Egerváry Research Group
on Combinatorial Optimization

Technical reportS

TR-2012-18. Published by the Egerváry Research Group, Pázmány P. sétány 1/C, H–1117, Budapest, Hungary. Web site: www.cs.elte.hu/egres . ISSN 1587–4451.

Complexity of equilibria in linear service-providing games

Tamás Király and Júlia Pap

November 2012
Revised March 2014

Complexity of equilibria in linear service-providing games

Tamás Király* and Júlia Pap**

Abstract

A fundamental problem in algorithmic game theory is to determine the hardness of computing equilibria in various classes of games. It is difficult to pinpoint the properties that affect hardness, since computing an equilibrium seems to be hard even for 2-player games with very special payoff matrices. To address this problem, we introduce an LP-based model (a special case of the Generalized Nash Equilibrium Problem) where the interaction of players is limited to providing services to each other at fixed prices. This enables us to investigate computational complexity as a function of the structure of provider-customer relationships.

Our model guarantees the existence of equilibria with polynomial bit-size. If the service prices are optimal in some sense, then socially optimal solutions are in equilibrium and can be computed in polynomial time. However, for arbitrary service prices, the computation of an equilibrium is PPAD-complete. To study the complexity in terms of the structure of interactions between players, we introduce a digraph describing the provider-customer relatioships, and give a polynomial-time algorithm if every strong component of the digraph is a simple directed cycle. The proof is based on a new algorithmic result on approximating a fixed point of a multidimensional function having a cyclic structure.

1 Introduction

The complexity of computing equilibria is a fundamental problem of algorithmic game theory. A sequence of results, culminating in the breakthough paper of Chen, Deng and Teng [6], showed that Nash-equilibrium is PPAD-hard to approximate, even for 2-players, sparse payoff matrices [6], and 0-1 payoff [1]. However, there are important classes of problems, like some market equilibrium problems [18, 19] and certain types of congestion games [2], where an equilibrium can be computed efficiently. For detailed surveys on these issues, see [10, 17].

One approach to capture the structural properties of interactions in games is to consider graphical games, where each player has only 2 strategies, and the payoff of

*MTA-ELTE Egerváry Research Group, Eötvös University, Budapest. Email: tkiraly@cs.elte.hu

**Department of Operations Research, Eötvös University, Budapest. Email: papjuli@cs.elte.hu

a player depends only on the strategies of its neighbours (and himself) in a given graph. It was shown by Elkind, Goldberg and Goldberg [8] that Nash equilibrium is computable in polynomial time if the graph has maximum degree 2, but it is PPAD-complete for maximum degree 3 and constant pathwidth.

In this paper we follow a different approach to study the relationship between the structure of interactions and the complexity of computing equilibria. We consider a special case of the so-called Generalized Nash Equilibrium Problem that was introduced by Arrow and Debreu [4] (under the name of "abstract economy"). In our problem the strategy space of each player is described by a linear program, and interactions between players are restricted to services being provided at fixed prices, expressed by linear equations. Our model guarantees the existence of equilibria that are vertices of a given polyhedron, like in the case of 2-player Nash equilibrium. We prove that finding equilibria in this model is also PPAD-complete, but there is a polynomial-time algorithm if the structure of services is sufficiently simple. We also show that under some conditions it is possible to modify service prices in such a way that socially optimal solutions (which can be computed in polynomial time) become equilibria.

Our results generalize previous work on multicommodity flow games. The study of flow problems with selfish agents dates back to the papers of Kalai and Zemel [13, 14], and the multicommodity version was studied by Derks and Tijs [7]. Agarwal and Ergun [3] introduced multicommodity flow games where each agent owns a given fraction of each arc of the network. An agent also has a set of demands, and revenue is generated by satisfying a fraction of these demands. Instead of the usual concept of transferable utilites, only a restricted form of transfer is allowed: agents pay specified capacity exchange prices for the use of arc capacities, divided proportionally among the owners of the arc. In [3], a method is proposed for determining capacity exchange prices which provide incentives for agents to route according to the optimal flow. The results were further generalized in [11].

Another precedent to our model is the multiplayer multicommodity flow problem introduced by the present authors with co-authors [5]. As in the aforementioned model, each agent controls a subnetwork, but players have hard demands that must be satisfied. Instead of the exchange of arc capacities, there are bilateral agreements specifying source–destination pairs between which one agent undertakes to route the traffic of another in exchange for a specified per-unit payment.

As we show in Section 1.5, the LP-based model considered in this paper includes the above problems. From another viewpoint, it can be seen as a linear special case of the Generalized Nash Equilibrium Problem, where the available strategies of a given player can vary depending on how we fix the strategies of the other players (see [9] for a comprehensive survey on generalized Nash equilibrium). We note that these types of problems are sometimes referred to as "pseudo-games", but we will call them games in this paper. The advantage of linearity is that best-response strategies can always be computed in polynomial time by linear programming, and equilibria can be characterized as certain faces of a bounded polyhedron.

1.1 Strategies and costs

The game involves a set of *players*, denoted by $[n] = \{1, \ldots, n\}$, and a set S of *services*. Each service may have several providers and several customers. The set of services where player i is a possible provider (resp. customer) is denoted by S^i (resp. T^i). We do not require S^i and T^i to be disjoint, unless explicitly stated. Each provider of a service is required to contribute a fixed proportion of the service. These *service ratios* are given as non-negative rationals r_s^i for every service s and agent i with $s \in S^i$, satisfying $\sum_{i:s\in S^i} r_s^i = 1$.

The possible strategies of player i form a polyhedron that is defined using three types of variables:

- x_s^i, for $s \in S^i$, represents the amount of service s provided,
- y_s^i, for $s \in T^i$, is the amount bought of service s,
- z^i is a vector of additional variables.

The strategy space of player i is a bounded polyhedron P^i of the form

$$A_1^i x^i + A_2^i y^i + A_3^i z^i \leq b^i, \tag{1}$$

$$x^i \geq \mathbf{0}, \tag{2}$$

$$y^i \geq \mathbf{0}, \tag{3}$$

where the matrices A_1^i, A_2^i, A_3^i can be arbitrary except that A_1^i is assumed to be non-negative. There is also a cost vector c^i of the same dimension as (x^i, y^i, z^i).

Let $P_\times$ denote the direct product $P^1 \times \cdots \times P^n$, the combined strategy space. The *social cost* of a strategy vector $(x, y, z) \in P_\times$ is $c^\mathsf{T}(x, y, z)$. We use the notation c_s^i for the cost of variable x_s^i.

Service s has a per-unit *service price* p_s. A customer j with $s \in T^j$ pays $p_s y_s^j$ for service s, and the income of provider i from service $s \in S^i$ is $p_s x_s^i$.

1.2 Feasible and weakly feasible solutions

Strategy vectors in $P_\times$ do not necessarily adhere to the given service ratios; they do not even satisfy the condition that the total amount provided of a given service should equal the total amount bought. These conditions are captured by the notions of feasibility and weak feasibility. A strategy vector $(x, y, z) \in P_\times$ is *feasible* if

$$x_s^i = r_s^i \sum_{j:s\in T^j} y_s^j \quad \text{for every } i \text{ and } s \in S^i.$$

The equation is called the *personal service equation* of agent i for service s. We also use a weaker version of these conditions: strategy $(x, y, z) \in P_\times$ is *weakly feasible* if the following *weak feasibility equations* hold:

$$\sum_{i:s\in S^i} x_s^i = \sum_{j:s\in T^j} y_s^j \quad \text{for every } s \in S.$$

The polyhedron of feasible solutions is denoted by P_{feas}, while the polyhedron of weakly feasible solutions is P_{weak}. Clearly, weakly feasible solutions are feasible for some service ratios, but not necessarily for the prescribed ones.

A *socially optimal solution* is a feasible solution that is optimal for the cost vector c in P_{feas}. We say that r is a *socally optimal service ratio vector* if socially optimal solutions are also optimal in P_{weak} for c. In other words, the service ratios are socially optimal if no other service ratios can achieve lower social cost.

