remaining two reductions will be used to show **PPAD**-membership of Personalized Equilibrium. We show how to reduce Personalized Equilibrium to ϵ-Approximate Personalized Equilibrium, as long as ϵ is sufficiently small. Finally, we reduce ϵ-Approximate Personalized Equilibrium to End of the Line, thereby completing the proof that finding personalized equilibria (as well as ϵ-approximate personalized equilibria) is **PPAD**-complete. We now present some lemmas on LP compactness and representations of solutions to LPs with rational coefficients, which will be useful in showing that Personalized Equilibrium $\le_P$ ϵ-Approximate Personalized Equilibrium. If a rational r can be written as a/b, where a is an integer that can be represented by at most α bits and b is an integer that can be represented by at most β bits, then we say that r can be represented by at most $\alpha + \beta$ bits.

Lemma 4.9. *If t, b, t_i, b_i are β-bit integers and y_i, z_i are γ-bit integers (for $1 \le i \le n$), then either $\sum_{i=1}^{n} \frac{t_i y_i}{b_i z_i} \ge \frac{t}{b}$ or $\sum_{i=1}^{n} \frac{t_i y_i}{b_i z_i} < \frac{t}{b} - \frac{1}{2^{(n+1)\beta + n\gamma}}$.*

Proof. Suppose we have $\sum_{i=1}^{n} \frac{t_i y_i}{b_i z_i} < \frac{t}{b}$. Then the difference $\frac{t}{b} - \sum_{i=1}^{n} \frac{t_i y_i}{b_i z_i}$ is at least $1/(b \cdot \prod_i b_i \prod_i z_i)$, which is at least $1/2^{\beta + n\beta + n\gamma}$ since b and all b_i's are at most 2^{β} each, and each z_i is at most 2^{γ}. ∎

Lemma 4.10. *Given an LP with n variables and coefficients of the form $\frac{a}{b}$ for integers a and b, each represented by at most β bits, each coordinate of a vertex solu-*

tion for the program must be representable by $\frac{c}{d}$ for integers c and d, each represented by at most $q_1(n,\beta) = n^2(\log n + \beta) + (n+1)\beta)$ bits.

Proof. Each vertex is the solution to a system $Ax = v$ of n linear equations over n variables (x) in which every entry of A and v is represented by a rational number, whose numerator and denominator are of size at most 2^β. By scaling, say by multiplying each side of an equation by the product of the denominators of the coefficients on the left-hand side, we can convert this system to one in which every entry of A is an integer of size at most $2^{n\beta}$ and every entry of v is an integer of size at most $2^{(n+1)\beta}$. For convenience, we continue to refer to this equivalent linear system as $Ax = v$.

Now, x is given by $A^{-1}v$. Each entry of A^{-1} is equal to the ratio of the determinant of an $(n-1)\times(n-1)$ matrix to the determinant of an $n \times n$ matrix. The former is no more than $n! \leq n^n$ times the largest integer in A and so has size at most $n^n 2^{n\beta}$. The same is true for the latter. Thus for any i, j, $(A^{-1})_{ij}$ can be represented by at most $n \log n + n\beta$ bits for both its numerator and denominator. Since each entry of v is of size at most $2^{(n+1)\beta}$, x_i, given by $(A^{-1})_i \cdot v$, has numerator of size at most $n(n^n 2^{n\beta})^n 2^{(n+1)\beta}$ and denominator of size at most $(n^n 2^{n\beta})^n 2^{(n+1)\beta}$. The number of bits needed in the representation of each is at most $n^2(\log n + \beta) + (n+1)\beta$, thus completing the proof of the lemma. □

LEMMA 4.11 (LP compactness). *If an LP with n variables and rational coefficients, each represented by at most β bits, is such that there is a point obeying each constraint to within $\frac{1}{2^{6n^4\beta}}$, then the LP is feasible.*

Proof. Let P be an LP given by $Ax \leq b$, which satisfies the conditions of the lemma yet is infeasible. Consider the LP P' in which we replace each constraint $a_i x \leq b_i$ by the constraint $a_i x \leq b_i + m$, for all i, and set the objective to the minimization of m. By the conditions of the lemma, there exists a solution to P' that achieves an objective of at most $\frac{1}{2^{6n^4\beta}}$; furthermore, since P is infeasible, the value of P' is positive. Therefore, P' is bounded and has an optimal value at a vertex of the polytope associated with P'.

Since every coefficient of P' can be represented by at most β bits, by Lemma 4.10 we obtain that every vertex can be represented by at most $\gamma = n^2(\log n+\beta)+(n+1)\beta$ bits. Thus, we have a point y which satisfies every inequality of P to within an error of $\frac{1}{2^{6n^4\beta}}$, and every coordinate of which is representable by at most $n^2(\log n + \beta) + (n+1)\beta$ bits. Fix a constraint $a_i x \leq b_i$ of P, and let ε equal $a_i y - b_i$. We are given that $\varepsilon \leq \frac{1}{2^{6n^4\beta}}$. We apply Lemma 4.9 with β and γ (of that lemma) both set to $n^2(\log n + \beta) + (n+1)\beta$ and obtain that ε is either at most 0 or greater than $1/2^{(2n+1)(n^2 \log n+\beta)+(n+1)\beta}$, which is at least $1/2^{6n^4\beta}$. Thus, ε is at most 0, and y satisfies the constraint. This applies for every constraint of P, and hence we obtain that P is feasible, yielding a contradiction. □

We use the next lemma to show that an approximate personalized equilibrium *almost* satisfies the constraints of Theorem 4.4. As long as ϵ is small enough, this will imply by Lemma 4.11 that an ϵ-approximate personalized equilibrium will point us to a feasible LP that finds an exact personalized equilibrium.

LEMMA 4.12. *An ϵ_1-approximate personalized equilibrium (with $\epsilon_1(|E|,\beta) = \frac{1}{2^{|E|^2(\log n+\beta)+(|E|+1)\beta}}$) will obey every constraint in program (4.4) to within $\epsilon_2 = \frac{1}{d}$ for β-bit integer d if each utility value is representable as $\frac{a}{b}$ for integers a and b of at most β bits each.*

Proof. Assume for the sake of contradiction that we have an ϵ_1-approximate personalized equilibrium that does not satisfy some constraint of the program to within ϵ_2.

Let $w_i(e)$ equal the weight placed by player i on hyperedge e in this approximate equilibrium. The first set of constraint of program (4.4) must be satisfied to within ϵ_1, while the equality and nonnegativity constraints need to be satisfied exactly since they are required in the definition of a feasible weight assignment. Therefore, assume that this equilibrium does not satisfy the min constraint to within ϵ_2 for some player i and some subset $F \subseteq E_i$ for which LP (4.5) is feasible.

Consider a solution δ^* for LP (4.5) for this i and F. Let $M = \frac{\epsilon_2}{\max_{e \in F} |\delta^*(e)|}$, and let $\delta'(e) = \delta^*(e) \cdot M$. M is well defined since $\delta^*(e) < 0$ for all $e \in F$. Furthermore, for all e with $\delta(e) < 0$ (i.e., for all $e \in F$), $|\delta'(e)| \leq \epsilon_2 \leq w_i(e)$.

Now consider the following slightly adjusted LP:

$$\begin{aligned}
&\text{maximize} \sum_{e \in E_i} \delta(e) u_i(e) \quad \text{such that} \\
&\sum_{e: s \in e} \delta(e) = 0, \quad s \in S_j,\ 1 \leq j \leq k,\ j \neq i, \\
&\delta(e) \geq -\epsilon_2 \quad (e \in F), \\
&\delta(e) \geq 0 \quad (e \notin F).
\end{aligned} \tag{4.6}$$

By our choice of δ' and the analysis above, δ' obeys each constraint of the new LP (4.6), and δ' gives a maximization value > 0. LP (4.6) has $|E_i| \leq |E|$ variables, each coefficient $u_i(e)$ can be represented as $\frac{a}{b}$ for β-bit integers a and b, and coefficient $\epsilon_2 = \frac{1}{d}$ for β-bit integer d. The maximization point of an LP will be at a vertex, so by Lemma 4.10, each dimension of the maximization point of LP (4.6) will be representable by $\frac{c}{b}$ for integers c and d, each of size at most $|E|^2(\log n+\beta)+(|E|+1)\beta$ bits.

Let δ equal the solution to LP (4.6), and consider the alternate assignment for player i specified by $w_i^*(e) = w_i(e) + \delta(e)$. By the first constraint of LP (4.6), player i still places a total weight of 1 on all of the edges. By the second and third constraints, $\delta(e) < 0$ if and only if $e \in F$, in which case $|\delta(e)| \leq \epsilon_2$ (when $w_i(e) \geq \epsilon_2$), so $w_i^*(e) \geq 0$. In other words, we still have a valid weight assignment for an ϵ_1-approximate personalized equilibrium. Since we know that LP (4.6) has a solution > 0, and (from above) each coordinate of the solution is representable as $\frac{c}{d}$ for integers c and d, each representable using at most $|E|^2(\log n + \beta) + (|E| + 1)\beta$ bits, this gives a total utility for player i that is more than $\frac{1}{2^{|E|^2(\log n+\beta)+(|E|+1)\beta}} = \epsilon_1$ larger than the original total utility. Therefore, the original solution was not a valid ϵ_1-approximate personalized equilibrium, contradicting our assumption and completing the proof. ∎

THEOREM 4.13. *There is a polynomial-time reduction from finding an exact personalized equilibrium in a game, which has $|E|$ edges and every utility representable with at most β bits, to finding an ϵ-approximate personalized equilibrium for any $\epsilon \leq \frac{1}{2^{q(|E|,\beta)}}$, where $q(x, y)$ is a polynomial in x and y.*

Proof. The reduction consists of two steps. In the first step, we find an ϵ-approximate personalized equilibrium for the given game in which we want to find an exact personalized equilibrium. In the second step, we consider the following LP for $w_i(e)$ and $p_i(s)$ for all $i \in 1, \dots, k,\ s \in S_i,\ e \in E_i$, where $E_i' \subset E_i$ is equal to the set of all edges e such that player i assigned $\leq 1/(|E| \cdot 2^{6|E|^4\beta})$ weight to edge e in the approximate equilibrium:

$$\sum_{e \in E_i: s \in e} w_i(e) \leq \sum_{e \in E_j: s \in e} w_j(e), \quad 1 \leq i, j \leq k,\ s \in S_j, \tag{4.7}$$

$$\sum_{e \in E_i} w_i(e) = 1, \quad 1 \le i \le k,$$
$$w_i(e) = 0 \quad \forall e \in E_i',\ 1 \le i \le k,$$
$$w_i(e) \ge 0, \quad 1 \le i \le k,\ e \in E_i \setminus E_i'.$$

We let $q(|E|, \beta)$ equal $\epsilon_1(|E|, \log|E| + 6|E|^4\beta)$, where ϵ_1 is as defined in the statement of Lemma 4.12. We first observe that by Lemma 4.12 the approximate equilibrium found in the first step will satisfy all the constraints of program (4.4) to within $1/(|E| \cdot 2^{6|E|^4\beta})$. Second, we claim that any solution for LP (4.7) (if it exists) also exactly satisfies program (4.4). This is because the only difference between the two programs is that the "min" constraint of program (4.4) has been replaced by the constraint in LP (4.7) that $w_i(e)$ equals 0 for all $e \in E_i'$. Since the approximate equilibrium satisfied each min constraint to within $\epsilon_1(|E|, 6|E|^4\beta)$, if we ensure that all of the values that are at most $\epsilon_1(|E|, 6|E|^4\beta)$ are, in fact, equal to 0, then the min constraints will be exactly satisfied.

