Finding equilibria in linear service-providing games

Tamás Király and Júlia Pap

October 2016
Finding equilibria in linear service-providing games

Tamás Király* and Júlia Pap

Abstract

A fundamental problem of algorithmic game theory is to determine the hardness of computing equilibria in various classes of games. In this paper, we examine a class of LP-based generalized Nash equilibrium problems where the interaction of players is limited to providing services to each other at fixed prices. This limited interaction between players enables the study of computational complexity as a function of the structure of provider-customer relationships.

We show that the problem of computing an equilibrium is PPAD-complete in general, but it is in P if every strong component of the digraph describing the provider-customer relationships is a simple directed cycle. The proof is based on a new result on approximating fixed points in a special case of Kakutani’s fixed point theorem.

We also give sufficient conditions for the existence of service prices under which any socially optimal solution is in equilibrium. If such prices exist, then an equilibrium can be computed in polynomial time. This generalizes an earlier result of Agarwal and Ergun on service networks [Agarwal, Ergun, Mechanism design for a multicommodity flow game in service network alliances, 2008].

1 Introduction

The complexity of computing equilibria is one of the fundamental problems of algorithmic game theory. A sequence of results, culminating in the breakthrough paper of Chen, Deng and Teng [6], showed that Nash-equilibrium is PPAD-hard to approximate, even for two players, sparse payoff matrices [6], and 0-1 payoff [1]. However, there are important classes of problems, like some market equilibrium problems [20, 21] and certain types of congestion games [2], where an equilibrium can be computed efficiently. For detailed surveys on these topics, see [10, 19].

One approach to capture the structural properties of interactions in games is to consider graphical games, where each player has only two possible strategies, and the payoffs of players depend only on the strategies of their neighbours (and themselves) 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 the problem is PPAD-complete for maximum degree 3 and constant pathwidth.

*MTA-ELTE Egerváry Research Group, Department of Operations Research, Eötvös University, Budapest. Emails: {tkiraly,papjuli}@cs.elte.hu

October 2016
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 a bounded polyhedron, and interactions between players are restricted to services being provided to each other at fixed prices, which can be expressed using linear equations. This structure guarantees that if equilibria exist, then they form the union of faces of a polyhedron, as in the case of bimatrix games. Furthermore, best-response strategies can be computed in polynomial time using linear programming.

Under some natural conditions, equilibria are guaranteed to exist, and we show that finding one is PPAD-complete. However, we manage to give a polynomial-time algorithm if the structure of services is sufficiently simple.

For the general case, when equilibria may not exist, we study whether service prices can be changed in such a way that socially optimal solutions (which can be computed in polynomial time) become equilibria.

Some of these results can be seen as a generalization of 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 utilities, 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 was 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 model of Agarwal and Ergun, each agent controls a subnetwork, but instead of the exchange of arc capacities, there are bilateral agreements specifying source-destination pairs where one agent undertakes to route the traffic of another in exchange for a specified 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 (see [9] for a comprehensive survey on this topic). Thus our results shed some light on the computational complexity of linear Generalized Nash Equilibrium with limited interactions.

An earlier version of the present paper, containing a subset of the results, appeared as a technical report [15].

1.1 Description of the linear service-providing game

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 share of the service. These service ratios are given as non-negative rationals $r^i_s$ for every service $s$ and agent $i$ with $s \in S^i$, and they satisfy $\sum_{i: s \in S^i} r^i_s = 1$. In addition, each service $s \in S$ has a per-unit service price $p_s$.

The strategy space of Player $i$ is a bounded polyhedron $P^i \subseteq \mathbb{R}^{d_i}$. Here $d_i \geq |S^i| + |T^i|$, and $P^i$ given in the form

\[ \begin{align*}
A^i_1 x^i + A^i_2 y^i + A^i_3 z^i &\leq b^i, \\
x^i &\geq 0, \\
y^i &\geq 0,
\end{align*} \]

where $x^i \in \mathbb{R}^{S^i}$, $y^i \in \mathbb{R}^{T^i}$, and the matrices $A^i_1$, $A^i_2$, $A^i_3$ can be arbitrary except that $A^i_3$ is assumed to be non-negative. There is also a cost vector $c^i \in \mathbb{R}^{d_i}$. We use the notation $c^i_s$ for the cost of variable $x^i_s$. The three types of variables can be interpreted as follows:

- $x^i_s$, for $s \in S^i$, represents the amount of service $s$ provided by Player $i$,
- $y^i_s$, for $s \in T^i$, is the amount of service $s$ bought by Player $i$,
- $z^i$ is a vector of additional variables.

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 $c_\times^T (x, y, z)$.

In addition to the social cost, players have to pay the prices of services to each other. For a given service $s$, a customer $j$ with $s \in T^j$ pays $p_s y^j_s$ for service $s$, while the income of provider $i$ from service $s \in S^i$ is $p_s x^i_s$. To account for these transfers, we define the 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 $(c^*_p)^T (x^i, y^i, z^i)$ on $P^i$.