Given the service prices p, we define a *modified cost vector* c_p by decreasing the cost of variable x^i_s by p_s for all $s \in S^i$, and increasing the cost of variable y^i_s by p_s for all $s \in T^i$. Using this notation, the personal interest of player i is to minimize c^i_p on P^i. Of course, if each player optimizes independently on their polyhedra, then the resulting vector is typically not feasible.

Not every instance of the problem has a feasible or even a weakly feasible solution. However, there is a natural condition that turns out to be sufficient for feasibility. An instance of the problem is called *safe* if for any $(x, y, z) \in P_\times$ and any $i \in [n]$ we can replace (x^i, y^i, z^i) by a vector $(\overline{x}^i, \overline{y}^i, \overline{z}^i) \in P^i$ such that the resulting set of vectors satisfies all personal service equations of agent i. We will see that safe instances have solutions in equilibrium, which implies that they have feasible solutions.

1.3 Equilibria and perfect prices

A feasible solution is an equilibrium if for every i it is true that if the solutions of other players are fixed, then player i cannot increase his profit without violating one of his personal service equations. More formally, $(x, y, z) \in P_{\text{feas}}$ is an *equilibrium* if there is no $i \in [n]$ and $(\overline{x}^i, \overline{y}^i, \overline{z}^i) \in P^i$ such that

$$c_p^{i\mathsf{T}}(\overline{x}^i, \overline{y}^i, \overline{z}^i) < c_p^{i\mathsf{T}}(x^i, y^i, z^i)$$

and the set of vectors obtained by replacing (x^i, y^i, z^i) by $(\overline{x}^i, \overline{y}^i, \overline{z}^i)$ satisfies all personal service equations of agent i. This corresponds to the notion of generalized Nash equilibrium (see [9]), because if we fix the strategies of all players except i, then the set of available strategies of player i can be defined as the set of elements in P^i that satisfy the personal service equations of i with respect to the fixed strategies of others.

A service price vector p is called *perfect* for a feasible solution (x, y, z) if the vector (x, y, z) is optimal for the cost vector c_p in the polyhedron $P_\times$. The motivation for this definition is that the following all hold if p is perfect for (x, y, z):

- (x, y, z) is an equilibrium if the service prices are set at p (in fact, players cannot improve even by violating their own personal service equations),
- no player (or coalition) can increase its profit by terminating or restricting a service,
- no player (or coalition) can increase its profit by persuading others to purchase more of a service.

In other words, if prices are perfect for a solution, then the solution is in equilibrium, and the prices are high enough so that players are not motivated to restrict a service, but also low enough so that no one is motivated to decrease prices in order to get more customers.

An easy observation is that perfect prices are possible only for socially optimal solutions and only for socially optimal service ratios. Indeed, the c_p-cost of a *weakly feasible* solution is the same as its c-cost, so a weakly feasible solution that is not optimal for c in P_{weak} cannot be optimal in $P_\times$ with respect to c_p.

1.4 Summary of the results

In Section 2 we show that if the service ratios are socially optimal, then there exist service prices p that are perfect for every socially optimal feasible solution. Our theorem is a generalization of the result of Agarwal and Ergun [3] on capacity exchange prices in service network alliances (see section 1.5.1).

Theorem 1.1. *Suppose that the service ratios are socially optimal. Then it is possible to compute service prices p in polynomial time that are perfect for every socially optimal solution $(x, y, z) \in P_{\text{feas}}$. If $c_s^i \geq 0$ for every i, then the price p_s can be chosen to be nonnegative.*

In general, it is not even true that we can always find prices for which there exists an equilibrium. Nevertheless, we can prove the following.

Theorem 1.2. *If $S^i \cap T^i = \emptyset$ for every player i, then there are prices for which all socially optimal solutions are in equilibrium.*

In Section 3 we investigate the structure and existence of equilibria. We first show that the set of equilibria is always the (perhaps empty) union of some faces of P_{feas}, and prove the following existence result.

Theorem 1.3. *In a safe instance there always exists an equilibrium.*

This theorem can be derived from earlier results on the existence of generalized Nash equilibria, but we prefer to give a simple direct proof using Kakutani's fixed point theorem.

In Section 4 we prove that finding an equilibrium is PPAD-complete – even in the Multiplayer Multicommodity Flow Problem, described in Section 1.5.2, where each service has one provider and one customer.

Theorem 1.4. *It is PPAD-complete to find an equilibrium in safe instances of the MMF problem.*

As a consequence, it is probably not possible to devise a mechanism that steers players quickly to an equilibrium, since the computation of an equilibrium is intractable unless all problems in PPAD can be solved in polynomial time.

In Section 5 we present a polynomial-time algorithm for finding an equilibrium in the special case when each strong component of the digraph representing the provider-customer relationships is a simple cycle. To be more precise, this auxiliary directed

graph $D^* = ([n], A^*)$ has arcs from the providers of each service to the customers, with possible parallel arcs but excluding loops. A pair of oppositely directed arcs is also considered as a simple cycle.

Theorem 1.5. *An equilibrium can be found in polynomial time in safe instances where every strong component of D^* is a simple directed cycle.*

In view of this result, the study of other special cases might offer new insights on the borderline between polynomially solvable and PPAD-complete equilibrium problems. For example, it is open whether there is a polynomial algorithm in case of 3 services, with 1 provider and 1 customer for each.

Our polynomial algorithm can be seen as a more general result on the approximation of fixed points in a certain class of fixed-point problems. Given m interval-valued mappings $\varphi_1, \ldots, \varphi_m$ on the unit interval, all with the closed graph property, Kakutani's fixed point theorem implies that there is a vector x such that $x_{i+1} \in \varphi_i(x_i)$ $(i = 1, \ldots, m-1)$ and $x_1 \in \varphi_m(x_m)$ (a *cyclically fixed vector*). We show an algorithm for finding m arbitrarily small intervals such that their direct product contains a cyclically fixed vector.

Theorem 1.6. *Let $\varphi_i : [0,1] \to \mathcal{P}([0,1])$ $(i \in [m])$ be given as above with a function evaluation oracle, and let $0 < \varepsilon < 1$. In $O(m^2 \log(\frac{1}{\varepsilon}))$ steps we can find intervals $I_1, \ldots, I_m \subseteq [0,1]$ of length at most ε such that there is a cyclically fixed vector x with $x_i \in I_i$ $(i \in [m])$.*

1.5 Relation to multicommodity flow games

In this section we briefly describe how our model generalizes previous work on games involving multicommodity flows and services.

1.5.1 Service network alliances

Agarwal and Ergun [3] gave the following model for multicommodity flow games in service network alliances. There are n agents, and a common directed graph $D = (V, A)$. Each arc $a \in A$ has a capacity u_a, and given ownership ratios r_a^i $(i \in [n])$ with $\sum_{i=1}^n r_a^i = 1$. Each agent i has a demand set Q^i, where a demand $q \in Q^i$ is characterized by a source s_q, a sink t_q, a per-unit revenue w_q, and an upper bound u_q. Agent i should have a flow of size $f_q \le u_q$ from s_q to t_q; the revenue generated by this flow is $w_q f_q$. The total flow traversing an arc a should not exceed the capacity u_a.

The main result of [3] is a mechanism that distributes the benefits of collaboration by assigning a capacity exchange price p_a to each arc $a \in A$. If the total flow of agent i on arc A is f_a^i, then i has to pay an amount of $p_a f_a^i$, which is distributed among the agents according to the ratios r_a^j $(j \in [n])$. Since the payment to herself can be ignored, agent i actually pays an amount of $(1 - r_a^i) p_a f_a^i$ to the other agents. The paper shows that, given a socially optimal multicommodity flow f^*, it is possible to compute capacity exchange prices with the property that no agent is motivated to deviate from f^*.