We next observe that the approximate equilibrium found in the first step is also an approximate solution to LP (4.7) to within $\epsilon = 1/(2^{6|E|^4\beta})$. By Lemma 4.11, LP (4.7) is feasible and has an exact solution, which by our claim above is also an exact solution to program (4.4). Furthermore, this solution can be found in polynomial time. By Theorem 4.4, we can thus obtain a polynomial-time method to find an exact personalized equilibrium. ∎

We next show in Theorem 4.16 that ϵ-Approximate Personalized Equilibrium is in **PPAD**, which together with Theorem 4.13 places Personalized Equilibrium in **PPAD**. We first present two technical claims used in the proof of Theorem 4.16.

Lemma 4.14. *Given a $q \times r$ matrix A, a q-vector b such that $Ax \ge b$ is feasible, and $p \in \mathbb{R}^r$ and $e \in \mathbb{R}^q$ such that $Ap \ge b - e$, there exists p' satisfying $Ap' \ge b$ such that $\|p - p'\|_\infty$ is at most $e_{\max}(A_{\max}D_{\max})^q$, where $e_{\max} = \max_j |e_j|$, $A_{\max} = \max\{1, \max_{ij}|A_{ij}|\}$, and $D_{\max}$ is the maximum of 1 and the largest absolute value of the determinant of any submatrix of the matrix consisting of the columns of A and b.*

Proof. We find a point y such that $Ay \ge b - Ap$ and $\|y\|_\infty \le e_{\max}(A_{\max}D_{\max})^q$. Setting $p' = p + y$ gives us the desired lemma. We first note that $Ay \ge b - Ap$ is feasible since it is satisfied by the point $x - p$, where x satisfies $Ax \ge b$. Let $d = b - Ap$. So our goal is to find a y satisfying $Ay \ge d$. By our assumption on p, we have $d_i \le e_{\max}$ for all i.

Consider the following algorithm for constructing y. Set $y = 0$ and L to be the empty LP. At the end of i iterations, we will maintain the invariant that $\|y\|_\infty \le (A_{\max}D_{\max})^i e_{\max}$. Find any constraint $A_k y \ge d_k$ not in L that is not satisfied. (If no such constraint exists, then we are done.) Add this constraint to L. By the invariant on $|y|$, it follows that $|A_k y|$ is at most $A_{\max}^{i+1}D_{\max}^i e_{\max}$. Since $d_k \le e_{\max}$, $d_k > A_k y > -A_{\max}^{i+1}D_{\max}^i e_{\max}$, and $A_{\max}$ and $D_{\max}$ are at least 1, it follows that $|d_k|$ is at most $\max\{e_{\max}, A_{\max}^{i+1}D_{\max}^i e_{\max}\}$. Since the right-hand side of every inequality of L is at most this number, and the left-hand side is a submatrix of A, by Cramer's rule there exists a vertex of L, every coordinate of which has magnitude at most $A_{\max}^{i+1}D_{\max}^i e_{\max}$ times the largest entry in the determinant of any submatrix of A, which is at most $D_{\max}$. (Note that L is feasible since $Ay \ge d$ is feasible.) This yields the desired invariant $\|y\|_\infty \le (A_{\max}D_{\max})^{(i+1)}e_{\max}$.

The above procedure stops in at most q iterations and yields a point y such that

$\|y\|_\infty \le e_{\max}(A_{\max}D_{\max})^q$, thus completing the proof of the lemma. ∎

COROLLARY 4.15. *Let A be a $q \times r$ matrix, b be a q-vector, and $\widehat{p}$ be the lexicographically smallest vector in $Ax \ge b$. Let $b' \in \mathbb{R}^q$ be such that $Ax \ge b'$ is feasible. If $\widehat{p'}$ is the lexicographically smallest vector in $Ax \ge b'$, then $\|\widehat{p} - \widehat{p'}\|_\infty$ is at most $e_{\max}(A_{\max}D_{\max})^{qr+r(r+1)}$, where $e_{\max} = \max_j |b_j - b'_j|$, $A_{\max} = \max_{i,j}|A_{ij}|$, and $D_{\max}$ is the largest determinant, in absolute value, of any submatrix of the matrix consisting of columns from A and b.*

Proof. Let L and L' denote the LPs $Ax \ge b$ and $Ax \ge b'$. Let $\widehat{p}$ and $\widehat{p'}$ be as defined in the statement of the corollary. We apply Lemma 4.14 with (A, p, b, e) replaced by $(A, \widehat{p}, b', b - b')$ to obtain a point p' satisfying L' such that $|\widehat{p}_1 - p'_1|$ is at most $e_{\max}(A_{\max}D_{\max})^q$. Similarly, we apply Lemma 4.14 with (A, p, b, e) replaced by $(A, \widehat{p'}, b, b' - b)$ to obtain a point p satisfying L such that $|p_1 - \widehat{p'}_1|$ is at most $e_{\max}(A_{\max}D_{\max})^q$. Since $\widehat{p}$ and $\widehat{p'}$ are lexicographically smallest for L and L', it follows that $\widehat{p}_1 \le p_1 \le \widehat{p'}_1 + e_{\max}(A_{\max}D_{\max})^q$. Similarly, $\widehat{p'}_1 \le p'_1 \le \widehat{p}_1 + e_{\max}(A_{\max}D_{\max})^q$. Thus, $|\widehat{p}_1 - \widehat{p'}_1|$ is at most $e_{\max}(A_{\max}D_{\max})^q$.

We now add constraints $x_1 = \widehat{p'}_1$ and $x_2 = \widehat{p'}$ to the LPs L and L', respectively. We next apply Lemma 4.14 with (A, p, b, e) replaced by $(\tilde{A}, p, \tilde{b}, \tilde{e})$, where $\tilde{A}$ is the constraint matrix of revised program L, $\tilde{b}$ is the right-hand side of the revised L, and $\tilde{e}$ is the vector obtained by adding two additional coordinates to e, each with magnitude at most $|\widehat{p}_1 - \widehat{p'}_1|$ (for the two new inequality constraints resulting from the addition of $x_1 = \widehat{p'}_1$). Following an argument similar to that used in the previous paragraph, we obtain that $|\widehat{p}_1 - \widehat{p'}_1|$ and $|\widehat{p}_2 - \widehat{p'}'_2|$ are both at most $e_{\max}(A_{\max}D_{\max})^{2q+2}$. Repeating this for all the coordinates yields the desired inequality that $\|\widehat{p} - \widehat{p'}\|_\infty$ is at most $e_{\max}(A_{\max}D_{\max})^{qr+r(r+1)}$. ∎

THEOREM 4.16. ϵ-APPROXIMATE PERSONALIZED EQUILIBRIUM $\le_P$ END OF THE LINE.

Proof. We used fixed point theorems to prove the existence of a personalized equilibrium, and relaxing the problem to finding ϵ-approximate equilibria automatically moves us from a continuous to a discrete world. Here, we show that finding an ϵ-approximate equilibrium is in **PPAD**. This is not surprising given that several discrete fixed point problems have been shown to be in the class **PPAD**. Our proof uses the machinery already established for proving that finding an approximate Nash equilibrium in r-player games is in **PPAD** [13].

Let k denote the number of players, and let the players be numbered 0 through $k-1$. Let m equal the total number of edges ($|E|$) in the given personalized equilibrium game instance. We map any collection of weight assignments $\{w_p\}$ satisfying $\sum_{e\in E_p} w_p(e) = 1$ for all p to a unique point $\overline{w} \in \mathbb{R}^m$ as follows: for the ith edge e of p, $\overline{w}_t = w_p(e)$, where $t = i + \sum_{j<p}|E_j|$. For player p, let D_p denote the set $\{i + \sum_{j<p}|E_i| : 1 \le i \le |E_i|\}$. Let W denote the set $\{x \in \mathbb{R}^m : \sum_{t\in D_p} x_t = 1$ for all $p\}$. Then, there is a one-to-one correspondence between weight assignments $\{w_p\}$ satisfying $\sum_{e\in E_p} w_p(e) = 1$ for all p and the set W. For notational convenience, given a point $\overline{w}$ in W, we equate $w_p(e)$ to $\overline{w}_t$, where $t = i + \sum_{j<p}|E_j|$ and e is the ith edge of E_p; we also use w_p to denote the projection of $\overline{w}$ to D_p. To carry out the proof of [13, section 3.2] to personalized equilibria, we need to define a polynomial-time computable function $f : W \to W$ satisfying the following conditions:

1. For $\overline{w} \in W$, for every player p, $\sum_{t\in D_p} f(\overline{w})_t = 1$.
2. If $\|\overline{w} - \overline{w'}\|_\infty < \delta$, then $\|f(\overline{w}) - f(\overline{w'})\|_\infty < U_{\max}2^{\mathrm{poly}(n,m,k)}\delta$, where m is the number of edges and $U_{\max}$ is the maximum payoff entry in the given instance.

3. If $\|f(\overline{w}) - \overline{w}\|_\infty < \epsilon_1$, then $\overline{w}$ represents an ϵ-approximate personalized equilibrium. Here, we will use any $\epsilon_1 \leq \frac{\epsilon}{nU_{\max}}$. In the proof from [13], ϵ_1 affects only the number of nodes in the END OF THE LINE graph.

Essentially, we are reducing the problem of finding an ϵ-approximate personalized equilibrium to one of finding a suitably approximate fixed point to a Lipschitz map f.

We define f as follows: $f(\overline{w}) = \overline{x}$, where x_p is the lexicographically least best response to w_{-p}. We next show that f satisfies the three conditions listed above.

The first condition is immediate from the fact that any best response is required to assign a total weight of 1 to its edges.

For the second condition, fix $\overline{w}$, $\overline{w'}$, and a player p. Then x_p is obtained by solving a best response LP for player p given the weights w_{-p} assigned by the other players. The LP, which we denote by $\mathcal{O}$ for this proof, is over the $|E_p|$ coordinates of x_p and maximizes a linear utility $u_p(x_p)$ subject to linear constraints $B \cdot x_p^T \leq c$. We note that every element of B is either 0 or 1, and every element of c is either 0, 1, or a coordinate of w_{-p}. Similarly, if $f(\overline{w'}) = \overline{x'}$, then x'_p is an optimal solution to an LP $\mathcal{O}'$, which maximizes $u_p(x'_p)$ subject to $B \cdot (x'_p)^T \leq c'$, where c' is derived from w'_{-p} in the same way as c is derived from w_{-p}.

Let U and U' denote the optimal values of $\mathcal{O}$ and $\mathcal{O}'$. We first argue that if $\|\overline{w} - \overline{w'}\|_\infty \leq \delta$, then $|U - U'| \leq mU_{\max}(nk)!\delta$, where m is the number of edges and $U_{\max}$ is the maximum payoff entry in the given game. We note that x_p satisfies the constraints of $\mathcal{O}'$ to within δ since the c and c' vectors differ in each entry by at most δ. We also know that $\mathcal{O}'$ is feasible. The number of variables and constraints in both $\mathcal{O}$ and $\mathcal{O}'$ are $n + m$ and $nk + m + 1$. Therefore, by Lemma 4.14, there exists a point y_p that satisfies the constraints of $\mathcal{O}'$ such that $\|y_p - x_p\|_\infty \leq (nk)!\delta$; here we use the fact that every entry in the constraint matrix and vector of $\mathcal{O}$ and $\mathcal{O}'$ is at most 1. Thus, the utility achieved by y is at least $u(x_p) - mU_{\max}(nk)!\delta$, yielding $U' \geq U - mU_{\max}(nk)!\delta$. Similarly, we have $U \geq U' - mU_{\max}(nk)!\delta$. This gives the desired bound $|U - U'| \leq mU_{\max}(nk)!\delta$.