The reader may notice that, up to this point, the model does not guarantee that the total amount bought of service $s$ is the same as the total amount sold. The equations corresponding to these conditions are called service equations, and are described below.

### 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 for Player $i$ if

\[ x^i_s = r^i_s \sum_{j: s \in T^j} y^j_s \quad \text{for every } s \in S^i. \]

The equation for a given $s \in S^i$ is called the personal service equation of agent $i$ for service $s$. A strategy vector is feasible if it is feasible for every player. We also use a
1.3 Equilibria

weaker version of these conditions: a strategy vector \((x, y, z) \in P_x\) is weakly feasible if the following weak feasibility equations hold:

\[
\sum_{i \in S^i} x^i_s = \sum_{j \in T^i} y^i_s \text{ for every } s \in S.
\]

The polyhedron of feasible solutions is denoted by \(P_{\text{feas}}\), while the polyhedron of weakly feasible solutions is \(P_{\text{weak}}\). 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 \(P_{\text{feas}}\). We say that \(r\) is a socially optimal service ratio vector if socially optimal solutions are also optimal in \(P_{\text{weak}}\) for \(c\). In other words, the service ratios are socially optimal if no other service ratios can achieve lower social cost.

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 \((x, y, z) \in P_x\) and any \(i \in [n]\) we can replace \((x^i, y^i, z^i)\) by a vector \((\hat{x}^i, \hat{y}^i, \hat{z}^i) \in P^i\) such that the resulting strategy vector \((\hat{x}^i, \hat{y}^i, \hat{z}^i, x^{-i}, y^{-i}, z^{-i})\) is feasible for Player \(i\) (here \(x^{-i}, y^{-i}, z^{-i}\) denote the strategy vectors for all players except for Player \(i\)). We will see that safe instances are guaranteed to have an equilibrium, which implies that they have a feasible solution.

1.3 Equilibria

A feasible solution is an equilibrium if players cannot increase their profit without violating one of their personal service equations, provided that the strategies of other players are fixed. More formally, \((x, y, z) \in P_{\text{feas}}\) is an equilibrium if there is no \(i \in [n]\) and \((\hat{x}^i, \hat{y}^i, \hat{z}^i) \in P^i\) such that \(c^iT(\hat{x}^i, \hat{y}^i, \hat{z}^i) < c^iT(x^i, y^i, z^i)\) and the strategy vector \((\hat{x}^i, \hat{y}^i, \hat{z}^i, x^{-i}, y^{-i}, z^{-i})\) satisfies all personal service equations of Player \(i\). 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 responses 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.

It should be noted that this notion of equilibrium is stronger than the equilibrium notion in constrained games (see e.g. Rosen [18]). Indeed, if we consider the constrained game with the feasible set \(P_{\text{feas}}\), then the feasible response of a player must satisfy the personal service equations of all players, while here it has to satisfy only their own personal service equations. This means that if a player is a customer of a service but not a provider, then they can choose the amount of service to buy irrespective of how much other players buy.

We include another characterization of equilibria using the affine subspaces defined
by personal service equations. For \( i \in [n] \) and \( y \in \mathbb{R}^{T^i} \times \cdots \times \mathbb{R}^{T^m} \), let
\[
\Phi^i(y) = \{(x^i, y^i, z^i) : x^i, y^i \text{ satisfy} \]
\[
\hat{x}^i_s = r^i_s \left( \hat{y}^i_s + \sum_{j \neq i : s \in T^j} y^j_s \right) \quad \text{if } s \in S^i \cap T^i \quad \text{and} \]
\[
\hat{x}^i_s = r^i_s \sum_{j : s \in T^j} y^j_s \quad \text{if } s \in S^i \setminus T^i \}.
\]

In other words, \( \Phi^i(y) \) is an affine subspace consisting of the vectors \((\hat{x}^i, \hat{y}^i, \hat{z}^i)\) that satisfy the personal service equations of Player \( i \) with respect to \( y^{-i} \). Let \( \Phi(y) = \Phi^1(y) \times \cdots \times \Phi^n(y) \). Observe that \( \text{lin}(\Phi(y)) \), the linear hull of \( \Phi(y) \), 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 \( \Phi(y) \). Moreover, \((x, y, z) \in P_{\text{feas}} \) is an equilibrium if and only if it is optimal in \( P_\times \cap \Phi(y) \) for the objective function \( c_p \).

It is easy to construct instances where no equilibrium exists. However, existence of equilibria in safe instances can be proved by a standard application of Kakutani’s fixed point theorem.