We can model this problem in the framework of the present paper by assigning a service to each arc $a \in A$, i.e. $S = A$. All agents are potential customers of all services, while the providers of service a are the agents with $r_a^i > 0$. The service ratios are determined by the values r_a^i. In order to define the polyhedron P^i, we consider the vector variable z^i to be composed of vector variables z_q^i for each demand $q \in Q^i$. In P^i, the variables z_q^i ($q \in Q^i$) describe the multicommodity flow polyhedron of agent i, with cost defined as the opposite of revenue. The variable y_a^i equals the total flow of agent i on arc a (as expressed by the appropriate z variables), and it is upper bounded by u_a. The variable x_a^i has the single constraint $0 \leq x_a^i \leq r_a^i u_a$; the variables x and y have cost 0.

With these definitions, feasibity is the same as in the original problem, and the c_p^i-cost of agent i represents his total cost in the original problem. It is easy to see that the service ratios are always socially optimal in this case. Indeed, the only conditions on the variables x are $0 \leq x_a^i \leq r_a^i u_a$ for every i and a, so any weakly feasible solution can be transformed into a feasible solution of the same social cost by appropriately modifying the values of the x variables without changing $\sum_{i:r_a^i>0} x_a^i$ for any arc a. Therefore Theorem 1.1 implies the result of Agarwal and Ergun on capacity exchange prices.

1.5.2 The Multiplayer Multicommodity Flow (MMF) Problem

In the MMF problem defined in [5], n agents have separate networks $D^i = (V, A^i)$ on a common node set. Each arc a has a cost c_a and a capactity u_a. Each agent i has a set Q^i of hard demands. A demand $q \in Q^i$ is characterized by a source s_q, a sink t_q and a size d_q. The aim of agent i is to satisfy all his demands at minimum cost. A subset of arcs $B^i \subseteq A^i$ are so-called *contractual arcs*, each with a designated agent called the provider. A contractual arc has a price p_a and a multiplier γ_a. If $a = uv \in B^i$ and the total flow of agent i on a is f_a^i, then an additional demand of size $\gamma_a f_a^i$ from u to v appears in the network of the provider of a (a *contractual demand*). In exchange, i pays an amount of $f_a^i p_a$ to the provider. It is proved in [5] that in safe instances of the problem there always exists an equilibrium.

The MMF problem can be modeled in our framework by assigning a service to each contractual arc, thus each service has one customer and one provider (hence the service ratios are trivial, and socially optimal). Let R^i be the set of contractual demands of agent i. We consider the vector variable z^i to be composed of vector variables z_q^i for each demand $q \in Q^i \cup R^i$. In P^i, the variables z_q^i describe the multicommodity flow polyhedron of agent i, with the modification that if $q \in R^i$, then the size of demand q is $\gamma_{a(q)} x_{a(q)}^i$, where $a(q)$ is the contractual arc corresponding to q. The variable y_a^i for a contractual arc $a \in B_i$ is equal to the total flow of agent i on arc a.

2 Existence of perfect prices

In this section we prove Theorems 1.1 and 1.2, and give an example where no prices can guarantee an equilibrium. We start by explaining the polyhedral tools used in

the proof.

For a polyhedron P and a face F of P, let opt.cone(F, P) denote the set of objective vectors c for which every point of F is optimal in P, that is, the *optimal cone* of F in P. The *tangent cone* of a point in P is the set of feasible directions from the point. The relative interior of a set $X \subseteq \mathbb{R}^n$ is denoted by relint(X), while lin$X$ is the linear translation of the affine hull of X. We will need the following consequence of Farkas' Lemma for optimal cones.

Lemma 2.1. *Let P_1 be a polyhedron, Π an affine subspace and $P_2 = P_1 \cap \Pi$. Let F_2 be a face of P_2 and let F_1 be the smallest face of P_1 that contains F_2. Then*

(i) $\text{opt.cone}(F_2, P_2) = \text{opt.cone}(F_1, P_1) + (\text{lin}\Pi)^\perp$,

(ii) $\text{relint}(\text{opt.cone}(F_2, P_2)) = \text{relint}(\text{opt.cone}(F_1, P_1)) + (\text{lin}\Pi)^\perp$.

Proof. To prove the "$\supseteq$" containment in part (i), let $w \in \text{opt.cone}(F_1, P_1)$ and $a \in (\text{lin}\Pi)^\perp$. Then for any $x \in F_2$ and $x' \in P_2$, $(w+a)x = wx+ax \geq wx'+ax = (w+a)x'$, so $w + a \in \text{opt.cone}(F_2, P_2)$.

For the "$\subseteq$" containment suppose that $w \in \text{opt.cone}(F_2, P_2)$ but w is not in the cone $\text{opt.cone}(F_1, P_1) + (\text{lin}\Pi)^\perp$. By Farkas' Lemma the latter implies that there is a vector y for which $wy > 0$ but $(w' + a)y \leq 0$ for every $w' \in \text{opt.cone}(F_1, P_1)$ and $a \in (\text{lin}\Pi)^\perp$. Clearly $y \in \text{lin}\Pi$. Let x^* be a vector in relint(F_2). Then opt.cone(F_1, P_1) is generated by the normal vectors of the facets of P_1 that x^* satisfies with equality. This means that y is in the tangent cone of P_1 in x^*, thus, since $y \in \text{lin}\Pi$, y is also in the tangent cone of P_2 in x^*. This contradicts $wy > 0$.

Part (ii) follows from part (i). □

We are ready to show that if the service ratios are socially optimal, then there is a price vector that is perfect for every socially optimal solution.

Proof of Theorem 1.1. We assume that P_{feas} is non-empty. Let F_{opt} be the optimal face in P_{weak} minimizing the cost c. By the assuption that service ratios are optimal, all socially optimal solutions are on F_{opt}. Let $F_\times$ be the minimal face of $P_\times$ which contains F_{opt}. Let furthermore H be the subspace determined by the equations

$$\sum_{i:s\in S^i} x_s^i = \sum_{j:s\in T^j} y_s^j \quad \text{for every } s \in S.$$

We apply Lemma 2.1 with $P_1 = P_\times$, $\Pi = H$, $P_2 = P_{\text{weak}}$, $F_1 = F_\times$, and $F_2 = F_{\text{opt}}$. By consequence, there is a vector $h \in (\text{lin}H)^\perp$ for which $c + h$ is a member of relint$(\text{opt.cone}(F_\times, P_\times))$. Note that h can be computed in polynomial time by linear programming. By the definition of H, there is a vector $p \in \mathbb{R}^S$ such that

- The component of h corresponding to x_s^i is $-p_s$,
- the component of h corresponding to y_s^i is p_s,
- the components of h corresponding to z are 0.

Let the price of service s be p_s. Since $c+h = c_p$, it follows that any socially optimal solution $(x, y, z) \in F_{\text{opt}} \subseteq F_\times$ is optimal for objective function c_p in the polyhedron $P_\times$. That is, p is perfect for any socially optimal solution.

To prove the second part of the theorem, assume that c_s^i is nonnegative for every i. Suppose that p_s is negative, and let (x, y, z) be a socially optimal solution. Since $c_s^i - p_s$ is positive, we decrease the c_p-cost by decreasing x_s^i. Since the describing matrices A_1^i are nonnegative, the modified vector is also in $P_\times$ if $x_s^i \geq 0$. Therefore $p_s < 0$ implies that $x_s^i = \mathbf{0}$ for every i in every socially optimal solution. We claim that if p' is obtained from p by setting $p'_s = 0$, then p' is also perfect. Indeed, there is a vector (x', y', z') of minimum $c_{p'}$-cost in $P_\times$ for which ${x'}_s^i = 0$ for every i, because we can decrease ${x'}_s^i$ without increasing the $c_{p'}$-cost. Now $c_p^\mathsf{T}(x', y', z') \leq c_{p'}^\mathsf{T}(x', y', z')$ by the definition of p'. On the other hand, the c_p-cost of a socially optimal solution (x, y, z) is the same as its $c_{p'}$-cost because $x_s^i = 0$ and $y_s^i = 0$ for every i, hence $c_{p'}^\mathsf{T}(x, y, z) = c_p^\mathsf{T}(x, y, z) \leq c_p^\mathsf{T}(x', y', z') \leq c_{p'}^\mathsf{T}(x', y', z')$, which means that (x, y, z) is is optimal for $c_{p'}$ in $P_\times$. □