By definition, we have that x_p is the lexicographically least element of the feasibility LP consisting of the constraints of $\mathcal{O}$ together with the constraint $u(x_p) \geq U$. Let us call this LP $\mathcal{P}$. Similarly, x'_p is the lexicographically least element of the feasibility LP $\mathcal{P}'$ consisting of the constraints of $\mathcal{O}'$ together with the constraint $u(x'_p) \geq U'$. We note that P and P' have the same set of variables and the same constraint matrix; that is, P and P' can be written down as $Ax \geq b$ and $Ax \geq b'$, respectively. Since $\|\overline{w} - \overline{w'}\|_\infty \leq \delta$ and $|U - U'| \leq mU_{\max}(nk)!\delta$, we have $\|b - b'\|_\infty \leq mU_{\max}(nk)!\delta$. We now apply Corollary 4.15 to obtain that $\|x'_p - x_p\|_\infty$ is at most $U_{\max}2^{\mathrm{poly}(n,m,k)}\delta$.

For the third condition, recall our definition of an ϵ-approximate personalized equilibrium. We require that (3a) for every player p, $\sum_{e \in E_p} w_p(e) = 1$; (3b) for each player pair p and q, and for each strategy s, $\left|\sum_{e:s\in e} w_p(e) - \sum_{e:s\in e} w_q(e)\right| \leq \epsilon$; and (3c) for any best response weight assignment w_p^* for any player p, $\sum_e w_p^*(e)u_p(e) - \sum_e w_p(e)u_p(e) \leq \epsilon$. (3a) is immediate, and we have $\sum_e w_p(e) = 1$. For (3b), recall that x_p is the exact best response to w_{-p}. So we have $\sum_{e:s\in e} x_p(e) \leq \sum_{e:s\in e} w_q(e)$ for all players p and q, and strategy $s \in S_q$. Since $|\overline{x} - \overline{w}|_\infty$ is at most ϵ_1, we have $\sum_{e:s\in e} x_p(e) - \sum_{e:s\in e} w_q(e) \leq m\epsilon_1$ for all players p and q, and strategy $s \in S_q$.

Finally, we consider condition (3c). By the definition of f, we have that x_p is a best response to w_{-p}. For any edge e of E_p, we are given that $w_p(e) - x_p(e) < \epsilon_1$. Thus, $\sum_e x_p(e)u_p(e) - \sum_e w_p(e)u_p(e) \leq \epsilon_1 \cdot |E_p|U_{\max} \leq \epsilon$. This completes the proof of the theorem. $\square$

5. Scarf's lemma and fractional stability problems. This section discusses the complexity of a number of well-known combinatorial problems that can be categorized as *fractional stability problems*. We begin with Scarf's lemma, a fundamental result in combinatorics, originally introduced to prove that every balanced cooperative game with nontransferable utilities has a nonempty core (see section 5.3) [40]. The core (no pun intended) of Scarf's proof is an elegant and constructive combinatorial argument, which has been applied to diverse combinatorial problems, including fractional stable matchings in hypergraphic preference systems, strong kernels in digraphs, and the fractional stable paths problem (FSPP) [2, 1, 27, 20]. We first show that the computational version of Scarf's lemma is **PPAD**-complete. We then establish the **PPAD**-completeness of stable matchings in hypergraphic preference systems, strong kernels in digraphs, the core of balanced games with nontransferable utility, the FSPP, and a fractional version of the bounded budget connection game [29, 30].

5.1. Scarf's lemma. In the computational version of Scarf's lemma (Scarf) we are given matrices B, C and a vector b satisfying the conditions in Theorem 5.1, and the goal is to find $\alpha \in \mathbb{R}^n_+$ satisfying the desired properties.

Theorem 5.1 (Scarf's lemma [40]). *Let $I = [\delta_{ij}]$ be an $m \times m$ identity matrix. Let $[n] = \{1, 2, \ldots, n\}$. Let $m < n$, and let B be an $m \times n$ real matrix such that $b_{ij} = \delta_{ij}$ for $1 \leqslant i, j \leqslant m$. Let b be a nonnegative vector in $\mathbb{R}^m$ such that the set $\{\alpha \in \mathbb{R}^n_+ : B\alpha = b\}$ is bounded. Let C be an $m \times n$ matrix such that $c_{ii} \leqslant c_{ik} \leqslant c_{ij}$ whenever $i, j \leqslant m$, $i \neq j$, and $k > m$. Then there exists a subset $J \subset [n]$ of size m such that*

(P1) *$B\alpha = b$ for some $\alpha \in \mathbb{R}^n_+$ such that $\alpha_j = 0$ whenever $j \notin J$, and*

(P2) *for every $k \in [n]$, there exists $i \in [m]$ such that $c_{ik} \leqslant c_{ij}$ for all $j \in J$.*

A subset $J \subset [n]$ of size m is called a *feasible basis* of (B, b) if it satisfies (P1), and *subordinating* if it satisfies (P2). To compute α of Scarf, it suffices to have a $J \subseteq [n]$ that is simultaneously subordinating and a feasible basis. Once such J is computed, α can be computed by solving a system of linear equations. Also, given a solution α, J is easy to compute, since J is α's support. Hence finding α and J are computationally equivalent, to within polynomial time. In Theorem 5.2, we argue that Scarf's original proof [40] (we follow the presentation in [1]), together with Todd's orientation technique [45], gives an end-of-the-line argument for the existence of a subordinating and feasible basis, thus showing that Scarf is in **PPAD**.

Scarf: Given matrices B and C and vector $b \in \mathbb{R}^m$, all obeying the requirements from Theorem 5.1, find a subset of m column indices that is a feasible basis for (B, b) and is subordinating for C.

Theorem 5.2. Scarf $\leq_P$ End of the Line.

Proof. We first apply standard perturbation techniques to remove "degeneracies" in the input. We say that the pair (B, b) is *degenerate* if b is in the cone spanned by fewer than m columns of B, and *nondegenerate* otherwise. We first apply a small perturbation b' of b such that the pair (B, b) is nondegenerate and every feasible basis for (B, b') is also a feasible basis for (B, b). Such a perturbation can be found in polynomial time using standard techniques in linear programming (e.g., see Chapter 10 of [12]).

Similarly, by slightly perturbing C, we obtain an ordinal-generic matrix C' (i.e., all the elements in each row of C' are distinct) satisfying the assumptions of Theorem 5.1, and by making perturbation small enough, we ensure that any subordinating set for C' is also subordinating for C. For the sake of completeness, we present a simple polynomial-time computable perturbation. Let δ equal the minimum, over all i, j, i', j' such that $c_{ij} \neq c_{i'j'}$, $|c_{ij} - c_{i'j'}|$. That is, δ is the minimum positive difference between any two terms in C. We then set the elements of C' as follows: c'_{ii} is set to c_{ii} for all i; c'_{ik} is set to $c_{ik} + \delta(k-m)/(n+1)$ for all i and $k > m$; c'_{ij} is set to $c_{ij} + \delta(n-m+j)/(n+1)$ for all i and $1 \leq j \leq m$, $j \neq i$. We first argue that C' is ordinal-generic; that is, all the terms in any row of C' are distinct. If $c'_{ij} = c'_{ik}$ for any $j \neq k$, then $c_{ij} + \delta_j = c_{ik} + \delta_k$ for δ_j, δ_k satisfying $0 < |\delta_j - \delta_k| < \delta$. But this is impossible by the definition of δ.

We next show that C' satisfies the conditions of Scarf's lemma. That is, whenever $i, j \leq m$, $i \neq j$, and $m < k \leq n$, we have $c'_{ii} \leq c'_{ik} \leq c'_{ij}$. Since $c_{ii} \leq c_{ik} \leq c_{ij}$, $c'_{ii} = c_{ii}$, $c'_{ik} = c_{ik} + \delta(k-m)/(n+1)$, and $c'_{ij} = c_{ij} + \delta(n-m+j)/(n+1)$, the desired claim follows.

Finally, we show that any subordinating set for C' is also subordinating for C. If $c'_{ik} \leq c'_{ij}$ for any given i, j, k, then $c_{ik} \leq c_{ij} + \delta'$ for some δ' satisfying $|\delta'| < \delta$. By the definition of δ, we then have $\delta' = 0$, implying that $c_{ik} \leq c_{ij}$. This completes the desired claim.

Hence, we may assume for the remainder of the proof that (B, b) is nondegenerate and that C is ordinal-generic. Lemma 5.3 is well known. Its proof requires that $\{\alpha \in \mathbb{R}^n_+ : B\alpha = b\}$ is bounded and (B, b) is nondegenerate. For the proof of Lemma 5.4, we refer the reader to [40] or [2] or page 1127 of Schrijver's book *Combinatorial Optimization* [41]. The proof of Lemma 5.4 uses the assumption that C is ordinal-generic.

LEMMA 5.3. *Let J be a feasible basis for (B, b), and let $k \in [n] \setminus J$. Then there exists a unique $j \in J$ such that $J + k - j$ (i.e., $J \cup \{k\} \setminus \{j\}$) is a feasible basis. Also, given J and k, we can find j in polynomial time.*

LEMMA 5.4 (Scarf [40]). *Let K be a subordinating set for C of size $m-1$. Then there are precisely two elements $j \in [n] \setminus K$ such that $K + j$ is subordinating for C, unless $K \subseteq [m]$, in which case there exists precisely one such j. Given K, we can find values of j in polynomial time.*

The natural pivot rules arising from Lemmas 5.3 and 5.4 are called the *feasible pivot rule* and the *ordinal pivot rule*, respectively.

The original proof of Scarf's lemma [40, 2] uses an "*undirected* end of the line argument," thus showing its **PPA**-membership. It is easy to see that **PPAD** $\subseteq$ **PPA**; however it is unknown whether **PPAD** $=$ **PPA**. To prove **PPAD**-membership of SCARF, we need a "*directed* end of the line argument." Shapley [43] presented a geometric orientation rule for the equilibrium points of (nondegenerate) bimatrix games based on the Lemke–Howson algorithm [31]. Extending Shapley's rule, Todd [45] developed a similar orientation theory for generalized complementary pivot algorithms. We now apply Todd's orientation technique to prove **PPAD**-membership of SCARF.

Let $X = \{1, 2, \ldots, n\}$. A subset of X of cardinality m is called an m-subset. Let X_m denote the collection of ordered (with the natural ordering defined by X) m-tuples of distinct elements of X. Two m-tuples in X_m are *equivalent* if and only if one is an even permutation of the other. Let P be any element of an equivalent set. We denote the corresponding equivalent set by $\overline{P}$. If $P' \in X_m$ is an odd permutation of $P \in X_m$, then we call $\overline{P'}$ the *negative* of P and write $\overline{P'} = -\overline{P}$. Let $P = (e_1, \ldots, e_n) \in X_n$. For $\mu = \pm 1$, we say $\overline{P}$ contains $\mu(\overline{P \setminus e_i})$ positively (negatively) if $\mu(-1)^i$ is positive (negative).

Let $e \in X$ be a specific element. Let $\mathcal{F}$ be the set of all feasible bases containing e, and let $\mathcal{S}$ be the set of all subordinating sets of size m *not* containing e. Note that both $\mathcal{F}$ and $\mathcal{S}$ are m-subsets of $[n]$. Let $V(\mathcal{F}, \mathcal{S}, e)$ be the set of pairs $(\overline{F}, \overline{S}) \in \mathcal{F} \times \mathcal{S}$

satisfying either (i) $\overline{F} = \pm\overline{S}$ (called a matched pair) or (ii) $e \in F$, $e \notin S$, and $F \backslash S = \{e\}$ (called an unmatched pair). A matched pair $(\overline{T}, \overline{T})$ is positive, while $(\overline{T}, -\overline{T})$ is negative. An unmatched pair $(\overline{F}, \overline{S})$ is positive (negative) if $\overline{F}$ is contained in $(\overline{S \cup e})$ positively (negatively).