**Theorem 1.1.** In a safe instance there always exists an equilibrium.

**Proof.** Let \( C = \{ y : \exists x, z \text{ s.t. } (x, y, z) \in P_\times \} \). Note that \( C \) is compact and convex, and safeness of the instance means that \( P_\times \cap \Phi(y) \) is non-empty for any \( y \in C \). For \( y \in C \), let
\[
\varphi(y) = \{ \hat{y} \in C : \exists \hat{x}, \hat{z} \text{ s.t. } (\hat{x}, \hat{y}, \hat{z}) \text{ is optimal for } c_p \text{ in } P_\times \cap \Phi(y) \}.
\]

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 \( \varphi \) 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 \( \lim_{k \to \infty} \hat{x}_k = \hat{x} \) and \( \lim_{k \to \infty} \hat{z}_k = \hat{z} \) such that \((\hat{x}_k, b_k, \hat{z}_k)\) is optimal for \( c_p \) in \( P_\times \cap \Phi(a_k) \). The convergence implies that \((\hat{x}, b, \hat{z})\) is optimal for \( c_p \) in \( P_\times \cap \Phi(a) \), and thus \( b \in \varphi(a) \).

By Kakutani’s Theorem, there exists a vector \( y \) with \( y \in \varphi(y) \). By the definition of \( \varphi \), 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. \( \square \)

### 1.4 Summary of the results

In Section 2 we investigate the structure of equilibria. We first show that the set of equilibria is always the (perhaps empty) union of some faces of \( P_{\text{feas}} \). If the service ratios are socially optimal, then we can modify service prices in such a way that every socially optimal feasible solution becomes an equilibrium. In fact, we prove a stronger statement: every socially optimal feasible solution becomes optimal for the cost vector \( c_p \) in the polyhedron \( P_\times \), so not even a coalition can profit from changing their behaviour. This is a generalization of the result of Agarwal and Ergun [3] on capacity exchange prices in service network alliances (see Section 1.5.1).
1.4 Summary of the results

Theorem 1.2. Suppose that the service ratios are socially optimal. Then it is possible to compute service prices $p$ in polynomial time such that every socially optimal solution $(x, y, z) \in P_{\text{feas}}$ is optimal for the cost vector $c_p$ in the polyhedron $P_x$. If $c_i \geq 0$ for every $i$, then the price $p_i$ can be chosen to be nonnegative.

In case of sub-optimal service ratios, it may be impossible to find prices for which there exists an equilibrium. However, as the following result shows, this can happen only if some players are both providers and customers of the same service.

Theorem 1.3. 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 prove that finding an equilibrium is PPAD-complete, even if there are only two players.

Theorem 1.4. It is PPAD-complete to find an equilibrium in a safe instance of the two-player linear service-providing game.

In Section 4 we present a polynomial-time algorithm for finding an equilibrium in the special case when each strong component of the digraph $D^*$ representing the provider-customer relationships is a simple cycle. The digraph $D^* = ([n], A^*)$ 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.

Our polynomial algorithm follows from a new result on the approximation of fixed points in a certain class of fixed-point problems. Given $m$ interval-valued functions $\varphi_1, \ldots, \varphi_m$ 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 $\varepsilon$ such that there is a cyclically fixed vector $x$ with $x_i \in I_i$ ($i \in [m]$).

We present another polynomial-time solvable case in Section 5.

Theorem 1.7. An equilibrium can be found in polynomial time in safe instances where Player 1 is the sole customer of service $s_1$, and is the sole provider of all other services.

In view of these results, the study of other special cases might offer new insights on the borderline between polynomially solvable and PPAD-complete equilibrium problems. The results imply that finding an equilibrium is in P if $D^*$ has 3 arcs, but the case of 4 arcs remains open.
1.5 Relationship 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 gave the following model for multicommodity flow games in service network alliances. There are agents, and a common directed graph $D = (V, A)$. Each arc $a \in A$ has a capacity $u_a$, and given ownership ratios $r^i_a$ ($i \in [n]$) with $\sum_{i=1}^n r^i_a = 1$. 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 $f_q \leq u_q$ from $s_q$ to $t_q$; the revenue generated by this flow is $w_q f_q$. The total flow traversing arc $a$ should not exceed the capacity $u_a$.

The main result of 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 $f^i_a$, then $i$ has to pay an amount of $p_a f^i_a$, which is distributed among the agents according to the ratios $r^j_a$ ($j \in [n]$). Since the payment to themselves can be ignored, agent $i$ actually pays an amount of $(1 - r^i_a)p_a f^i_a$ 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 $r^i_a > 0$. The service ratios are determined by the values $r^i_a$. In order to define the polyhedron $P^i$, we consider the vector variable $z^i_a$ to be composed of vector variables $z^i_q$ for each demand $q \in Q^i$. In $P^i$, the variables $z^i_q$ ($q \in Q^i$) describe the multicommodity flow polyhedron of agent $i$, with cost defined as the opposite of revenue. The variable $y^i_a$ 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 $x^i_a$ has the single constraint $0 \leq x^i_a \leq r^i_a u_a$; the variables $x$ and $y$ have cost 0.

With these definitions, feasibility is the same as in the original problem, and the $c^i_p$-cost of agent $i$ represents the opposite of his profit 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 $0 \leq x^i_a \leq r^i_a u_a$ for every $i$, 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 $\sum_{i:r^i_a > 0} x^i_a$ for any arc $a$. Therefore Theorem 1.2 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 , $n$ agents have separate networks $D^i = (V, A^i)$ on a common node set. Each arc $a$ has a cost $c_a$ and a capacity $u_a$. 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 $\gamma_a$. If $a = w \in B_i$ 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.