We now show that if the service ratios are not socially optimal, then in general we cannot even expect to find prices that guarantee the existence of an equilibrium. Consider the following example with two services and two players, each player being a customer and a provider of both services. For sake of simplicity, we use the notation $S = \{1, 2\}$, so we have variables $x_1^1, x_2^1, x_1^2, x_2^2$ and $y_1^1, y_2^1, y_1^2, y_2^2$. The possible strategies of the two players are defined by the two polyhedra

$$P^1 = \{(x^1, y^1) : x^1 \geq \mathbf{0},\ y^1 \geq \mathbf{0},\ x_1^1 \leq 1,\ y_1^1 + y_2^1 = 2,\ 2x_1^1 \leq y_1^1\},$$
$$P^2 = \{(x^2, y^2) : x^2 \geq \mathbf{0},\ y^2 \geq \mathbf{0},\ x_2^2 \leq 1,\ y_2^2 + y_1^2 = 2,\ 2x_2^2 \leq y_2^2\}.$$

Let $r_1^1 = r_2^1 = r_1^2 = r_2^2 = \frac{1}{2}$. The costs of the x variables are 0, while the costs of the y variables are defined by $c(y_1^1) = 1$, $c(y_2^1) = -1$, $c(y_1^2) = -1$, $c(y_2^2) = 1$.

It is easy to see that the only feasible solution is $x_1^1 = x_2^1 = x_1^2 = x_2^2 = 1$, $y_1^1 = 2$, $y_2^1 = 0$, $y_1^2 = 0$, $y_2^2 = 2$. If $p_1 > p_2 - 4$, then this is not an equilibrium because player 1 is better off decreasing x_1^1 and y_1^1 to 0 and increasing x_2^1 and y_2^1 to 2.

On the other hand, if $p_2 > p_1 - 4$, then player 2 profits by decreasing x_2^2 and y_2^2 to 0 and increasing x_1^2 and y_1^2 to 2. Therefore there are no prices for which an equilibrium exists.

We now prove Theorem 1.2, which shows that examples like the above are only possible if there are players who are both providers and customers of the same service.

Proof of Theorem 1.2. As in the proof of Theorem 1.1, let H be the subspace defined by the weak feasibility equations. Let furthermore

$$P_1 = P_\times \cap \{(x, y, z) : x_s^i = r_s^i \sum_{j : s \in S^j} x_s^j \quad \text{for every } i \text{ and } s \in S^i.\}$$

Notice that $P_{\text{feas}} = P_1 \cap H$. Let F_{opt} be the optimal face in P_{feas} minimizing the cost c. Let F_1 be the minimal face of P_1 that contains F_{opt}.

We apply Lemma 2.1 with P_1, $\Pi = H$, $P_2 = P_{\text{feas}}$, F_1 and F_2. By consequence, there is a vector $h \in (\text{lin}H)^\perp$ for which $c + h$ is a member of relint(opt.cone(F_1, P_1)). Since h is in $(\text{lin}H)^\perp$, there are prices p such that

- The component of h corresponding to x^i_s is $-p_s$,
- the component of h corresponding to y^i_s is p_s,
- the components of h corresponding to z are 0.

Since $c + h = c_p$, any socially optimal solution $(x, y, z) \in F_{\text{opt}} \subseteq F_1$ is optimal for objective function c_p in the polyhedron P_1. Now we are done by the observation that since $S^i \cap T^i = \emptyset$ for each player i, a strategy change by player i preserving his personal service equations actually preserves all values of x^i, so the obtained vector remains in P_1. This means that each socially optimal solution is in equilibrium because it is optimal for c_p in P_1. □

3 Structure and existence of equilibria

Form now on, we assume that the service prices are fixed. We have seen that the set of equilibrium solutions can be empty even if P_{feas} is non-empty. Moreover, even if non-empty, the set of equilibrium solutions is not necessarily convex or connected. This is in contrast to socially optimal solutions that form a face of P_{feas}. However, we show that if equilibria exist, then they form the union of some faces of P_{feas}.

For $i \in [n]$ and $y \in \mathbb{R}^{T^1 \times \cdots \times T^n}$ let

$$\Phi^i(y) = \{(\overline{x}^i, \overline{y}^i, \overline{z}^i) : \overline{x}^i, \overline{y}^i \text{ satisfy}$$

$$\overline{x}^i_s = r^i_s \left(\overline{y}^i_s + \sum_{j \neq i : s \in T^j} y^j_s \right) \quad \text{if } s \in S^i \cap T^i \text{ and} \tag{4}$$

$$\overline{x}^i_s = r^i_s \sum_{j : s \in T^j} y^j_s \quad \text{if } s \in S^i \setminus T^i\}, \tag{5}$$

in other words, $\Phi^i(y)$ is an affine subspace consisting of vectors $(\overline{x}^i, \overline{y}^i, \overline{z}^i)$ that satisfy the personal service equations of player i with respect to $(y^j)_{j \in [n] \setminus \{i\}}$. Let furthermore $\Phi(y) = \Phi^1(y) \times \cdots \times \Phi^n(y)$. Observe that $\text{lin}\Phi(y)$ is the same for every y.

By definition, a vector $(x, y, z) \in P_\times$ is feasible if and only if it is in $\Phi(y)$. Furthermore, $(x, y, z) \in P_{\text{feas}}$ is an equilibrium if and only if it is optimal in $P_\times \cap \Phi(y)$ for the objective function c_p.

Lemma 3.1. *The set of equilibrium solutions is the (perhaps empty) union of some faces of P_{feas}. As a consequence, the set of equilibria is either empty, or there is an equilibrium that is a vertex of P_{feas} and hence its bit-complexity is polynomial in the input size.*

Proof. Let $(x, y, z) \in P_\times \cap \Phi(y)$ be an equilibrium, i.e. optimal in $P_\times \cap \Phi(y)$ for the objective function c_p, and let F_2 be the minimal face of $P_\times \cap \Phi(y)$ containing it. If we apply Lemma 2.1 with $P_1 = P_\times$, $\Pi = \Phi(y)$, and F_2, we obtain that $c_p \in \text{opt.cone}(F_2, P_\times \cap \Phi(y)) = \text{opt.cone}(F_1, P_\times) + (\text{lin}\Phi(y))^\perp$, where F_1 is the minimal face of $P_\times$ containing (x, y, z).

Suppose that (x', y', z') is on the minimal face of P_{feas} containing (x, y, z). Then on the one hand $(x', y', z') \in F_1$, and on the other hand $(x', y', z') \in P_\times \cap \Phi(y')$. But $(\text{lin}\Phi(y'))^\perp = (\text{lin}\Phi(y))^\perp$, and therefore $c_p \in \text{opt.cone}(F_1, P_\times) + (\text{lin}\Phi(y'))^\perp \subseteq \text{opt.cone}((x', y', z'), P_\times \cap \Phi(y'))$, so (x', y', z') is also an equilibrium. □

In order to show the existence of equilibria in safe instances (Theorem 1.3), we resort to the following fundamental fixed-point theorem of Kakutani.

Theorem 3.2 (Kakutani [12])**.** *Let C be a compact convex set in $\mathbb{R}^d$ and let $\varphi : C \to \mathcal{P}(C)$ be a set-valued function with the following properties:*

- *$\varphi(x)$ is a nonempty convex subset of C for every $x \in C$,*
- *the graph of φ is closed.*

Then there exists a fixed point of φ, that is, an element $x \in C$ for which $x \in \varphi(x)$.

Let $C = \{y : \exists x, z \text{ s.t. } (x, y, z) \in P_\times\}$. Note that C is compact and convex, and a safe instance means that $P_\times \cap \Phi(y)$ is non-empty for any $y \in C$.