LEMMA 5.5 (Todd [45]). (a) *Every matched pair is adjacent to one unmatched pair by a feasible pivot, or one unmatched pair by an ordinal pivot, but not both.* (b) *Every unmatched pair is adjacent to one pair by a feasible pivot and one pair by an ordinal pivot.*

LEMMA 5.6 (Todd [45]). (a) *If two unmatched pairs are adjacent by a feasible pivot, they have opposite signs.* (b) *If a matched pair and an unmatched pair are adjacent by a feasible pivot, they have the same sign.* (c) *If two pairs are adjacent by an ordinal pivot, they have opposite signs.*

Similarly to [45], we construct a directed graph with vertices representing the pairs in $V(\mathcal{F}, \mathcal{S}, e)$. If two unmatched pairs are adjacent by a feasible pivot, we add a directed edge from the negative pair to the positive pair. If a matched pair is adjacent by a feasible pivot to an unmatched pair, we add a directed edge from the matched pair to the unmatched pair if both are positive and in the reverse direction if both are negative. If two pairs are adjacent by an ordinal pivot, we add a directed edge from the positive pair to the negative pair. From Lemmas 5.5 and 5.6, it follows that each unmatched pair has in-degree 1 and out-degree 1. Each matched pair has in-degree 0 and out-degree 1 if positive, and in-degree 1 and out-degree 0 if negative. It is easy to see that $[m]$ is in $\mathcal{F}$ and is not subordinating. By Lemma 5.4 there exists $f \neq e$ such that $[m] - e + f$ is in $\mathcal{S}$. We shall use the pair $([m], [m] - e + f)$ as the initial source of END OF THE LINE. This gives the required **PPAD** property. ∎

In section 5.2, we establish the **PPAD**-hardness of FRACTIONAL HYPERGRAPH MATCHING, which reduces to SCARF in polynomial time [1], thus completing the proof that SCARF is **PPAD**-complete.

5.2. Hypergraphic preference systems. A hypergraphic preference system is a pair $(H, \mathcal{O})$, where $H = (V, E)$ is a hypergraph and $\mathcal{O} = \{\preceq_v :\ v \in V\}$ is a family of linear orders, $\preceq_v$ being an order on the set of edges containing the vertex v. A set M of edges is called a *stable matching* with respect to the preference system if (a) it is a matching, and (b) for every edge e there exists a vertex $v \in e$ and an edge $m \in M$ containing v such that $e \preceq_v m$. A nonnegative function w on the edges in H is called a *fractional matching* if $\sum_{v \in h} w(h) \leq 1$ for every vertex v. A fractional matching w is called *stable* if every edge e contains a vertex v such that $\sum_{v \in h, e \preceq_v h} w(h) = 1$.

Aharoni and Fleiner [1] used Scarf's lemma to prove that every hypergraphic preference system has a fractional stable matching. This naturally leads to the following computational problem: given a hypergraphic preference system $(H, \mathcal{O})$, find a fractional stable matching.

FRACTIONAL HYPERGRAPH MATCHING: Given a hypergraph H and a preference ordering $\mathcal{O}$ for each node in H, find a stable fractional matching.

We first observe that the proof of [1] is a polynomial-time reduction from FRACTIONAL HYPERGRAPH MATCHING to SCARF, thus placing it in **PPAD**. We now show that FRACTIONAL HYPERGRAPH MATCHING is **PPAD**-hard via a reduction from preference games.

THEOREM 5.7. DEGREE d PREFERENCE GAME $\leq_P$ FRACTIONAL HYPERGRAPH MATCHING.

Proof. We are given a preference game over players $[n] = \{1, \ldots, n\}$. We construct the hypergraph matching instance $(H, \mathcal{O})$, $H = (V, E)$. The set V of vertices is $[n] \cup \{i^* : i \in [n]\}$; that is, we have two vertices i and i^* for each player i. The set of edges is given by

$$\{\{i^*\} : i \in [n]\} \bigcup \{\{i, i^*\} \cup J_i : i \in [n], J_i \subseteq \text{in}(i)\}\}.$$

(Note that J_i is a subset of players that prefer i over themselves.)

We next describe the linear order for a given vertex i. Let e_1 and e_2 be two edges containing i. By our construction of E, there exists a unique i_1 such that $\{i_1, i_1^*\}$ is a subset of e_1. Similarly, there is a unique i_2 such that $\{i_2, i_2^*\}$ is a subset of e_2. If $i_1 \neq i_2$, then we require that $e_1 \succeq_i e_2$ if and only if $i_1 \succeq_i i_2$; if $i_1 = i_2$, then we require that $e_1 \succeq_i e_2$ whenever $e_1 \supseteq e_2$. We set $\succeq_i$ to be any linear extension of the partial order satisfying the preceding condition. Finally, for any vertex i^*, we select any linear order in which $e_1 \succeq_{i^*} e_2$ whenever $\{i, i^*\}$ is a subset of e_1 and $e \succeq_{i^*} \{i^*\}$ for all e.

The number of vertices in the above hypergraph is $2n$, and the number of edges is at most $n(2^d + 1)$, where d is the maximum in-degree of the preference game. Since we are given a preference game of constant degree, the above is a polynomial-time construction.

We show that there is a stable solution for the preference game if and only if there is a stable fractional matching for the hypergraph preference system. Suppose w is a stable solution for the preference game: w_{ij} represents the weight assigned by player i to player j. For a given player j, we sort all the players i in $\text{in}(j)$ in nonincreasing order of the w_{ij} values; let the order be $j_1, j_2, \ldots, j_{d_j}$, where d_j is the in-degree of j. To every edge of the form $\{j, j^*\} \cup \{j_1, \ldots, j_k\}$, $1 \leq k < d_j$, we assign the weight $w_{j_k j} - w_{j_{k+1} j}$. We assign weight $w_{j_{d_j}}$ to the edge $\{j, j^*\} \cup \text{in}(j)$ and weight $w_{jj} - w_{j_1 j}$ to the edge $\{j, j^*\}$. Finally, we assign weight $1 - w_{jj}$ to the edge $\{j^*\}$. This ensures the following:

$$\begin{aligned} \textstyle\sum_{e:\{j,j^*\} \subseteq e} f(e) &= w_{jj} \quad \forall\, j, \\ \textstyle\sum_{e:\{j,j^*,i\} \subseteq e} f(e) &= w_{ij} \quad \forall\, j,\, i \in \text{in}(j). \end{aligned}$$

We next argue that the fractional matching f thus defined is stable.

There are three types of edges for us to consider: (1) $e = \{j, j^*, j_1, j_2, \ldots, j_k\}$ for some j, k; (2) $e = \{j, j^*\} \cup S$ for some $j, S \neq \{j_1, j_2, \ldots, j_k\}$ for any k; and (3) $e = \{j^*\}$ for some j.

First consider $e = \{j, j^*, j_1, j_2, \ldots, j_k\}$ for some j, k. We separate this into two cases. The first case is when there is no proper subset of e that has positive weight. In this case, we argue that j is a vertex in e such that $\sum_{h \succeq_j e} f(h) = 1$. $\sum_{h \succeq_j e} f(h) = \sum_{i \succeq_j j} \sum_{h:\{i,i^*,j\} \subseteq h} f(h) + \sum_{e \subseteq h} f(h) = \sum_{i \succeq_j j} w_{ji} + \sum_{h:e \subseteq h} f(h) + \sum_{h:h \subset e} f(h) = \sum_{i \succeq_j j} w_{ji} + \sum_{h:\{j,j^*\} h} f(h) = \sum_{i \succeq_j j} w_{ji} + w_{jj} = 1$. The second case is when there is some proper subset e' of e with positive weight. Say $e' = \{j, j^*, j_1, j_2, \ldots, j_{s-1}\}$ for $s \leq k$. Because $s \leq k$, $j_s \in e$. We will show that j_s is a vertex in e such that $\sum_{h \succeq_{j_s} e} f(h) = 1$. Since e' has positive weight, $w_{j_{s-1} j} - w_{j_s j} > 0 \Rightarrow w_{j_s j} < w_{jj}$. Therefore, since w is a preference game equilibrium, $\sum_{i \succeq_{j_s} j} w_{j_s i} = 1$. So, $\sum_{i \succeq_{j_s} j} \sum_{h:\{i,i^*,j_s\} \subseteq h} f(h) = 1 \Rightarrow \sum_{h:\{j,j^*,j_s\} \subseteq h} f(h) + \sum_{i \succeq_{j_s} j, i \neq j} \sum_{h:\{i,i^*,j_s\} \subseteq h} f(h) = 1 \Rightarrow \sum_{h \succeq_{j_s} e} f(h) = 1$.

Next consider $e = \{j, j^*\} \cup S$ for some $j, S \neq \{j_1, j_2, \ldots, j_k\}$ for any k. Now, pick edge $e' \supset e$, $e' = \{j_1, j_2, \ldots, j_k\}$ for $j_k \in e$. Again, we can separate this into two cases based on whether or not there is a proper subset of e' with positive weight. If there is no such proper subset, then j will have $\sum_{h \succeq_j e} f(h) = 1$, by the same argument as above. If there is a proper subset $e'' \subset e'$ with positive weight, we will argue that j_k satisfies $\sum_{h \succeq_{j_k} e} f(h) = 1$. Since $\{j, j^*\} \subseteq e''$, $j_k \notin e''$, $\sum_{h=\{j,j^*\} \cup S} f(h) \geq f(e'') + \sum_{h:\{j,j^*,j_k\} \subseteq h} f(h) \Rightarrow w_{jj} \geq f(e'') + w_{j_k j}$. We picked e'' such that $f(e'') > 0$, so $w_{jj} > w_{j_k j}$. As in the previous paragraph, this implies that $\sum_{h \succeq_{j_k} e} f(h) = 1$.

To complete this direction of the lemma, consider an edge $\{j^*\}$ for some j. By our construction, this is the least preferred edge for j^*, and the assignment of weight $1 - w_{jj}$ guarantees that the sum of the weights of all edges containing j^* equals 1.

We next prove the other direction of the lemma. Suppose f is a stable fractional matching for the hypergraph preference system. We construct the following assignment for the preference game. We set w_{ij} to be the sum of the weights of edges containing the subset $\{j, j^*, i\}$. It is easy to see that $w_{ij} \leq w_{jj}$ for all i and j. It remains to argue the stability of w.

First we claim that if f is stable, then for any S_1 and S_2 such that $S_1 - S_2$ and $S_2 - S_1$ are both nonempty, at most one of $f(\{j, j^*\} \cup S_1)$ and $f(\{j, j^*\} \cup S_2)$ is positive. To see this, observe that if both are positive, then for every vertex v in the edge $e = \{j, j^*\} \cup S_1 \cup S_2$, the sum of weights assigned to edges that are at least as much preferred by v as e is less than 1 since v is in either $\{j, j^*\} \cup S_1$ or $\{j, j^*\} \cup S_2$, both of which have positive weight and are less preferred than e by v. This implies that such a matching could not be stable for edge e. Thus, in f, the positive weights to edges that are a superset of $\{j, j^*\}$ are all assigned to a chain of edges $e_1 \subset e_2 \subset \cdots \subset e_k$ for some k. Define e_{k+1} to be $\{j, j^*\} \cup \mathrm{in}(j)$. We next observe that for every vertex v in $e_i - e_{i-1}$, $1 < i \leq k+1$, the sum of the weights of the edges that v prefers at least as much as e_i equals 1. This is because such a vertex exists in $e_{i-1} \cup \{v\}$ (by the definition of stable matching) and v is the only possibility.