The MMF problem can be modeled in our framework by assigning a service to a face of $P$ for each contractual arc, with a size $q_i$. The set of contractual demands of agent $i$ consists of vector variables $q_i$. In exchange, $i$ pays an amount of $q_i p_a$ to the provider. A contractual arc has a price $p_a$ and a size $q_i$. This way, we obtain the same notion of equilibrium as in [5].

## 2 Structure of equilibria and price modifications

In this section we prove Theorems 1.2 and 1.3, and give an example where no prices can guarantee an equilibrium. We start by explaining the polyhedral tools used in the proofs.

For a polyhedron $P$ and a face $F$ of $P$, let $\text{opt.cone}(F, P)$ 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 $\text{relint}(X)$, while $\text{lin}(X)$ is the linear translation of the affine hull of $X$. The following lemma is a direct consequence of Farkas’s Lemma.

**Lemma 2.1.** Let $P_1$ be a polyhedron, $\Pi$ 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 $x' \in P_2$, $(w + a)^{\top} x = w^{\top} x + a^{\top} x \geq w^{\top} x' + a^{\top} x = (w + a)^{\top} x'$, 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 $w^{\top} y > 0$ but $(w' + a)^{\top} y \leq 0$ 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 $\text{relint}(F_2)$. Then $\text{opt.cone}(F_1, P_1)$ 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 $w^{\top} y > 0$.

Part (ii) follows from part (i). $\square$
The first application of the lemma is a proof that if a vector in $P_\text{feas}$ is an equilibrium, then the minimal face containing it consists of equilibria. Recall the definition of $\Phi(y)$ in [4].

**Lemma 2.2.** The set of equilibria is the (perhaps empty) union of some faces of $P_\text{feas}$. As a consequence, the set of equilibria is either empty, or there is an equilibrium that is a vertex of $P_\text{feas}$ 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 $P_\text{feas}$ 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. 

Next we show that if the service ratios are socially optimal, then there is a price vector such that every socially optimal feasible solution is optimal for $c_p$ in the polyhedron $P_\times$.

**Proof of Theorem 1.2** We assume that $P_\text{feas}$ is non-empty. Let $F_{\text{opt}}$ be the optimal face in $P_{\text{weak}}$ minimizing the cost $c$. By the assumption that service ratios are optimal, all socially optimal solutions are on $F_{\text{opt}}$. Let $F_\times$ be the minimal face of $P_\times$ which contains $F_{\text{opt}}$. Let furthermore $H$ be the subspace determined by the equations

$$\sum_{i: s \in S_i} x_i^s = \sum_{j: s \in T_j} y_j^s \text{ for every } s \in S.$$ 

We apply Lemma 2.1 with the definitions $P_1 = P_\times$, $\Pi = H$, $P_2 = P_{\text{weak}}$, $F_1 = F_\times$, and $F_2 = F_{\text{opt}}$. By consequence, there is a vector $h \in \text{lin}(H)^\perp$ for which $c + h$ is in $\text{relint}(\text{opt.cone}(F_\times, P_\times))$. 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 $x_i^s$ is $-p_s$,
- The component of $h$ corresponding to $y_j^s$ 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 $P_\times$.

To prove the second part of the theorem, assume that $c_i^s$ is nonnegative for every $i$. Suppose that $p_s$ is negative, and let $(x, y, z)$ be a socially optimal solution. Since $c_i^s - p_s$ is positive, the $c_p$-cost decreases if we decrease $x_i^s$. Since the describing matrices $A_1^s$ are nonnegative, the modified vector is also in $P_\times$ if $x_i^s \geq 0$. Therefore $p_s < 0$.
implies that $x_i = 0$ for every $i$ in every socially optimal solution. We claim that if $p'$ is obtained from $p$ by setting $p'_s = 0$, then $p'$ also satisfies the properties in the theorem. Indeed, there is a vector $(x', y', z')$ of minimum $c_{p'}$-cost in $P_\times$ for which $x'_i = 0$ for every $i$, because we can decrease $x'_i$ without increasing the $c_{p'}$-cost. Now $c^T_{p'}(x', y', z') \leq c^T_p(x', y', z')$ 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 $x'_i = 0$ and $y'_i = 0$ for every $i$, hence $c^T_{p'}(x, y, z) = c^T_p(x, y, z) \leq c^T_p(x', y', z')$, which means that $(x, y, z)$ 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 $x_1, x_2, x_1', x_2'$ and $y_1, y_2, y_1', y_2'$. The possible strategies of the two players are defined by the two polyhedra

$$P^1 = \{(x^1, y^1) : x^1 \geq 0, y^1 \geq 0, x^1_1 \leq 1, y^1_1 + y^1_2 = 2, 2x^1_1 \leq y^1_1\},$$

$$P^2 = \{(x^2, y^2) : x^2 \geq 0, y^2 \geq 0, x^2_1 \leq 1, y^2_2 + y^2_1 = 2, 2x^2_1 \leq y^2_2\}.$$ 

Let $r^1_1 = r^2_1 = r^2_2 = \frac{1}{2}$. 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 $x^1_1 = x^2_1 = x^2_2 = 1, y^1_1 = 2, y^1_2 = 0, y^2_1 = 0, y^2_2 = 2$. 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 $x^1_2$ and $y^1_2$ 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 $x^2_1$ and $y^2_1$ to 2. Therefore there are no prices for which an equilibrium exists.