Proof of Theorem 1.3. For $y \in C$, let

$$\varphi(y) = \{\overline{y} \in C : \exists \overline{x}, \overline{z} \text{ s.t. } (\overline{x}, \overline{y}, \overline{z}) \text{ is optimal for } c_p \text{ in } P_\times \cap \Phi(y)\}.$$

Since the instance is safe, $\varphi(y)$ is nonempty for every $y \in C$. The set $\varphi(y)$ is the projection of a face of $P_\times \cap \Phi(y)$, so it is convex. It remains to show that the graph of φ is closed. If a_k is a convergent sequence in C with $\lim_{k\to\infty} a_k = a$, then $\lim_{k\to\infty} \Phi(a_k) = \Phi(a)$. Suppose that $b_k \in \varphi(a_k)$ and $\lim_{k\to\infty} b_k = b$. Since the polyhedra are bounded, there exist convergent sequences $\lim_{k\to\infty} \overline{x}_k = \overline{x}$ and $\lim_{k\to\infty} \overline{z}_k = \overline{z}$ such that $(\overline{x}_k, b_k, \overline{z}_k)$ is optimal for c_p in $P_\times \cap \Phi(a_k)$. The convergence implies that $(\overline{x}, b, \overline{z})$ is optimal for c_p in $P_\times \cap \Phi(a)$, and thus $b \in \varphi(a)$.

By Theorem 3.2 there exists a fixed point, that is, a vector y with $y \in \varphi(y)$. By the definition of φ, there exist x, z such that (x, y, z) is optimal in $P_\times \cap \Phi(y)$ for the objective function c_p. This means that (x, y, z) is an equilibrium. □

4 PPAD-completeness

In this section we show that the problem of finding an equilibrium in a safe instance of the Multiplayer Multicommodity Flow Problem (described in Section 1.5.2) is PPAD-complete. Membership in PPAD follows from the fact that the computational version of Kakutani's fixed point theorem is in PPAD, as shown by Papadimitriou [16]. The proof in [16] is a reduction to Sperner's Lemma (via Brouwer's fixed point theorem), and it only works if φ has the following property: given a sufficiently small simplex that is known to contain a fixed point of φ, we can compute a fixed point in polynomial time. It can be checked (using a similar argument as the proof of Corollary 5.4) that the function φ defined in the proof of Theorem 1.3 has this property.

PPAD-hardness is shown by a reduction of two-player Nash equilibrium to our problem. To be more precise, we reduce approximate 2-Nash, so we use the following fundamental result of Chen, Deng, and Teng [6].

Theorem 4.1 ([6]). *For any $\alpha > 0$, the problem of computing an $m^{-\alpha}$-approximate Nash equilibrium of a two-player game is PPAD-complete, where m is the number of strategies.*

We need a couple of remarks in order to use this theorem. First, it is well known that the problem of finding two-player Nash equilibria can be reduced to finding symmetric Nash equilibria in symmetric games, so we will assume that the game is symmetric, with utility matrix $A \in \mathbb{R}^{m\times m}$. Second, the above theorem is valid if the matrix A is normalized in the sense that its entries are bounded. For our purposes it is convenient to say that a symmetric two-player game is *normalized* if the elements of the matrix A are rationals in the interval $[1, 2]$. Third, approximate equilibria can be defined in several ways; we use a definition in [6]: x^* is an *ε-well supported approximate symmetric Nash equilibrium* if $x^*_k > 0$ implies that $\sum_{j=1}^m a_{kj}x^*_j > \max_{i\in[m]} \sum_{j=1}^m a_{ij}x^*_j - \varepsilon$. Finally, it is convenient to set $\alpha = 1$. To sum up, we use the following form of the theorem.

Corollary 4.2 ([6]). *The problem of computing a $\frac{1}{m}$-well supported approximate symmetric Nash equilibrium of a symmetric normalized two-player game is PPAD-complete.*

The above problem will be called $\frac{1}{m}$-APPROXIMATE 2-NASH in this paper.

Proof of Theorem 1.4. We have to reduce $\frac{1}{m}$-APPROXIMATE 2-NASH to finding an equilibrium in a safe instance of MMF. Given a symmetric normalized game defined by a matrix $A \in [1, 2]^{m\times m}$, we construct a safe instance of MMF featuring 4 agents. In order to make the construction more understandable, the agents are named Decision Maker, Combiner, Maximizer and Inverter. The high level view of the role of the agents is that the Decision Maker decides the values of x satisfying $x \geq 0$ and $\sum_{j=1}^m x_j = 1$, while the other agents are "gadgets" that compute $m(M - \sum_{j=1}^n a_{kj}x_j)$ for every k, where $M = \max_{i\in[m]} \sum_{j=1}^m a_{ij}x_j$. These values then appear in the network of the Decision Maker as contractual demands, in such a way that in an equilibrium solution x is guaranteed to be a $\frac{1}{m}$-well supported approximate Nash equilibrium. The prices will be 0 on all contractual arcs. The details of the construction, illustrated on Figure 1, are the following.

The Decision Maker has one normal demand: a unit commodity from s to t. His network contains m internally disjoint st-paths: $s, u_{0j}, u_{1j}, \ldots, u_{mj}, v_j, w_j, t$ ($j = 1, \ldots, m$), each arc having cost 0 and capacity 1. The arcs $u_{i-1,j}u_{ij}$ ($i, j \in [m]$) are contractual arcs to the Combiner, and the multiplier of $u_{i-1,j}u_{ij}$ is a_{ij}.

The network of the Decision Maker also contains nodes s_j, t_j ($j = 1, \ldots, m$), arcs s_jt_j with cost 1 and capacity $2m$, and arcs s_jv_j, w_jt_j with cost 0 and capacity 1.

For a feasible solution x, we will denote by x_j the flow value on the arc su_{0j} in the network of the Decision Maker.

The network of the Combiner consists of paths $u_{i-1,j}, v_i, w_i, u_{ij}$ for $i, j \in [m]$, all arcs having cost 0 and capacity 2. The arcs v_iw_i are contractual arcs to the Maximizer with multiplier 1. Note that the Combiner has a unique way to route his contractual demands, and in a feasible solution the flow value on arc v_iw_i is $\sum_{j=1}^m a_{ij}x_j$, so this is the contractual demand appearing at the Maximizer.

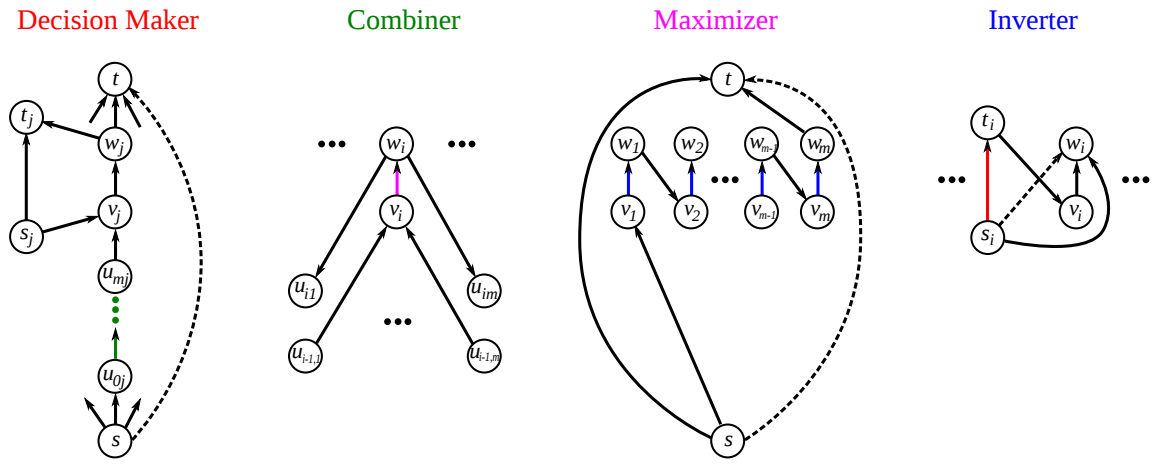

Figure 1: Illustration of the networks of various agents. Dashed arrows are normal demands, coloured arrows are contractual arcs.