Consider any $w_{i\ell} > 0$. To establish stability of w, we prove by a contradiction that for all j such that $j \succeq_i \ell$, $w_{ij} = w_{jj}$. Suppose not; then there exist a j such that $j \succeq_i \ell$, and two edges $e, e' \supseteq \{j, j^*\}$ with $i \in e$, $i \notin e'$, and $f(e') > 0$. Let e denote the smallest edge containing i in the chain $e_1 \subset e_2 \subset \cdots \subset e_{k+1}$ mentioned in the preceding paragraph. (Since $i \in e_{k+1}$, e exists.) By the argument above, the sum of the weights of the edges that i prefers at least as much as e equals 1. However, $w_{i\ell} > 0$ implies that there exists an edge $e'' \supseteq \{\ell, \ell^*, i\}$ with $f(e'') > 0$, yielding a contradiction since i prefers e over e''. $\square$

5.3. Cooperative games with nontransferable utilities.

DEFINITION 5.8. *A game with nontransferable utilities over n players is specified by a function V that for each subset S of $N = \{1, 2, \ldots, n\}$ returns a set $V(S)$ of outcomes—each outcome being a vector of utility values, one component for each player in S. A collection T of coalitions is* balanced *if there exists an assignment of nonnegative reals δ_S for each coalition S in T such that for all v, $\sum_{S: v \in S} \delta_S = 1$. We say that u is* attainable *by S if $u \in V(S)$. A game is* balanced *if and only if for any balanced collection T and any u, if u_S is attainable by all S in T, then u is attainable by N.*

The core of a cooperative game is a solution concept which requires that no set of players have an incentive to secede. In games with nontransferable utilities, the core is defined as the set

$$\{x \in V(N) : \forall S \text{ there is no } y \in S \text{ such that } x_i > y_i \; \forall i \in S\}.$$

As mentioned earlier, Scarf [40] proved that every balanced game has a nonempty core. We define Core-Balanced-NTU below, a natural computational version of this claim. Scarf's proof [40], which is a reduction to Scarf, together with Theorem 5.9, establishes its **PPAD**-completeness.

Core-Balanced-NTU: The game is specified by a set N of players, a collection $\mathcal{S}$ of proper subsets of N (the coalitions), and for each $S \in \mathcal{S}$, vectors $u_1, \ldots, u_{k_S}$ in $\mathbb{R}^{|S|}$ such that $V(S) = \{u \in \mathbb{R}^{|S|} : \exists j \; u \leq u_j\}$. For a coalition $S \notin \mathcal{S}$, $V(S) = \{0\}^{|S|}$ and $V(N)$ is defined as the set of all u for which there exists a balanced collection T such that u_S is attainable by all S in T. The goal is to find an element in the core.

Theorem 5.9. Fractional Hypergraph Matching $\leq_P$ Core-Balanced-NTU.

Proof. Suppose we are given a hypergraph H and, for each vertex i, a preference ranking among all edges containing i. We first add, for each vertex i in H, a new vertex i^* and edge $\{i, i^*\}$. We set the preference of i for the edge $\{i, i^*\}$ to be the least among all the edges containing i. Let N denote the new set of nodes and E the new set of edges. For $S \in E$ and $i \in N$, let $r_i(S)$ denote the rank of S in i's preference list, with 0 assigned to the least preferred edge (thus for every i, $r_i(\{i, i^*\}) = 0$). We now define a balanced cooperative game with nontransferable utilities. For each node in N, we have a player in the game. For any coalition S, we consider two cases. If $S \in E$, then we have a single vector $r_S = (r_{i_1}(S), r_{i_2}(S), \ldots, r_{i_{|S|}}(S))$, where $S = \{i_1, i_2, \ldots, i_{|S|}\}$. Note that, by definition, if $S \notin E$ and $S \neq N$, then $V(S) = 0^{|S|}$.

For N, note that $V(N)$ is precisely the set of all u such that u_S is attainable by all S in some balanced collection T. We first observe that we can determine in polynomial time whether a given u is in $V(N)$. For each S, if $u \leq r_S$, then we have a variable x_S for S. Now we simply solve the LP

$$\sum_{S: i \in S} x_S = 1 \quad \forall i,$$
$$x_S \geq 0 \quad \forall S.$$

It is easy to see that the LP is feasible if and only if u is in $V(N)$. Consider any balanced collection T; if we have a u such that u_S is attainable by all coalitions S in T, then the above LP would be feasible—the δ_S values that verify the balanced condition yield the solution for the above LP, and hence u is attainable by N. For the other direction, consider any u that is attainable by N. Then, by our construction the above LP is feasible. The x_S values we obtain precisely specify the δ_S values, meaning that u_S is attainable by every S for which $\delta_S > 0$.

It is straightforward to compute the above reduction in time polynomial in H. We finally claim that from an element of the core, a fractional stable hypergraph matching can be obtained in polynomial time. Suppose u is in the core. Since u is attainable by N, we find the x_S values that satisfy the above LP. We claim that the weights x_S yield a stable fractional hypergraph matching in H. Consider any edge S'. Since u is in the core, there exists a player i in S' such that the utility for i in u is at least as high as that for i in $V(S')$. Since u is attained by N, the utility (preference) of i in each S for which $x_S > 0$ is also at least as high as that of i in S'. Thus, x_S yields a stable matching. ∎

5.4. Fractional stable paths problem. The fractional stable paths problem, introduced in [20], is defined as follows. Let G be a graph with a distinguished destination node d. Each node $v \neq d$ has a list $\pi(v)$ of simple paths from v to d and a preference relation $\succeq_v$ among the paths in $\pi(v)$. For a path S, we also define $\pi(v, S)$ to be the set of paths in $\pi(v)$ that have S as a suffix. A *proper suffix* S of P is a suffix of P such that $S \neq P$ and $S \neq \emptyset$.

A *feasible fractional paths solution* is a set $w = \{w_v : v \neq d\}$ of assignments $w_v : \pi(v) \to [0, 1]$ satisfying the following:

1. *Unity condition*: for each node v, $\sum_{P \in \pi(v)} w_v(P) \leq 1$.
2. *Tree condition*: for each node v and each path S with start node u, $\sum_{P \in \pi(v,S)} w_v(P) \leq w_u(S)$.

In other words, a feasible solution is one in which each node chooses at most 1 unit of flow to d such that no suffix is filled by more than the amount of flow placed on that suffix by its starting node. A feasible solution w is *stable* if for any node v and path Q starting at v, one of the following holds:

(S1) $\sum_{P \in \pi(v)} w_v(P) = 1$, and for each P in $\pi(v)$ with $w_v(P) > 0$, $P \geq_v Q$; or

(S2) there exists a proper suffix S of Q such that $\sum_{P \in \pi(v,S)} w_v(P) = w_u(S)$, where u is the start node of S, and for each $P \in \pi(v, S)$ with $w_v(P) > 0$, $P \geq_v Q$.

In other words, in a stable solution, if node v has not fully chosen paths that it prefers at least as much as Q, then it has completely filled path Q by filling some suffix with paths it prefers at least as much as Q.

We define a computational version, FRACTIONAL SPP.

> FRACTIONAL SPP: Given a graph, a destination node, and a preference list for each node across paths to the destination, find a weight assignment for each node to the paths in its preference list that is both feasible (satisfies the unity and tree conditions) and stable (every path satisfies property (S1) or (S2)).

We note that Haxell and Wilfong [21] show FRACTIONAL HYPERGRAPH MATCHING $\leq_P$ FRACTIONAL SPP (see section 5.2), and the problem of finding a fractional co-acyclic kernel (related to STRONG KERNEL; see section 5.5) can also be reduced to FRACTIONAL SPP. Together with our reduction from FRACTIONAL SPP to PERSONALIZED EQUILIBRIUM (see Theorem 5.11), this gives an alternate proof of **PPAD**-membership for these two problems.

5.4.1. PPAD-completeness.

THEOREM 5.10. PREFERENCE GAME $\leq_P$ FRACTIONAL SPP.

Proof. We are given a preference game over player set $[n]$, including preference relation $\succeq_i$ for all $i \in 1, \ldots, n$. We will convert this into an FSPP. Create a node v_i for each i. Also create a universal destination node d. For all i, define P_{ii} = the path (v_i, d). For all i, j, define P_{ij} = the path (v_i, v_j, d). Let $\pi(v_i)$ (the set of preferred paths for v_i) = $\{P_{ij} : j \succeq_i i\}$. If $k \succeq_i j$, then $P_{ik} \succeq_i P_{ij}$.

Let $w_i(j)$ refer to the amount of weight placed by node v_i on path P_{ij} in an FSPP solution, and let $w_i(i)$ be the amount of weight placed by i on path P_{ii}. Now we will show that w is a fractional stable paths solution if and only if w defines an equilibrium of the preference game.

w is a fractional stable paths solution $\Rightarrow$ w is an equilibrium of the preference game. By the unity condition, for each i, $\sum_{j:P_{ij} \in \pi(v_i)} w_i(j) \leq 1 \Rightarrow \sum_j w_i(j) \leq 1$. P_{ii} starts at v_i, and there is no proper final suffix of P_{ii}, so condition (S1) must apply for P_{ii}. Therefore, $\sum_{j:P_{ij} \in \pi(v_i)} w_i(j) = \sum_j w_i(j) = 1$, as required for the preference game. By the tree condition, for any i, j, $\sum_{P \in \pi(v,P_{jj})} w_i(P)$ is less than or equal to

$w_j(j)$, which implies that $w_i(j) \leq w_j(j)$. So w is a feasible weight assignment for the preference game.

Now suppose for contradiction that w is not lexicographically maximal (with respect to w_{-i}) for player i in the preference game. Then there are some feasible weight assignment w' and some j such that $\sum_{k \succeq_i j} w_i(k) < \sum_{k \succeq_i j} w'_i(k)$. Take the lexicographically maximal such w' and the highest preference such j (from i's preference list). By this choice of w' and j, $\sum_{k \succ_i j} w_i(k) = \sum_{k \succ_i j} w'_i(k)$, so $\sum_{k =_i j} w_i(k) < \sum_{k =_i j} w'_i(k)$. There must be some j' with $j' =_i j$ such that $w_i(j') < w'_i(j')$. Consider path $P_{ij'}$. (S2) is not true by our choice of j' and the fact that w' is a feasible solution (so $w'_i(j') \leq w_{j'}(j')$). However, since $\sum_{k =_i j'} w_i(k) < \sum_{k =_i j'} w'_i(k)$, there must be some path P_{ik} such that $k \prec_i j'$ with $w_i(k) > 0$. So (S1) is also not true, and w is not a stable solution—a contradiction.

w is an equilibrium of the preference game $\Rightarrow$ w is a fractional stable paths solution. We know that $\sum_j w_i(j) = 1$ for all i. This immediately satisfies the unity condition. Since w is a feasible set of weights for the preference game, $w_i(j) \leq w_j(j)$ for all i, j. This means that the weight placed on P_{ij} is at most the weight placed on P_{jj}. Since P_{ij} is the only path from v_i that passes through node v_j, the tree condition holds. Now consider any path P_{ij} from node i. Case 1: $w_i(j) = w_j(j)$. In this case, condition (S2) is satisfied. Case 2: $w_i(j) < w_j(j)$. Because w is lexicographically maximal, any weight assignment w' with $\sum_{k \succeq_i j} w'_i(k) \geq \sum_{k \succeq_i j} w_i(k)$ must be infeasible. We said that $w_i(j) < w_j(j)$, so it is only possible for all such w' to be infeasible if $\sum_{k \succeq_i j} w_i(k) = 1$. Then $\sum_{k \prec_i j} w_i(k) = 0$, so (S1) is satisfied. ∎

THEOREM 5.11. FRACTIONAL SPP $\leq_P$ PERSONALIZED EQUILIBRIUM.

Proof. Suppose we are given an instance of FRACTIONAL SPP, consisting of a set of nodes V, a set of preferred paths $\pi(v)$ for all $v \in V$, and a preference relation $\succeq_v$ for each set $\pi(v)$. We can also find $\pi(v, S)$, the set of all $P \in \pi(v)$ such that S is a subpath of P. Let $q_v(P) =$ the number of paths Q such that $P \succeq_v Q$; note that $q_v(P) \geq 1$ for all P in $\pi(v)$.

We will create the following instance of PERSONALIZED EQUILIBRIUM. The set of players is V. The set of strategies S_v for a node v is $\pi(v) \cup \{N_v\}$ (N stands for "no path"). For node v, there is exactly one edge defined for each strategy. Edge P' for strategy $P = \{S : P \in \pi(v, S)\}$. The edge for strategy N_v given is the singleton edge $\{N_v\}$. The payoffs to player v are $u_v(P') = q_v(P) + 1$, $u_v(N) = 1$. Since every player has a singleton edge, the LP (4.1) is always feasible, so the resulting game is well behaved, and hence a personalized equilibrium exists by Theorem 4.1.