We now prove Theorem 1.3 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 1.3** As in the proof of Theorem 1.2, let $H$ be the subspace defined by the weak feasibility equations. Let furthermore

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

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

We apply Lemma 2.1 with the definitions $P_1$, $H = P_2 = P_{\text{feas}}$, $F_1$, and $F_2 = F_{\text{opt}}$. By consequence, there is a vector $h \in \text{linc}(H)^\perp$ for which $c + h$ is a member of $\text{relint}(\text{optcone}(F_1, P_1))$. Since $h$ is in $\text{linc}(H)^\perp$, 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 their personal service equations actually preserves all values of $x^i$, so the obtained vector remains in $P_1$. This means that every socially optimal solution is in equilibrium because it is optimal for $c_p$ in $P_1$.

3 PPAD-completeness

In this section we show that the problem of finding an equilibrium in a safe instance 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 [17]. The proof in [17] is a reduction to Sperner’s Lemma (via Brouwer’s fixed point theorem), and it only works if the set-valued function $\varphi$ has the following property: given a sufficiently small simplex that is known to contain a fixed point of $\varphi$, we can compute a fixed point in polynomial time. This property holds for the function $\varphi$ defined in the proof of Theorem 1.1 because equilibria form the union of faces of $P_{\text{feas}}$ by Lemma 2.2.

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 3.1** ([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 about 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 $A \in \mathbb{Q}^{m \times m}$. We can also assume without loss of generality that the entries of $A$ are positive. Second, there are several ways to define approximate equilibria; we use a definition in [6]: $x^*$ is an $\varepsilon$-well supported approximate symmetric Nash equilibrium if $x^*_j > 0$ implies $\sum_{k=1}^m a_{jk}x^*_k > \max_{i \in [m]} \sum_{k=1}^m a_{ik}x^*_k - \varepsilon$. Finally, it is convenient to set $\alpha = 1$. To sum up, we use the following form of the theorem.

**Corollary 3.2** ([6]). The problem of computing a $\frac{1}{m}$-well supported approximate symmetric Nash equilibrium of a symmetric 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 the linear service-providing game. Given a symmetric game defined by a matrix $A \in \mathbb{Q}^{m \times m}$, we construct a safe instance with two players, with the property that any equilibrium of the safe instance corresponds to a $\frac{1}{m}$-well supported approximate symmetric Nash equilibrium. This is enough for showing
hardness, since an algorithm for finding an equilibrium in safe instances would solve \( \frac{1}{m} \text{-APPROXIMATE 2-NASH} \).

The construction involves two players and 2m services; services 1, \ldots, m are provided by Player 2 to Player 1, while services \( m + 1, \ldots, 2m \) are provided by Player 1 to Player 2. For simplicity, we will denote the variables associated to the \( j \)-th service by \( x_j^1, y_j^1 \) \( (j = 1, \ldots, m) \) and \( x_j^2, y_j^2 \) \( (j = m + 1, \ldots, 2m) \). All service prices are 0. The linear system of Player 1 is the following.

\[
\begin{align*}
\sum_{j=1}^{m} y_j^1 &= 1 \\
z_j^1 &\geq y_j^1 + x_{m+j}^1 - 1 \quad (j = 1, \ldots, m) \\
x^1, y^1, z^1 &\geq 0 \\
\min \sum_{j=1}^{m} z_j^1
\end{align*}
\]

Player 2 has the following linear system.

\[
\begin{align*}
z_j^2 &\geq \sum_{k=1}^{m} a_{jk} x_k^2 \\
z_0^2 &\geq z_j^2 \quad (j = 1, \ldots, m) \\
y_{m+j}^2 &= m(z_0^2 - z_j^2) \\
x^2, y^2, z^2 &\geq 0 \\
\min \sum_{j=0}^{m} z_j^2
\end{align*}
\]

Suppose that \((x, y, z)\) is an equilibrium. In particular, \((x, y, z)\) is feasible, so \(x_j^1 = y_j^1\) and \(x_{m+j}^1 = y_{m+j}^1 \ (j = 1, \ldots, m)\). Let \( M = \max_{j \in [m]} \sum_{k=1}^{m} a_{jk} x_k^2 \). Since Player 2 cannot improve, \(z_j^2\) must be equal to \(\sum_{k=1}^{m} a_{jk} x_k^2\), and \(z_0^2\) must be equal to \(M\). Let \(i \in [m]\) be an index for which \(\sum_{k=1}^{m} a_{ik} x_k^2 = M\); then \(y_{m+i}^2 = 0\), so \(x_{m+i}^1 = 0\).