The network of the Maximizer consists of a path $s, v_1, w_1, v_2, w_2, \ldots, v_m, w_m, t$, all arcs having cost 0 and capacity 2. The arcs $v_i w_i$ are contractual arcs to the Inverter, with multiplier 1. There is also an arc st with cost 1 and capacity 2. The Maximizer has a normal demand of 2 from s to t. The routing that the Maximizer has to choose in an equilibrium solution is the following: he must route his contractual demands on the edges $v_i w_i$; he must route as much of his normal demand on the long $s-t$ path as possible, and route the rest on the arc st. Let M denote $\max_{i\in[m]} \sum_{j=1}^{m} a_{ij}x_j$ (note that $M \leq 2$). Then the portion of the normal demand routed on the long path is $2-M$ units, so the contractual demand appearing at the Inverter between v_i and w_i is $2 - M + \sum_{j=1}^{m} a_{ij}x_j$ units.

Inverter has a path s_i, t_i, v_i, w_i for each $i \in [m]$ with arcs of cost 0 and capacity 2. The arcs $s_i t_i$ are contractual arcs to the Decision Maker, with multiplier m. In addition he has arcs $s_i w_i$ ($i \in [m]$) with cost 1 and capacity 2, and normal demands of size 2 from s_i to w_i. In an equilibrium solution, he routes his contractual demands on the arcs $v_i w_i$, and routes as much of his normal $s_i - w_i$ demand on the path of cost zero as possible, the bottleneck being the arc $v_i w_i$. Thus $M - \sum_{j=1}^{m} a_{ij}x_j$ is routed on the path of cost zero, and $2 - M + \sum_{j=1}^{m} a_{ij}x_j$ on the arc $s_i w_i$. This means that the contractual demand appearing at the Decision Maker between s_i and t_i is $m(M - \sum_{j=1}^{m} a_{ij}x_j)$.

It is easy to check that this construction results in a safe instance: we have added arcs of non-zero cost with enough capacity to carry the maximum demands.

Let us prove that an equilibrium x^* corresponds to a $\frac{1}{m}$-well supported approximate Nash equilibrium. Clearly, $x^* \geq 0$ and $\sum_{j=1}^{m} x_j^* = 1$, so we have to prove that $x_i^* > 0$ implies that $\sum_{j=1}^{m} a_{ij}x_j^* > M - \frac{1}{m}$, where $M = \max_{i\in[m]} \sum_{j=1}^{m} a_{ij}x_j^*$. In the discussion above we have showed that the Decision Maker has a contractual demand of $m(M - \sum_{j=1}^{m} a_{ij}x_j^*)$ from s_i to t_i. We know that there is an index k such that $\sum_{j=1}^{m} a_{kj}x_j^* = M$, so the contractual demand from s_k to t_k is 0. This means that the following is a feasible multicommodity flow in the network of the Decision Maker:

- route the normal demand on the path $s, u_{0k}, u_{1k}, \ldots, u_{mk}, v_k, w_k, t$,
- for $i \neq k$, let $\delta_i = m(M - \sum_{j=1}^{m} a_{ij} x_j^*)$. Route $\min\{1, \delta_i\}$ units of the $s_i - t_i$ contractual demand on the path s_i, v_i, w_i, t_i, and the rest on the arc $s_i t_i$.

It is easy to check that this is a minimum cost multicommodity flow for the given demands. Since x^* is an equilibrium solution, it must have the same cost as this one in the network of the Decision Maker. This is only possible if $\delta_i \leq 1 - x_i^*$ whenever $x_i^* > 0$. Thus $M - \sum_{j=1}^{m} a_{ij} x_j^* < \frac{1}{m}$ whenever $x_i^* > 0$, which means that x^* is a $\frac{1}{m}$-well supported approximate Nash equilibrium. □

5 Polynomially solvable cases

In the PPAD-completeness proof in Section 4, we reduced approximate 2-Nash to 4-agent MMCF instances where the provider-customer pairs constituted a directed 4-cycle on the set of agents. One may ask if this leaves room for an interesting class of service configurations where an equilibrium can be found in polynomial time.

A natural candidate is the class of problems where the customer-provider relationships form an acyclic digraph, and indeed it is not hard to show that an equilibrium can be computed efficiently in that case. The main result of this section is an efficient algorithm for a broader class of problems. Let us consider an auxiliary directed graph $D^* = ([n], A^*)$ on the set of players, in which there are $|S^i \cap T^j|$ parallel ij arcs if $i \neq j$ (so D^* does not have loops).

With this definition, the directed graph corresponding to the hard instances constructed in the proof of Theorem 1.4 is a 4-cycle *with many parallel arcs*. In contrast to this, Theorem 1.5 states that if the strongly connected components of D^* are *simple* directed cycles, then there is a polynomial-time algorithm to find a fixed-ratio equilibrium (here a pair of oppositely directed arcs is also considered as a simple cycle).

The main tool of the proof is an algorithm for finding approximate fixed points of a special kind of mapping, as described in Theorem 1.6. Let $\varphi_i : [0,1] \to \mathcal{P}([0,1])$ $(i \in [m])$ be mappings such that $\varphi_i(t)$ is a non-empty interval for every $t \in [0,1]$, and the graph of φ_i is closed. We are interested in finding a fixed point of the mapping $(x_1, x_2, \ldots, x_m) \mapsto (\varphi_m(x_m), \varphi_1(x_1), \ldots, \varphi_{m-1}(x_{m-1}))$. In other words, we are looking for a vector $x = (x_1, \ldots, x_m)$ for which $x_{i+1} \in \varphi_i(x_i)$ for every $i \in [m]$. Here and later in this section, unless otherwise stated, we consider the indices modulo m, i.e. $x_{m+1} = x_1$ and $\varphi_m = \varphi_0$.

Definition. A vector $x = (x_1, \ldots, x_m)$ that satisfies $x_{i+1} \in \varphi_i(x_i)$ for every $i \in [m]$ is called a *cyclically fixed vector*.

In order to have a meaningful definition of running time, we use the following oracle model: there is an evaluation oracle which, given $t \in [0,1]$ and $i \in [m]$, returns some $z \in \varphi_i(t)$ in one step. We also consider basic arithmetic operations as one step. Of course it is hopeless to compute a cyclically fixed vector exactly in this oracle model, but, as stated in Theorem 1.6, we can approximate it in polynomial number of steps.

Proof of Theorem 1.6. The algorithm itself is quite simple and its time complexity is straightforward, the more involved part being the proof of its correctness. During the algorithm we always follow the rule that if at some point the oracle returns $z \in \varphi_i(t)$, then the triplet (i, t, z) is stored, and we use z in all subsequent evaluations of $\varphi_i(t)$.

The intervals are determined successively in reverse order. To determine I_m, initially let $a_m = 0$ and $b_m = 1$. Let $\psi_m = \varphi_{m-1} \circ \varphi_{m-2} \circ \cdots \circ \varphi_1 \circ \varphi_m$. The function ψ_m is an interval-valued function with a closed graph, since it is the composition of such functions.

Let $t = (a_m + b_m)/2$. We can compute a value $z \in \psi_m(t)$ by m successive oracle calls. If $z \leq t$, then let $b_m = t$. If $z \geq t$, then let $a_m = t$. These steps are repeated until $b_m - a_m \leq \varepsilon$. Let $I_m = [a_m, b_m]$.

Suppose that we have already determined $I_{i+1} = [a_{i+1}, b_{i+1}]$. We modify the function φ_i as follows.

$$\varphi_i'(t) = \begin{cases} \varphi_i(t) \cap I_{i+1} & \text{if } \varphi_i(t) \cap I_{i+1} \neq \emptyset, \\ a_{i+1} & \text{if } z < a_{i+1} \text{ for every } z \in \varphi_i(t), \\ b_{i+1} & \text{if } z > b_{i+1} \text{ for every } z \in \varphi_i(t). \end{cases}$$

Note that this modification can be implemented simply by modifying the value returned by the oracle after each oracle call for φ_i: if the returned value is smaller than a_{i+1}, then we change it to a_{i+1}, and if it is greater than b_{i+1}, then we change it to b_{i+1}.