Suppose w is a set of weights in a personalized equilibrium of the game defined above. $w_v(P')$ represents the weight assigned by v to edge P'. We will show that w is a personalized equilibrium if and only if $w' : w'_v(P) = w_v(P')$ is a fractionally stable solution to the FRACTIONAL SPP instance.

First, assume w is a personalized equilibrium. Then, we know that for all v, the total weight placed by v on all edges is 1, or $w_v(N) + \sum_{P \in \pi(v)} w_v(P') = 1$. Therefore, $\sum_{P \in \pi(v)} w_v(P') \leq 1 \Rightarrow \sum_{P \in \pi(v)} w'_v(P) \leq 1$, so the unity condition holds. Also, the sum of weights placed by v on edges adjacent to a strategy S of another player v' is at most $w_{v'}(S)$. That is, for path $S \in \pi(v')$ ($v' \neq v$), $\sum_{P' : S \in P'} w_v(P') \leq w_{v'}(S') \Rightarrow \sum_{P \in \pi(v,S)} w_v(P') \leq w_{v'}(S') \Rightarrow \sum_{P \in \pi(v,S)} w'_v(P) \leq w'_{v'}(S)$, so the tree condition holds.

We next establish the stability condition. Take any path $Q \in \pi(v)$ and its corresponding edge e' in the PERSONALIZED EQUILIBRIUM instance. We consider two cases. The first is when every edge to which v assigns positive weight corresponds to paths that v prefers at least as much as Q. In this case, we know that v puts a total of weight 1 on edges with payoff at least $q_v(Q)+1$, or $\sum_{P':u_v(P')\geq q_v(Q)+1} w_v(P') = 1 \Rightarrow \sum_{P:P\succeq_v Q} w_v(P') = 1 \Rightarrow \sum_{P:P\succeq_v Q} w'_v(P) = 1$, so condition (S1) holds. The second case is when v assigns positive weight to some edge P' corresponding to a path P that v prefers less than Q. Since P' gives a smaller payoff to v than Q', it must be the case that v cannot add any more weight to the edge Q', implying that some constraint involving Q' is tight. By nature of the edges we've defined, any edge of v that intersects with Q' is either a superset of Q' or a subset of Q'. This means that there are some $S \in Q'$ and some node u with $S \in \pi(u)$ such that $\sum_{R':S\in R'} w_v(R') = w_{v'}(S')$ and, for all $R' : S \in R'$, if $w_v(R') > 0$, then $q_v(R) \geq q_v(Q)$. So, $\sum_{R\in\pi(v,S)} w_v(R') = w_u(S') \Rightarrow \sum_{R\in\pi(v,S)} w'_v(R) = w'_u(S)$, and, for all $R \in \pi(v,S)$ with $w'_v(R) > 0$, $R \succeq_v Q$, as required for condition (S2).

Next, assume w' is a fractionally stable solution. We can assign weights $w_v(P') = w'_v(P)$, $w_v(N) = 1-\sum_{P\in\pi(v)} w_v(P)$. The unity condition ensures that $\sum_{P\in\pi(v)} w'_v(P) \leq 1$, $w_v(N) \geq 0$, and we have a set of weights that sum to 1 for any player v. The tree condition says that $\sum_{P\in\pi(v,S)} w'_v(P) \leq w'_{v'}(S)$ for any $S \in \pi(v')$, which gives $\sum_{P':S\in P'} w_v(P') \leq w_{v'}(S')$, as required for a feasible solution. Finally, we must verify that w_v is a best response for player v. Let w^* be the best response weight function for v, and for the sake of contradiction, assume that w^* gives a better total payoff. Look at the edge P' with the highest $q_v(P)$ such that $w^*_v(P') > w_v(P')$. By nature of the edges we've defined, if $P' \cap Q'$ for $P', Q' \in \pi(v)$, then either $P' \subset Q'$ or $Q' \subset P'$. Therefore, for all edges P'' with $q_v(P'') > q_v(P')$, if $w^*_v(P'') < w_v(P'')$, then we could increase $w^*_v(P'')$ and improve the payoff, so P' is the highest utility edge in which w_v and w^*_v differ. Now look at edge P' with weights w in the FSPP. Since $w^*_v(P') > w_v(P')$ and $w^*_v(P'') = w_v(P'')$ for all P'' with higher payoff than P', for all $S \in P'$ ($s \in \pi(v')$ for some v'), $\sum_{R':S\in R', q_v(R)>q_v(P)} w_v(R') < w_{v'}(S') \Rightarrow \sum_{R\in\pi(v,S), q_v(R)>q_v(P)} w'_v(R) < w'_{v'}(S)$, so condition (S2) is not satisfied. However, since v puts less weight on edges with payoff at least as high as the payoff for P', the total payoff to v is $< q_v(P)+1$. Therefore, $\sum_{R':q_v(R)\geq q_v(P)} w_v(R') < 1$, so $\sum_{R\succeq_v P} w'_v(R) < 1$, so condition (S1) is also not satisfied. This means that w' was not a fractionally stable solution, contradicting our assumption. So w must have been a best response weighting for each v. ∎

5.4.2. Special cases of fractional SPP. Theorem 5.10 together with Theorem 3.3 implies that Fractional SPP is **PPAD**-hard. Therefore, it is natural to next consider special instances that might be easier to solve. For instance, in real-world Internet routing, we would like to see path preferences based primarily on shortest paths. What would happen if we restrict ourselves to path preferences that echo the real world? Unfortunately, by adding appropriate edge lengths to the above reduction, we show that Fractional SPP is **PPAD**-hard even if all preferences are based only on shortest path lengths.

Theorem 5.12. Fractional SPP *is* **PPAD**-*hard even if each node's preference list consists of all paths, ordered shortest to longest based on edge length (where each node defines its own edge lengths, which may not obey the triangle inequality).*

Proof. In the reduction from preference games to Fractional SPP in Theorem 5.10, each path in any preference list has either one hop (a direct path to the destination d) or two hops. For each of these two-hop paths $(i \to j \to d)$, let Q_{ij} = the number of paths P such that $P \geq_i (i \to j \to d)$. Notice that $(i \to j \to d) \geq_i (i \to k \to d)$ if and only if $Q_{ij} \leq Q_{ik}$. Now define the following lengths l on the edges of the graph from the perspective of node i. If $(i \to j \to d) \in \pi(i)$, then $l(i,j) = Q_{ij}$, $l(j,d) = 1$. $l(i,d) = Q_{ii} + 1$. Pick $M_i = \max_j Q_{ij} + 2$. Let $l(x,y) = M_i$ for all other edges. Now, any path $(i \to j \to d) \in \pi(i)$ will have length $Q_{ij} + 1$; path $(i \to d)$ will have length $Q_{ii} + 1$. This preserves the preference list across these paths. Most other paths will have a last segment of length M_i and so will be longer than $l(i,d)$. The only exception is paths that pass through a j such that $(i \to j \to d) \in \pi(i)$. However, for these paths, the only way to arrive at j without following the direct edge (i,j) would be to pass through an edge of length M_i, so these paths too will be longer than $l(i,d)$. ☐

THEOREM 5.13. FRACTIONAL SPP *is* **PPAD**-*hard even if all preferred paths are preference-ordered based on the path length (where each node defines its own distances on the edge lengths, and these distances form a metric and obey the triangle inequality), assuming we may only use edges from a given template graph.*

Proof. This is very similar to the proof of Theorem 5.12. However, in this case, we must remove from the template any edges directly from a node i created in the reduction from Theorem 5.10 to the destination d, since any of these edges would necessarily be a shortest path (and therefore a highest preference path) from the node i to d. Instead, we will add one additional node i' for every $i \neq d$ and replace all paths of the form $(i \to d)$ with a path $(i \to i' \to d)$. We must also remove from the template any edges of the form (x, j') for any $x \neq j$. Otherwise, a path ending in $(x \to j' \to d)$ would be at least as short as the same path ending in $(x \to j \to j' \to d)$, so we would not be able to enforce use of the new edges. Now we will define edge lengths l as follows (from the perspective of a node i).

If $(i \to j \to d) \in \pi(i)$, then $l(i,j) = Q_{ij}$, $l(j,j') = Q_{ij}$, $l(j',d) = 1$. $l(i,i') = 2Q_{ii} + 1$. $l(i',d) = 1$. For two paths $(i \to j \to d) \in \pi(i)$ and $(i \to k \to d) \in \pi(i)$, define $l(j,k) = Q_{ij} + Q_{ik}$. Let $M_i = \max_j 3Q_{ij}$. $l(x,y) = M$ for all other edges (x,y) (excluding the edges that have been removed from the template: (j,d) for all j and (j,k') for all $j \neq k$).

As in the previous proof, the preference order is preserved. However, we must also verify that triangle inequality holds. Clearly, the length 1 edges obey this, since they are the shortest edges in the graph. Consider a length Q_{ij} edge (i,j). Any other path that starts at i and ends at j must either traverse a length M_i edge into j or a length Q_{ij} edge into j, so this is the shortest route from i to j. Consider a length Q_{ij} edge (j,j'). A path that starts at j must traverse either a length M_i edge or a length Q_{ij} edge, so this is also a shortest route. Consider a length $Q_{ij} + Q_{ik}$ edge (j,k). Any path into or out of j must traverse an edge of length M_i or an edge of length Q_{ij}, and likewise for k. Therefore, a path out of j and into k must traverse at least $Q_{ij} + Q_{ik}$. Finally, consider any length M_i edge. At least one end of the edge must be at some x such that $(i \to x \to d)$ is not in $\pi(i)$, and any other edge into or out of this node will also have length M_i. Therefore, the lengths obey the triangle inequality. ☐

Note that, if any edge may be used and if the preferences are based on shortest path lengths for a metric defined for each node, then there is a trivial algorithm for finding an equilibrium: each node follows only the "direct to destination" path. Since a metric must obey the triangle inequality, this path length cannot be strictly longer (cannot be less preferred) than any path including additional nodes.

5.4.3. Approximate fractional SPP. There are two notions of approximation for FSPP: ϵ-solution is defined by Haxell and Wilfong [20], and ϵ-stable solution is defined by Kintali [25]. Below we present their definitions.

ϵ-solution (Haxell and Wilfong [20]). An ϵ-solution is defined as a set $w = \{w_v : v \neq d\}$ of assignments $w_v : \pi(v) \to [0,1]$ satisfying the following:

1. *Unity*: for each node v, $\sum_{P\in\pi(v)} w_v(P) \leq 1$.
2. ϵ-*tree*: for each node v, and each path S with start node u, $\sum_{P\in\pi(v,S)} w_v(P) \leq w_u(S) + \epsilon$.
3. For any node v and path Q starting at v, one of the following holds:
   - $\sum_{P\in\pi(v)} w_v(P) = 1$, and for each P in $\pi(v)$ with $w_v(P) > 0$, $P \geq_v Q$; or
   - there exists a proper suffix S of Q such that $\sum_{P\in\pi(v,S)} w_v(P) = w_u(S) + \epsilon$, where u is the start node of S, and for each $P \in \pi(v,S)$ with $w_v(P) > 0$, $P \geq_v Q$.

In other words, an ε-solution of [20] is a stable solution that overfills each subpath by at most ϵ.

ϵ-FSPP: Given an instance of FSPP, find an ϵ-solution.

Using Scarf's lemma, Haxell and Wilfong [20] proved that every instance of FSPP has an ϵ-solution.