We know that Player 1 cannot improve the cost if the values of the variables \(x_{m+j}^1\) are fixed to \(x_{m+j}^1 \ (j = 1, \ldots, m)\). In particular, we can consider the feasible strategy of Player 1 obtained by setting \(y_i^1\) to 1, \(y_j^1\) to 0 \((j \neq i)\), \(z_i^1\) to 0, and \(z_j^1\) to \(\max\{0, \hat{x}_{m+j}^1 - 1\}\) \((j \neq i)\). The cost of this solution is \(\sum_{j \neq i} \max\{0, \hat{x}_{m+j}^1 - 1\}\), which should not be lower than \(\sum_{j=1}^{m} z_j^1 = \sum_{j=1}^{m} \max\{0, y_j^1 + \hat{x}_{m+j}^1 - 1\}\). This means that \(y_j^1 + \hat{x}_{m+j}^1 \leq 1\) for every index \(j\) such that \(y_j^1 > 0\). Let \(x_j^* = y_j^1 \ (j = 1, \ldots, m)\); then \(x^*\) satisfies the following properties.

(i) \(\sum_{j=1}^{m} x_j^* = 1\)

(ii) \(x^* \geq 0\)

(iii) \(\max_{j \in [m]} \sum_{k=1}^{m} a_{jk} x_k^* = M\)

(iv) \(x_j^* + m(M - \sum_{k=1}^{m} a_{jk} x_k^*) \leq 1\) for every index \(j\) such that \(x_j^* > 0\).
By the first two properties, $x^*$ is a strategy vector for the Nash equilibrium problem, while the last property implies that $M - \sum_{k=1}^{m} a_{jk} x_k^* \leq \frac{1 - x_j^*}{m} \leq \frac{1}{m}$ for every index $j$ such that $x_j^* > 0$. Thus $x^*$ is a $\frac{1}{m}$-well supported approximate symmetric Nash equilibrium.

### 4 Polynomial-time algorithm for simple cycles

In the PPAD-completeness proof in Section 3, we reduced approximate 2-Nash to 2-player safe instances with both players providing many services to the other. 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 for every $i, j$ with $i \neq j$ (so $D^*$ does not have loops).

With this definition, the directed graph corresponding to the hard instance constructed in the proof of Theorem 1.4 is a 2-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 an equilibrium. Note that a pair of oppositely directed arcs is also considered to be a simple cycle.

The main tool of the proof is an algorithm for finding approximate fixed points for a special class of set-valued functions, as described in Theorem 1.6. Let $\varphi_i : [0, 1] \to \mathcal{P}([0, 1])$ ($i \in [m]$) be a set-valued function such that $\varphi_i(t)$ is a non-empty interval for every $t \in [0, 1]$, and the graph of $\varphi_i$ is closed. We are interested in finding a fixed point of the function $(x_1, x_2, \ldots, x_m) \mapsto (\varphi_m(x_m), \varphi_1(x_1), \ldots, \varphi_{m-1}(x_{m-1}))$. 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 count basic arithmetic operations as one step. Of course, it is impossible to compute a cyclically fixed vector exactly in this oracle model. However, as stated in Theorem 1.6, we can approximate it in a polynomial number of steps, in the sense that we can find arbitrarily small intervals whose direct product contains a cyclically fixed vector.

**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 the value \(z\) 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 \(\psi_m\) 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 \(\varphi_i\) 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 \(\varphi_i\): 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 \(\psi_i = \varphi_{i-1} \circ \varphi_{i-2} \circ \cdots \circ \varphi_0 \circ \varphi_{i-1} \circ \cdots \circ \varphi_i\).

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 4.1.** The \(m\)-th step of the itinerary of \((i, a_i)\) is not smaller than \(a_i\), and the \(m\)-th step of the itinerary of \((i, b_i)\) is not greater than \(b_i\). \(\square\)

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

Let us define functions \(\varphi^*_i : I_i \rightarrow \mathcal{P}(I_{i+1})\) 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 function \(\psi^*_k = \varphi^*_k \circ \varphi^*_{k-1} \circ \cdots \circ \varphi^*_1\). We can observe that \(\psi^*_m(I_1) \subseteq \psi^*_k(I_1)\) for any \(k\). We define \(\psi^*_0\) to be the identity function.

**Claim 4.2.** 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 \(\varphi_k \circ \psi^*_m(t) > b_{k+1}\) for every
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 \(\psi^*_i(I_1)\), provided that \((i - 1)m + k + j \geq 0\). We show this by induction on \(j\), first for \(j < m - k\). In this case \(\psi^*_{i-1}(I_1)\) is non-empty and so the set \(\phi_{k+j} \circ \psi^*_i(I_{m+k+j-1})\), which contains the \(j\)-th step of the itinerary, is equal to \(\psi^*_i(I_{m+k+j})\).