In order to compute I_i, initially let $a_i = 0$ and $b_i = 1$, and let $\psi_i = \varphi_{i-1} \circ \varphi_{i-2} \circ \cdots \circ \varphi_0 \circ \varphi'_{m-1} \circ \cdots \circ \varphi'_{i+1} \circ \varphi'_i$.

In a general step, let $t = (a_i + b_i)/2$, and let us compute a value $z \in \psi_i(t)$ by m oracle calls.

Definition. The sequence of the returned values of these m oracle calls is called the *itinerary* of the pair (i, t).

Let $b_i = t$ if $z \leq t$, and let $a_i = t$ if $z \geq t$. The above steps are repeated until $b_i - a_i \leq \varepsilon$, in which case we fix I_i to be the interval $[a_i, b_i]$. We can observe the following.

Observation 5.1. *The m-th step of the itinerary of (i, a_i) is greater than a_i, and the m-th step of the itinerary of (i, b_i) is smaller than b_i.* □

The algorithm described above computes the interval I_i in $O(m \log(\frac{1}{\varepsilon}))$ steps for a given i, so the total number of steps is $O(m^2 \log(\frac{1}{\varepsilon}))$. It remains to show that the intervals contain a cyclically fixed vector.

Let us define mappings $\varphi_i^* : I_i \to \mathcal{P}(I_{i+1})$ for each $i \in [m]$ by $\varphi_i^*(t) = \varphi_i(t) \cap I_{i+1}$. Note that the image for a value t is either a closed interval or the empty set. We also define for any positive integer k (this time not taken modulo m) the mapping $\psi_k^* = \varphi_k^* \circ \varphi_{k-1}^* \circ \cdots \circ \varphi_1^*$. We can observe that $\psi_{m+k}^*(I_1) \subseteq \psi_k^*(I_1)$ for any k. We define ψ_0^* to be the identity function.

Claim 5.2. *The set $\psi_k^*(I_1)$ is non-empty for every k.*

Proof. Indirectly, suppose that $im + k$ (where $1 \leq k \leq m$) is the smallest integer for which $\psi^*_{im+k}(I_1) = \emptyset$. We may assume w.l.o.g. that $\varphi_k \circ \psi^*_{im+k-1}(t) > b_{k+1}$ for every $t \in I_1$. This implies that $b_{k+1} \in \psi^*_{(i-1)m+k}(I_1)$ (provided that $(i-1)m + k \geq 0$), because $\varphi_k \circ \psi^*_{(i-1)m+k-1}(t)$ is an interval that contains b_{k+1}.

Let us examine the itinerary of $(k+1, b_{k+1})$. We claim that for any $j < m$ the j-th step of the itinerary is in $\psi^*_{(i-1)m+k+j}(I_1)$, provided that $(i-1)m+k+j \geq 0$. We show this by induction on j, first for $j < m - k$. In this case $\psi^*_{(i-1)m+k+j}(I_1)$ is non-empty and so the set $\varphi'_{k+j} \circ \psi^*_{(i-1)m+k+j-1}(I_1)$, which contains the j-th step of the itinerary, is equal to $\psi^*_{(i-1)m+k+j}(I_1)$.

Next, we show that for $m-k \leq j < m$ it also holds that the j-th step of the itinerary is in $\psi^*_{(i-1)m+k+j}(I_1)$. The difficulty here is that the itinerary proceeds according to the mapping φ_{k+j}, which, as opposed to φ'_{k+j}, may have values outside of I_{k+j+1}. Suppose that the j-th step is the first to be outside of I_{k+j+1}; we can assume w.l.o.g. that it is greater than b_{k+j+1}. This means that b_{k+j+1} is in $\psi^*_{(i-1)m+k+j}(I_1)$ (because $\psi^*_{(i-1)m+k+j}(I_1)$ is non-empty), so the itinerary of $(k+j+1, b_{k+j+1})$ leads to b_{k+1}, after which it takes the same steps as the itinerary of $(k+1, b_{k+1})$ — here we use that the evaluation oracle cannot return different values for the same input. Thus the m-th step of the itinerary of $(k+j+1, b_{k+j+1})$ is greater than b_{k+j+1}, contradicting Observation 5.1.

We can conclude that the $(m-1)$-th step of the itinerary of $(k+1, b_{k+1})$ is in $\psi^*_{im+k-1}(I_1)$. Therefore the m-th step of the itinerary is greater than b_{k+1} by our assumption, again contradicting Observation 5.1. □

Since the set $\psi^*_{im}(I_1)$ is a non-empty closed interval for every i, and $\psi^*_{(i+1)m}(I_1) \subseteq \psi^*_{im}(I_1)$, we have that

$$R = \cap_{i=1}^{\infty} \psi^*_{im}(I_1) \text{ is a non-empty interval } [a, b].$$

Claim 5.3. *R contains a fixed point of ψ^*_m.*

Proof. It is easy to see that $\psi^*_m(R) = R$. For $k \in [m]$, let

$$R_k = \{t \in R : \psi^*_k(t) = \emptyset, \text{ but } \psi^*_j(t) \neq \emptyset \text{ for } j < k\},$$

and let $R^* = R \setminus \cup_{j=1}^{m} R_j$. Our aim is to find an interval $I^* = [a^*, b^*] \subseteq R^*$ such that $\psi^*_m(I^*) = R$. To do this, we show that for every $k \in [m]$, there is an interval $I^*_k = [a^*_k, b^*_k] \subseteq R \setminus \cup_{j=1}^{k} R_j$ such that $\psi^*_m(I^*_k) = R$. We can start with $I^*_0 = R$; suppose that we have already determined I^*_{k-1}. If $t \in R_k \cap I^*_{k-1}$, then either $\psi^*_k([a^*_{k-1}, t]) \subseteq \psi^*_k([t, b^*_{k-1}])$ or vice versa because both are sub-intervals of I_{k+1} containing the same endpoint of I_{k+1}. Consequently, either $\psi^*_m([a^*_{k-1}, t]) = R$ or $\psi^*_m([t, b^*_{k-1}]) = R$. We may assume w.l.o.g. that there is at least one t for which $\psi^*_m([t, b^*_{k-1}]) = R$. Let $a^*_k = \sup\{t \in I^*_{k-1} : \psi^*_m([t, b^*_{k-1}]) = R\}$.

Because of the closed graph property, $a^*_k \notin R_k$ and $\psi^*_m([a^*_k, b^*_{k-1}]) = R$ holds. If $R_k \cap [a^*_k, b^*_{k-1}] = \emptyset$, then we can set $b^*_k = b^*_{k-1}$. Otherwise by the choice of a^*_k we have that $\psi^*_m([t, b^*_{k-1}]) \neq R$ for every $t \in R_k \cap ([a^*_k, b^*_{k-1}]$. On the other hand, for such a t both $\psi^*_k([a^*_k, t])$ and $\psi^*_k([t, b^*_{k-1}])$ contain the same endpoint of I_{k+1}, so one of them

contains the other. Since $\psi_m^*([a_k^*, t]) \cup \psi_m^*([t, b_{k-1}^*]) = R$, it follows that $\psi_m^*([a_k^*, t]) = R$. Let $b_k^* = \inf\{R_k \cap [a_k^*, b_{k-1}^*]\}$; then $\psi_m^*([a_k^*, b_k^*]) = R$, and $[a_k^*, b_k^*] \cap R_k = \emptyset$, as required.

We obtained an interval $I^* = [a^*, b^*] \subseteq R^*$ such that $\psi_m^*(I^*) = R$. Now it follows from the closed graph property of ψ_m^* on I^* that it has a fixed point in I^*. □

By definition, a fixed point of ψ_m^* implies the existence of a cyclically fixed vector x with $x_i \in I_i$ ($i \in [m]$). This concludes the proof of the theorem. □

We now show that if the graph of each mapping φ_i is a 2-dimensional polyhedral complex that can be described by inequalities of bit size M, then an exact cyclically fixed vector can be computed in a number of steps polynomial in M.