ϵ-*stable solution* (Kintali [25]). An ϵ-stable solution is a feasible solution such that for any node v and path Q starting at v, one of the following holds:

- $1-\epsilon \leq \sum_{P\in\pi(v)} w_v(P) \leq 1$, and for each P in $\pi(v)$ with $w_v(P) > 0$, $P \geq_v Q$; or
- there exists a proper suffix S of Q such that $w_u(S) - \epsilon \leq \sum_{P\in\pi(v,S)} w_v(P) \leq w_u(S)$, where u is the start node of S, and for each $P \in \pi(v,S)$ with $w_v(P) > 0$, $P \geq_v Q$.

In other words, an ϵ-stable solution is a feasible solution that may underfill a higher preference subpath by at most ϵ.

ϵ-STABLE-FSPP: Given an instance of FSPP, find an ϵ-stable solution.

We define a new notion of approximate equilibrium which encompasses both of these previous definitions, i.e., APPROXIMATE-FSPP.

ϵ-*approximate solution*. An ϵ-approximate solution is a solution that satisfies the unity condition, the ϵ-tree condition, and the condition that for any node v and path Q starting at v, one of the following holds:

- $1-\epsilon \leq \sum_{P\in\pi(v)} w_v(P) \leq 1$, and for each P in $\pi(v)$ with $w_v(P) > 0$, $P \geq_v Q$; or
- there exists a proper suffix S of Q such that $w_u(S) - \epsilon \leq \sum_{P\in\pi(v,S)} w_v(P) \leq w_u(S) + \epsilon$, where u is the start node of S, and for each $P \in \pi(v,S)$ with $w_v(P) > 0$, $P \geq_v Q$.

APPROXIMATE-FSPP: Given an instance of FSPP, find an ϵ-stable solution.

Any ε-solution or ε-stable solution is also an ε-approximate solution. Thus, **PPAD**-hardness of APPROXIMATE-FSPP also implies **PPAD**-hardness of both ϵ-FSPP and ϵ-STABLE-FSPP. Also, it is easy to verify that the reduction from Theorem 5.10 reduces finding an ϵ-approximate equilibrium in a preference game to APPROXIMATE-FSPP. Finally, since an exact solution to FRACTIONAL SPP is simultaneously ε-stable, ε-approximate, and an ε-solution, membership in **PPAD** of FRACTIONAL SPP also establishes the membership in **PPAD** of these three problems. (We note that the polynomial-time reduction from ϵ-FSPP to SCARF in [20] gives another proof of **PPAD**-membership of ϵ-FSPP.) We thus obtain the following theorem.

Theorem 5.14. *Approximate-FSPP, ϵ-FSPP, and ϵ-stable-FSPP are* **PPAD**-*complete.*

5.5. Kernels in digraphs. Let $D(V, A)$ be a directed graph. Let $I(v)$ denote the in-neighborhood of a vertex v; i.e., $I(v)$ is v together with the vertices u such that $(u, v) \in A$. A set K of vertices is a clique if every two vertices in K are connected by at least one arc. A set of vertices is called *independent* if no two distinct vertices in it are connected by an arc. A subset of V is called *dominating* if it meets $I(v)$ for every $v \in V$. A *kernel* in D is an independent and dominating set of vertices. A directed triangle shows that not all digraphs have kernels.

A nonnegative function f on V is called *fractionally dominating* if $\sum_{u\in I(v)} f(u) \geqslant 1$ for every vertex v. The function is *strongly dominating* if for all v, $\sum_{u\in K} f(u) \geqslant 1$ for some clique K contained in $I(v)$. A nonnegative function f on V is called *fractionally independent* if $\sum_{u\in K} f(u) \leqslant 1$ for every clique K. A *fractional kernel* is a function on V which is *both* fractionally independent and fractionally dominating. When it is also strongly dominating, it is called a *strong* fractional kernel. As in the integral case, a directed triangle shows that not all digraphs have fractional kernels.

An arc (u, v) is called *irreversible* if (v, u) is not an arc of the graph. A (directed) cycle in D is called *proper* if all of its arcs are irreversible. A digraph in which no clique contains a proper cycle is called *clique-acyclic*. Aharoni and Holzman [2] proved that every clique-acyclic digraph has a strong fractional kernel. We define a computational problem—Strong Kernel: given a clique-acyclic digraph $D(V, E)$ with largest clique of constant size, find a strong fractional kernel. For these graphs, the proof of [2] is a polynomial-time reduction from Strong Kernel to Scarf. Theorem 5.15 establishes **PPAD**-hardness of Strong Kernel.

Strong Kernel: Given a clique-acyclic digraph with the largest clique of constant size, find a fractional weight assignment to the nodes that is fractionally strongly dominating and fractionally independent.

Theorem 5.15. *Degree 3 Preference Game $\leq_P$ Strong Kernel.*

Proof. We are given a preference game over player set $[n]$. We construct the digraph $D = (V, E)$. For each player i, we introduce a vertex $\langle i, i\rangle$ and a vertex $\langle i, j\rangle$ for each j in out(i). We have an edge from $\langle i, j\rangle$ to $\langle i, k\rangle$ if i prefers j over k. For each $\langle i, j\rangle$, $j \neq i$, we also have an additional vertex $I(i, j)$ that has an edge from $\langle j, j\rangle$ and an edge into $\langle i, j\rangle$.

We now claim that the preference game has an equilibrium if and only if D has a strong fractional kernel. Let the preference game have an equilibrium w. Consider the following function f on V. We set $f(\langle i, j\rangle) = w_{ij}$ and $f(I(i, j)) = 1 - f(\langle j, j\rangle)$. We have two kinds of maximal cliques. One kind is the set $\{\langle i, j\rangle\}$ for a given i; we have $\sum_j f(\langle i, j\rangle) = \sum_j w_{ij} = 1$. The other maximal cliques are the edges $(\langle j, j\rangle, I(i, j))$ and $(I(i, j), \langle i, j\rangle)$. Since $f(I(i, j)) = 1 - f(\langle j, j\rangle)$ and $f(\langle i, j\rangle) \leq f(\langle j, j\rangle)$, it follows that f is fractionally independent.

We next show that f is fractionally strongly dominating. For vertex $I(i, j)$, this is immediate since $f(I(i, j)) + f(\langle j, j\rangle) = 1$. Consider a vertex $\langle i, j\rangle$. If $w_{ij} \neq w_{jj}$ (i.e., either $w_{ij} = 0$ or $w_{ij} < w_{jj}$), then the vertex $\langle i, j\rangle$ is covered by the clique consisting of $\langle i, j'\rangle$ over all j' that are at least as preferred to i as j. Otherwise, $w_{ij} = w_{jj}$, in which case $f(I(i, j)) + f(\langle i, j\rangle) = 1$. Thus, f is strongly dominating.

Suppose D has a strong fractional kernel f. We set $w_{ij} = f(\langle i, j\rangle)$. By the fractional independence property applied to the cliques formed by $I(i, j)$ and $\langle i, j\rangle$, we obtain that $w_{ij} \leq w_{jj}$. Consider a vertex $\langle i,j \rangle$. The set of vertices with edges into $\langle i,j \rangle$ is the union of two cliques—the set of $\langle i,k \rangle$ with $k \geq_i j$, and the set $\{I(i,j), \langle i,j \rangle\}$. If $w_{ij} \neq w_{jj}$ (i.e., either $w_{ij} = 0$ or $w_{ij} < w_{jj}$), then the sum of the weights in the first clique is 1; otherwise, the sum of the weights in the second clique is 1, yielding $w_{ij} = w_{jj}$. This establishes the stability of w.

Finally, we note that the graph constructed above satisfies the clique-acyclic property. Consider the two kinds of maximal cliques. In the first kind, consisting of the set $\{\langle i,j \rangle : j \in V\}$ for a given i, a cycle in the graph would imply a cycle in the order $\succ_i$, which is a contradiction. The other maximal cliques are the single edges $(\langle j,j \rangle, I(i,j))$ and $(I(i,j), \langle i,j \rangle$, neither of which can contain a cycle. □

5.6. Fractional bounded budget connection game. We define a fractional variant of the bounded budget connection game, as in [29, 30]. A fractional bounded budget connection game (fractional BBC game) is specified by a tuple $\langle V, d, c, b \rangle$ and a length function ℓ_u for each $u \in V$, where V is a set of nodes, $d \in V$ is a distinguished destination node, and $c : V \times V \to \mathbb{Z}$, $b : V \to \mathbb{Z}$, and $\ell_u : V \times V \to \mathbb{Z}$ (for each $u \in V$) are functions. For any $u, v \in V$, $c(u,v)$ denotes the cost to u of directly linking to v, and $\ell_x(u,v)$ denotes the length of the link (u,v) from the perspective of x if u has established this link. For any node $u \in V$, $b(u)$ specifies the budget u has for establishing outgoing directed links: the sum of the costs of the links established by u times the amount placed on each link should not exceed $b(u)$.

A strategy for node u is a weight function $w_u : V \to [0,1]$ that u places on each outgoing edge $(u,v) : v \in V$ such that $\sum_{(u,v)} c(u,v) \times w_u(v) \leq b(u)$. Let w_u denote a strategy chosen by node u, and let $W = \{w_u : u \in V\}$ denote the collection of strategies. The network formed by W is simply the directed, capacitated complete graph $G(W)$, in which the capacity of the directed edge (u,v) is $w_u(v)$. The utility of a node u is given by $-f(u)$, where $f(u)$ is the cost of a 1-unit minimum-cost flow from u to d, where the edge capacities are given by w_u and the costs are calculated using the length function from the perspective of u. To ensure that a unit flow is always feasible, for any node u that has no edge to destination d, we add an additional edge from u to d, with capacity ∞ and length equal to a large integer $M \gg n \max_{x,u,v} \ell_x(u,v)$; we refer to M as the *disconnection penalty*. In other words, if the maximum flow from u to d in the original network is $\alpha < 1$, then $f(u)$ is the cost of the minimum-cost α flow from u to d plus $(1-\alpha) \cdot M$.

FRACTIONAL BBC: Given a set V of nodes, a destination d, a cost function $c : V \times V \to \mathbb{Z}$, a budget function $b : V \to \mathbb{Z}$, and a length function $\ell_u : V \times V \to \mathbb{Z}$, find a weight assignment $w_u : V \to [0,1]$ for each $u \in V$ such that (a) $\sum_{(u,v)} c(u,v) \times w_u(v) \leq b(u)$, and (b) w_u minimizes the cost of a minimum-cost flow from u to d, assuming the capacity of an edge (x,y) is $w_x(y)$, and the edge costs are given by the ℓ_u function.

THEOREM 5.16. PREFERENCE GAME $\leq_P$ FRACTIONAL BBC.

Proof. We use a similar reduction from a PREFERENCE GAME to FRACTIONAL BBC. Given any instance $\mathbf{P}$ of the PREFERENCE GAME (consisting of a set of players S and a preference relation $\geq_i$ for each $i \in S$), we will create an instance $\mathbf{B}$ of FRACTIONAL BBC $= \langle V, d, c, b \rangle$, where $V = S$; d is an additional node; $\forall i, j \in V$: $c(i,j) = 1$; $\forall i$: $b(i) = 1$. Instance $\mathbf{B}$ has length function l_i for each $i \in V$, defined as follows. Let $p_i(k) =$ the number of j such that $j \geq_i k$. $\forall j \neq i$, $l_i(j,d) = 1$, $l_i(i,j) = p_i(j)$. $\forall j \neq i, k \neq i$, $l_i(j,k) = l_i(k,j) = |S| + 1$. $l_i(i,d) = 1 + p_i(i)$. Given a solution to **B**, define a solution to **P** by setting $w_i(j) =$ the weight placed on edge (i, j) (for $j \neq i$), and $w_i(i) =$ the weight placed on edge (i, d).