Next, we show that the \(j\)-th step of the itinerary is in \(\psi^*_i(I_{m+k+j})\) also for \(m - k \leq j < m\). The difficulty here is that the itinerary proceeds according to the function \(\phi_{k+j}\), which, as opposed to \(\phi_{k+j}\), 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 \(\psi^*_{i-1}(I_{m+k+j})\) (because \(\psi^*_{i-1}(I_1)\) is non-empty), so the itinerary of \((k + j + 1, b_{k+j+1})\) leads to \(b_{k+j+1}\), 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 4.1.

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 4.1.

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 4.3.** \(R\) contains a fixed point of \(\psi_m\).

**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 \(t \in R_k \cap I_{k-1}^*, \text{ then either } \psi_k^*(a_{k-1}^*, t) \subseteq \psi_k^*([t, b_{k-1}^*]) \text{ or vice versa because both are sub-intervals of } I_{k-1}\). 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 $I^* = [a^*, b^*] \subseteq \mathbb{R}^*$ such that $\psi^*_{m}(I^*) = R$. Now it follows from the closed graph property of $\psi^*_{m}$ on $I^*$ that it has a fixed point in $I^*$.

By definition, a fixed point of $\psi^*_{m}$ implies the existence of a cyclically fixed vector $x$ with $x_i \in I$, $(i \in [m])$. This concludes the proof of the theorem.

We now show that if the graph of each function $\varphi_i$ 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 4.4.** Let $\varphi_i : [0, 1] \to \mathcal{P}([0, 1])$ $(i \in [m])$ be given as in Theorem 1.6, with the additional property that the graph of each function $\varphi_i$ is a 2-dimensional polyhedral complex that can be described by inequalities of bit-size $M$. Then in $O(m^2 M^2)$ steps we compute a cyclically fixed vector.

**Proof.** Let $\varepsilon = \exp(-M^2)$. Since the graph of the function $\varphi_i$ is a 2-dimensional polyhedral complex of bit-size $M$, the choice of $\varepsilon$ guarantees that all vertices of this polyhedral complex in the interior of $I_i \times [0, 1]$ (if they 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 $\varphi_i$ has no vertex in the interior of $I_i \times [0, 1]$ $(i \in [m])$. This means that the set-valued functions $\varphi_i$ 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 a fixed point can be found by solving an LP.

We are now ready to prove that if the strongly connected components of $D^*$ are simple directed cycles, then an equilibrium in a safe instance of the linear service-providing game can be found in polynomial time.

**Proof of Theorem 1.6.** 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 $k$th 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 all 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 $\varphi_j$ 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 $(\tilde{x}^{i_{j+1}}, \tilde{y}^{i_{j+1}}, \tilde{z}^{i_{j+1}}) \in P^{i_{j+1}}$ such that, together with the values already fixed, it satisfies all personal service equations of player $i_{j+1}$. Let us take the set of such
Section 5. Another solvable case

vectors that have minimum $c_p$-cost, and let $\varphi_j(t)$ consist of the possible $\hat{y}^{i+1}_j$ values in these vectors. This is a non-empty subinterval of $[\ell_{j+1}, u_{j+1}]$. A value in $\varphi_j(t)$ can be computed using linear programming, and the graph of $\varphi_j$ 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 4.4 (by appropriately scaling the functions $\varphi_j$) 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.

\begin{proof}[Proof of Theorem 1.7] The set $\{t \in \mathbb{R} : t = y_{s_1}^1 \text{ for some } (x^1, y^1, z^1) \in P^1\}$ is clearly an interval; let us denote it by $[\ell, u]$. For $t \in [\ell, u]$, let
\[ P(t) = \{(x^{-1}, y^{-1}, z^{-1}) \in P_2 \times \cdots \times P_n : x_{s_1}^i = r_{s_1}^i t \text{ for every } i \text{ with } s_1 \in S^i\}, \]
and let $Q = \cup_{t \in [\ell, u]} P(t)$; observe that $Q$ is a polytope. We define a set-valued function $\varphi_1 : [\ell, u] \to \mathcal{P}(P_2 \times \cdots \times P_n)$ by
\[ \varphi_1(t) = \{(x^{-1}, y^{-1}, z^{-1}) \in P(t) : (x^{-1}, y^{-1}, z^{-1}) \text{ is optimal for } c_p \text{ in } P(t)\}. \]

It follows from Lemma 2.1 that $\varphi_1(t)$ is the intersection of $P(t)$ with a face of $Q$, and $\cup_{t \in [\ell, u]} \varphi_1(t)$ is the union of some faces of $Q$. We also define the polyhedron
\[ P^1(y^{-1}) = \{(x^1, y^1, z^1) \in P^1 : (x^1, y^1, z^1) \text{ satisfies the personal service equations of Player 1 w.r.t } y^{-1}\}, \]
and an interval-valued function $\varphi_2 : Q \to \mathcal{P}([\ell, u])$ by
\[ \varphi_2((x^{-1}, y^{-1}, z^{-1})) = \{t \in [\ell, u] : \exists (x^1, y^1, z^1) \text{ optimal for } c_p \text{ in } P^1(y^{-1}) \text{ s.t. } y_{s_1}^1 = t\}. \]

Finally, let $\psi = \varphi_2 \circ \varphi_1$.