Corollary 5.4. *Let $\varphi_i : [0,1] \to \mathcal{P}([0,1])$ ($i \in [m]$) be given as above. Then in $O(m^2M^2)$ steps we compute a cyclically fixed vector.*

Proof. Let $\varepsilon = \exp(-M^2)$. Since the graph of the function φ_i is a 2-dimensional polyhedral complex of bit size M, the choice of ε guarantees that all vertices of this polyhedral complex in the interior of $I_i \times [0,1]$ (if the exist at all) have the same first coordinate t, which can be computed in polynomial time. We modify the algorithm in the proof of Theorem 1.6 the following way: after achieving $b_i - a_i \leq \varepsilon$, we make an additional step with the above t in place of $t = (a_i + b_i)/2$. This way we obtain intervals I_i such that the graph of φ_i has no vertices in the interior of $I_i \times [0,1]$ ($i \in [m]$). This means that the mappings φ_i are "linear" on these intervals in the sense that their graph is of the form $\alpha_i x_i \leq x_{i+1} \leq \beta_i x_i$, and an equilibrium can be found by solving an LP. □

We are now ready to prove that an equilibrium can be found in polynomial time in a safe instance if the strongly connected components of D^* are simple directed cycles.

Proof of Theorem 1.5. Let us consider a safe instance where every strong component of D^* is a simple directed cycle. Let $C_1, \ldots, C_q$ be the family of strong components in reverse topological order, and for $1 \leq k \leq q$ let V_k denote the set of players in C_k. In the kth phase, we will compute a solution $(x^i, y^i, z^i) \in P^i$ of each player $i \in V_k$, in such a way that the personal service equations of players in V_k are satisfied. Because of the reverse topological order, these equations are not modified in later phases of the algorithm, so the solution obtained at the end is feasible.

Suppose that we have already determined the solutions of players up to V_{k-1}. Let $i_1, \ldots, i_m$ denote the players in V_k, in reverse order of the cycle C_k. Let s_j be the unique element of $S^{i_{j+1}} \cap T^{i_j}$ and let

$$[\ell_j, u_j] = \{t \in \mathbb{R} : \ P^{i_j} \cap \{(x^{i_j}, y^{i_j}, z^{i_j}) : y^{i_j}_{s_j} = t\} \neq \emptyset\}.$$

The function φ_j assigns to each $t \in [\ell_j, u_j]$ a nonempty subinterval of $[\ell_{j+1}, u_{j+1}]$ the following way. If we fix $y^{i_j}_{s_j}$ at t, then by the definition of safeness there exists a vector $(\overline{x}^{i_{j+1}}, \overline{y}^{i_{j+1}}, \overline{z}^{i_{j+1}}) \in P^{i_{j+1}}$ such that, together with the values already fixed, it satisfies all personal service equations of player i. Let us take the set of such vectors that have minimum c_p-cost, and let $\varphi_j(t)$ consist of the possible $\overline{y}^{i_{j+1}}_{s_{j+1}}$ values in these vectors.

This is a non-empty subinterval of $[\ell_{j+1}, u_{j+1}]$. A value of $\varphi_j(t)$ can be computed using linear programming, and the graph of φ_j is a 2-dimensional polyhedral complex describable by inequalities of bit size $O(M)$, where M is the size of the input.

We can use Corollary 5.4 (by appropriately scaling the functions φ_j) to find a cyclically fixed vector of $\varphi_1, \ldots, \varphi_m$. This means that we have found vectors $(x^i, y^i, z^i) \in P^i$ $(i \in V_k)$ such that the personal service equations of all players in V_k are satisfied, and for each $i \in V_k$, if we fix the solutions of other agents, then (x^i, y^i, z^i) has minimum c_p-cost among the vectors in P^i that satisfy the personal service equations of i.

The end result is a feasible solution because all personal service equations are satisfied, and it is an equilibrium by the above argument. □

6 Open questions

A notable open question is the computational complexity of finding an equilibrium when the number of services is constant. The answer is unknown even in the following simple setting: there are two players and only three services, two of them offered by player 1 to player 2, and one by player 2 to player 1. Although one might expect this case to be easily solvable, an algorithm similar to the one described in the proof of Theorem 1.6 does not seem to work.

From the mechanism design point of view, it would be interesting to find bargaining mechanisms that lead to perfect prices, or at least to some approximate version of perfectness.

Acknowledgements

We would like to thank the authors of [5] for fruitful discussions on the topic. Research was supported by grants CK80124 and K109240 from the National Development Agency of Hungary.

References

[1] T.G. Abbott, D.M. Kane, P. Valiant, *On the complexity of two-player win- lose games*, in 46th Symposium on Foundations of Computer Science IEEE Computer Society (2005), 113–122.

[2] H. Ackermann, H. Röglin, B. Vöcking, *On the impact of combinatorial structure on congestion games*, Journal of the ACM, 55:6 (2008), Article 25.

[3] R. Agarwal, Ö. Ergun, *Mechanism design for a multicommodity flow game in service network alliances*, Operations Research Letters 36 (2008), 520–524.

[4] K.J. Arrow, G. Debreu, *Existence of an equilibrium for a competitive economy*, Econometrica 22 (1954), 265–290.

[5] A. Bernáth, T. Király, E.R. Kovács, G. Mádi-Nagy, Gy. Pap, J. Pap, J. Szabó, L. Végh, *Algorithms for multiplayer multicommodity flow problems*, Central European Journal of Operations Research 21 (2013), 699–712.

[6] X. Chen, X. Deng, S-H. Teng, *Settling the complexity of computing two-player Nash equilibria*, Journal of the ACM, Volume 56 Issue 3 (2009), Article No. 14

[7] J.J.M. Derks, S.H. Tijs, *Stable outcomes for multicommodity flow games*, Methods of Operations Research, 50 (1985), 493–504.

[8] E. Elkind, L.A. Goldberg, P.W. Goldberg, *Nash Equilibria in Graphical Games on Trees Revisited*, in 7th ACM Conference on Electronic Commerce (2006), 100–109.

[9] F. Facchinei, C. Kanzow, *Generalized Nash equilibrium problems*, 4OR Volume 5, Issue 3 (2007), 173–210.

[10] P.W. Goldberg, *A Survey of PPAD-Completeness for Computing Nash Equilibria*, arXiv:1103.2709

[11] L. Guyi, Ö. Ergun, *Dual Payoffs, Core and a Collaboration Mechanism Based on Capacity Exchange Prices in Multicommodity Flow Games*, Proc. 4th International Workshop on Internet and Network Economics (2008), 61–69.

[12] S. Kakutani, *A generalization of Brouwer's fixed point theorem*, Duke Math. J. 8 (1941), 457–459.

[13] E. Kalai, E. Zemel, *Generalized network problems yielding totally balanced games*, Operations Research 30 (5) (1981) 998–1008.

[14] E. Kalai, E. Zemel, *Totally balanced games and games of flow*, Mathematics of Operations Research 7 (3) (1982) 476–478.

[15] E. Markakis, A. Saberi, *On the core of the multicommodity flow game*, Proceedings of the 4th ACM conference on Electronic commerce (2003), 93–97.

[16] C.H. Papadimitriou, *On the complexity of the parity argument and other inefficient proofs of existence*, Journal of Computer and System Sciences Volume 48, Issue 3 (1994), 498–532.

[17] T. Roughgarden, *Computing Equilibria: A Computational Complexity Perspective*, Economic Theory 42:1 (2010), 193–236.

[18] V.V. Vazirani, M. Yannakakis, *Market Equilibrium under Separable, Piecewise-Linear, Concave Utilities*, Journal of the ACM (JACM) 58:3 (2011), Article 10.

[19] L.A. Végh, *Strongly polynomial algorithm for a class of minimum-cost flow problems with separable convex objectives*, Proceedings of the 44th symposium on Theory of Computing (2012), 27–40.