Since the total cost for all edges is 1, and the total budget for a node is 1, each node in **B** will place total weight 1 on edges adjacent to it. This exactly corresponds to the requirement that $\sum_j w_i(j) = 1$ in **P**. The possible paths for a 1-unit flow from i to d in **B** are (1) the path consisting of only edge (i, d), which has cost $p_i(i)+1 \leq |S|+1$, (2) a path of the form (i, j, d) through some other node j, which has cost $p_i(j)+1 \leq |S|+1$, or (3) a path including some edge (j, k) for $j \neq i, k \neq i$, which has cost $> |S| + 1$. Therefore, a minimum-cost flow will use only paths of the form (i, d) and (i, j, d), so the requirement in **P** that $w_i(j) \leq w_j(j)$ corresponds to using the weight j places on edge (j, d) as a capacity on that edge when finding the minimum-cost flow. Now, we need only show that a node's best response in **B** exactly corresponds to a lexicographically maximal weight assignment in **P**.

Suppose we have a best response for node i in **B** that corresponds to a weight assignment w in **P** that is not lexicographically maximal for i. Then there is some assignment $w' = w'_i \cup \{w_j : j \neq i\}$ such that for some $j \in S$, $\sum_{k \geq_i j} w_i(k) < \sum_{k \geq_i j} w'_i(k)$. There must be some $k^+ \in S$ such that $k^+ \geq_i j$ and $w'_i(k^+) > w_i(k^+)$, and there must be some $k^- \in S$ such that $\neg(k^- \geq_i j)$ and $w'_i(k^-) < w_i(k^-)$. Suppose we move ϵ weight in the best response in **B** from P_{ik^-} to P_{ik^+}. $p_i(k^-) > p_i(k^+)$, so moving this weight will decrease the cost of a minimum-cost flow, contradicting the fact that this was a best response.

Suppose we have a lexicographically maximal weight assignment w for **P** that does not correspond to a best response for node i in **B**. Then, in **B**, i could move weight from some path P_{ij} to a different path P_{ik} to decrease the cost of its minimum-cost flow. This means that $p_i(k) < p_i(j)$, or the number of nodes preferred by i over k is smaller than the number of nodes preferred by i over j. Since preference relations are transitive, this implies that $k \geq_i j$. However, since P_{ik} has space left, $w_i(k) < w_k(k)$, so w is not lexicographically maximal. □

In the Fractional BBC instance constructed in the reduction in the proof of Theorem 5.16, the set of paths considered in the minimum-cost flow calculation is polynomial in size; in fact, each path has at most two hops. Thus, the **PPAD**-hardness of Fractional BBC holds even if we have an explicit set of flow paths specified in the problem instance, or if we place a constant bound on the number of hops in the flow path. (The latter special case is, in fact, subsumed by the former since the number of flow paths of constant length is polynomial and hence can be explicitly specified.)

We are able to place Fractional BBC in **PPAD** in the special case where the set of flow paths is explicitly given.

Theorem 5.17. *There is a polynomial-time reduction from* Fractional BBC*, restricted to instances where the set of flow paths is explicitly given, to* Personalized Equilibrium.

Proof. Consider any instance of fractional BBC, where the set of flow paths is given. Create a player in the Personalized Equilibrium game for each node in the BBC instance. The strategy set S_i for player i contains one action for each available outgoing edge for i in the instance. For each player i, for each given flow path for i, we add a hyperedge to E_i consisting of all the edges in the path. For any hyperedge in the Personalized Equilibrium game, a player's payoff is equal to the negative of the length of the path represented by that hyperedge. For each player i, the LP then computes a weight assignment w_i that achieves the minimum-cost flow using the given flow paths where the capacity constraints are given by the weights and the costs are given by the length function ℓ_i. Thus, the set of equilibria for the fractional BBC instance are exactly the same as the set of equilibria for the PERSONALIZED EQUILIBRIUM instance. This is clearly a polynomial-time reduction. ∎

Acknowledgments. We would like to thank H. Venkateswaran for helpful discussions, Gordon Wilfong for sending a preprint of [21], Christos Papadimitriou for pointing us to [45], and David Karger for his insightful motivation for the preference game. Finally, we are greatly indebted to the anonymous referees, whose detailed and insightful comments have significantly improved both the technical quality and the presentation of the paper.

REFERENCES

[1] R. AHARONI AND T. FLEINER, *On a lemma of Scarf*, J. Combin. Theory Ser. B, 87 (2003), pp. 72–80.

[2] R. AHARONI AND R. HOLZMAN, *Fractional kernels in digraphs*, J. Combin. Theory Ser. B, 73 (1998), pp. 1–6.

[3] R. J. AUMANN, *Subjectivity and correlation in randomized strategies*, J. Math. Econom., 1 (1974), pp. 67–96.

[4] V. BALA AND S. GOYAL, *A noncooperative model of network formation*, Econometrica, 68 (2000), pp. 1181–1229.

[5] O. N. BONDAREVA, *Some applications of the methods of linear programming to the theory of cooperative games*, Problemy Kibernet., No. 10 (1963), pp. 119–139.

[6] X. CHEN AND X. DENG, 3-*NASH Is PPAD-Complete*, Report TR05-134, Electronic Colloquium on Computational Complexity (ECCC), 2005.

[7] X. CHEN AND X. DENG *Settling the complexity of two-player Nash equilibrium*, in Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 2006, pp. 261–272.

[8] X. CHEN, X. DENG, AND S.-H. TENG, *Computing Nash equilibria: Approximation and smoothed complexity*, in Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 2006, pp. 603–612.

[9] X. CHEN, X. DENG, AND S.-H. TENG, *Settling the complexity of computing two-player Nash equilibria*, J. ACM, 56 (2009), 14.

[10] V. CONITZER AND T. SANDHOLM, *Complexity of determining nonemptiness of the core*, in Proceedings of the 4th ACM Conference on Electronic Commerce, 2003, pp. 230–231.

[11] V. CONITZER AND T. SANDHOLM, *Complexity of constructing solutions in the core based on synergies among coalitions*, Artificial Intelligence, 170 (2006), pp. 607–619.

[12] G. DANTZIG, *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ, 1998.

[13] C. DASKALAKIS, P. W. GOLDBERG, AND C. H. PAPADIMITRIOU, *The complexity of computing a Nash equilibrium*, SIAM J. Comput., 39 (2009), pp. 195–259.

[14] A. FABRIKANT, A. LUTHRA, E. MANEVA, C. H. PAPADIMITRIOU, AND S. SHENKER, *On a network creation game*, in Proceedings of the Twenty-Second Annual ACM Symposium on Principles of Distributed Computing (PODC), ACM, New York, 2003, pp. 347–351.

[15] Q. FANG, R. FLEISCHER, J. LI, AND X. SUN, *Algorithms for core stability, core largeness, exactness, and extendability of flow games*, in Proceedings of the 13th Annual International Conference on Computing and Combinatorics (COCOON), Springer-Verlag, Berlin, Heidelberg, 2007, pp. 439–477.

[16] D. GALE AND L. S. SHAPLEY, *College admissions and the stability of marriage*, Amer. Math. Monthly, 69 (1962), pp. 9–14.

[17] R. GARG, V. KUMAR, A. RUDRA, AND A. VERMA, *Coalitional games on graphs: Core structure, substitutes and frugality*, in Proceedings of the 4th ACM Conference on Electronic Commerce (EC), 2003, pp. 248–289.

[18] P. W. GOLDBERG AND C. H. PAPADIMITRIOU, *Reducibility among equilibrium problems*, in Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC), 2006, pp. 61–70.

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

[20] P. E. HAXELL AND G. T. WILFONG, *A fractional model of the Border Gateway Protocol (BGP)*, in Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2008, pp. 193–199.

[21] P. E. Haxell and G. T. Wilfong, *On the stable paths problem*, SIAM J. Discrete Math., 24 (2010), pp. 1137–1152.
[22] N. Immorlica, K. Jain, and M. Mahdian, *Game-theoretic aspects of designing hyperlink structures*, in Internet and Network Economics, Lecture Notes in Comput. Sci. 4286, Springer-Verlag, Berlin, Heidelberg, 2006, pp. 150–161.
[23] M. Jackson and A. Wolinsky, *A strategic model of social and economic networks*, J. Econom. Theory, 71 (1996), pp. 44–74.
[24] M. Kearns, M. L. Littman, and S. Singh, *Graphical models for game theory*, in Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence (UAI), Morgan Kaufmann, San Francisco, 2001, pp. 253–260.
[25] S. Kintali, *A distributed protocol for fractional stable paths problem*, in Proceedings of the DIMACS/DyDAn Workshop on Secure Internet Routing, Rutgers University, Piscataway, NJ, 2008.
[26] S. Kintali, L. J. Poplawski, R. Rajaraman, R. Sundaram, and S.-H. Teng, *Reducibility among fractional stability problems*, in Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 2009, pp. 283–292.
[27] T. Király and J. Pap, *Kernels, stable matchings, and Scarf's lemma*, in Combinatorial Optimization and Discrete Algorithms, RIMS Kôkyûroku Bessatsu B23, Res. Inst. Math. Sci., Kyoto, Japan, 2010, pp. 131–145.
[28] M. Kumabe and H. R. Mihara, *Computability of simple games: A characterization and application to the core*, J. Math. Econom., 44 (2008), pp. 348–366.
[29] N. Laoutaris, L. J. Poplawski, R. Rajaraman, R. Sundaram, and S.-H. Teng, *Bounded budget connection (BBC) games or how to make friends and influence people, on a budget*, in Proceedings of the Twenty-Seventh ACM Symposium on Principles of Distributed Computing (PODC), 2008, pp. 165–174.
[30] N. Laoutaris, L. J. Poplawski, R. Rajaraman, R. Sundaram, and S.-H. Teng, *Bounded Budget Connection (BBC) Games or How to Make Friends and Influence People, on a Budget*, preprint, arXiv:0806.1727v1 [cs.GT], 2008.
[31] C. E. Lemke and J. T. Howson, Jr., *Equilibrium points of bimatrix games*, J. Soc. Indust. Appl. Math., 12 (1964), pp. 413–423.
[32] E. Markakis and A. Saberi, *On the core of the multicommodity flow game*, in Proceedings of the 4th ACM Conference on Electronic Commerce, 2003, pp. 93–97.
[33] N. Megiddo and C. H. Papadimitriou, *On total functions, existence theorems and computational complexity*, Theoret. Comput. Sci., 81 (1991), pp. 317–324.
[34] J. F. Nash, Jr., *Equilibrium points in n-person games*, Proc. Natl. Acad. Sci. USA, 36 (1950), pp. 48–49.
[35] J. Nash, *Non-cooperative games*, Ann. of Math. (2), 54 (1951), pp. 286–295.
[36] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*, Cambridge University Press, New York, 2007.
[37] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*, MIT Press, Cambridge, MA, 1994.
[38] C. H. Papadimitriou, *On the complexity of the parity argument and other inefficient proofs of existence*, J. Comput. System Sci., 48 (1994), pp. 498–532.
[39] Y. Rehkter and T. Li, *A Border Gateway Protocol* 4 *(BGP-*4*)*, RFC 1771, Network Working Group, http://www.ietf.org/rfc/rfc1771.txt (1995).
[40] H. E. Scarf, *The core of an N person game*, Econometrica, 69 (1967), pp. 35–50.
[41] A. Schrijver, *Combinatorial Optimization. Polyhedra and Efficiency*, Vol. B, Springer-Verlag, Berlin, 2003.
[42] L. S. Shapley, *On balanced sets and cores*, Naval Res. Logist. Quarter., 14 (1967), pp. 453–460.
[43] L. S. Shapley, *A note on the Lemke-Howson algorithm. Pivoting and extensions*, Math. Programming Stud., No. 1 (1974), pp. 175–189.
[44] J. W. Stewart III, *BGP*4: *Inter-Domain Routing in the Internet*, Addison–Wesley, Boston, 1998.
[45] M. J. Todd, *Orientation in complementary pivot algorithms*, Math. Oper. Res., 1 (1976), pp. 54–66.
[46] K. Varadhan, R. Govindan, and D. Estrin, *Persistent route oscillations in inter-domain routing*, Comput. Networks, 32 (2000), pp. 1–16.