Intuitively, if we fix $y_{s_1}^1$ at $t$, then the other players can find an optimal $(\hat{x}^{-1}, \hat{y}^{-1}, \hat{z}^{-1})$ for $c_p$, since they are not providers of the other services; these optimal solutions form $\varphi_1(t)$. Now if we fix $(\hat{x}^{-1}, \hat{y}^{-1}, \hat{z}^{-1})$, then Player 1 can find an optimal $(\hat{x}^1, \hat{y}^1, \hat{z}^1)$ for $c_p$, and $\psi(t)$ consists of the possible values of $\hat{y}^1_{s_1}$ that arise this way. Since the instance is safe, $\psi(t)$ is nonempty for every $t \in [\ell, u]$. The graph of $\psi$ is closed because $\varphi_1$ and $\varphi_2$ both have the closed graph property.

EGRES Technical Report No. 2016-15
Section 6. Open questions

The algorithm for finding an equilibrium will have two phases. The first phase is similar to the algorithm in Section 4. Let $\varepsilon = \exp(-M^2)$, where $M$ is the size of the input. Initially, let $a = t$, $b = u$, and $t = (a+b)/2$. We compute a value $\hat{t} \in \psi(t)$ by first computing a vector $(\hat{x}^{-1}, \hat{y}^{-1}, \hat{z}^{-1}) \in \varphi_1(t)$, and then a value $\hat{t} \in \varphi_2((x^{-1}, y^{-1}, z^{-1}))$; both of these can be done by linear programming. If $\hat{t} \leq t$, then let $b = t$; otherwise let $a = t$. These steps are repeated until $b - a \leq \varepsilon$.

As in the proof of Corollary 4.4, we make an additional step in the algorithm. Since the polytope $Q$ has vertices of bit-size $M$, the choice of $\varepsilon$ guarantees that if $Q$ has vertices in $P(t)$ for some $t \in [a, b]$, then all of these vertices are in the same $P(t^*)$. Furthermore, this $t^*$ can be computed in polynomial time. We compute a value $\hat{t} \in \psi(t^*)$. If $\hat{t} \leq t^*$, then let $b = t^*$; otherwise let $a = t^*$. This is the end of the first phase, and at this point the following properties hold.

(i) There is a face $F$ of $Q$ such that, for any $t \in [a, b]$, $\varphi_1(t) = F \cap P(t)$ (this is due to the choice of $t^*$).

(ii) We have already computed a vector $(x, y, z)$ such that $(x^{-1}, y^{-1}, z^{-1}) \in \varphi_1(a)$, $(x^1, y^1, z^1)$ is optimal for $c_p$ in $P_1(y^{-1})$, and $y^1_{s_1} \geq a$.

(iii) We have already computed a vector $(\tilde{x}, \tilde{y}, \tilde{z})$ such that $(\tilde{x}^{-1}, \tilde{y}^{-1}, \tilde{z}^{-1}) \in \varphi_1(b)$, $(\tilde{x}^1, \tilde{y}^1, \tilde{z}^1)$ is optimal for $c_p$ in $P_1(\tilde{y}^{-1})$, and $\tilde{y}^1_{s_1} \leq b$.

Property (iii) implies that if $t = \lambda a + (1 - \lambda)b$ for some $0 \leq \lambda \leq 1$, then $\varphi'_1(t) := \lambda(x^{-1}, y^{-1}, z^{-1}) + (1 - \lambda)(\tilde{x}^{-1}, \tilde{y}^{-1}, \tilde{z}^{-1})$ is in $\varphi_1(t)$. Let $\varphi'_2$ be the restriction of $\varphi_2$ to the segment $S$ between the points $(x^{-1}, y^{-1}, z^{-1})$ and $(\tilde{x}^{-1}, \tilde{y}^{-1}, \tilde{z}^{-1})$. By properties (ii), (iii) and Kakutani’s fixed point theorem, there exists $t \in [\ell, u]$ and $s \in S$ such that $s \in \varphi'_1(t)$ and $t \in \varphi'_2(s)$. By Corollary 4.4, we can find such $t$ and $s$ in polynomial time, and this gives an equilibrium. 

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 four services, two of them offered by Player 1 to Player 2, and two offered by Player 2 to Player 1. Another promising direction is to find a common generalization of Theorems 1.3 and 1.7.

From the point of view of mechanism design, it would be interesting to find bargaining mechanisms that lead to prices satisfying the property in Theorem 1.2 or at least some approximate version of it.

Acknowledgements

We would like to thank the authors of [5] for fruitful discussions on the topic. The research was supported by grant no. K109240 of the Hungarian Scientific Research Fund (OTKA). The first author is supported by the MTA Bolyai Research Scholarship.
References


