MSc Thesis Applied Mathematics

Equilibria in the Two-Stage Facility Location Game With Unsplittable Clients

M.C. Vos

Supervisor: prof. dr. M. J. Uetz
Supervisor: dr. A. Skopalik
Committee member: dr. J.M. Meylahn

July 21, 2023

Department of Applied Mathematics
Faculty of Electrical Engineering,
Mathematics and Computer Science
Chair of Discrete Mathematics and Mathematical Programming

UNIVERSITY OF TWENTE.

Acknowledgements

I'm grateful for the opportunity to dedicate the past six months to working on this thesis. Focusing extensively on a single problem was quite different from my earlier experiences. The subject matter of this thesis was quite theoretical, necessitating intricate notation and complex proof techniques. As such, the most important skill I learned during these months was the clear and concise formulation of mathematical concepts and results. While there were some frustrating moments, particularly related to some of the open problems I was not able to solve, I am quite content with the overall result. Of course, I could not have done this thesis alone, which is why I would like to thank everyone who helped me along the way.

First and foremost, I want to thank Alexander Skopalik and Marc Uetz for being my supervisors, for the useful discussions during our meetings, and for the continuous feedback given during the entire process. I would like to thank Simon Krogmann; his presence and expertise in the area of my research added invaluable insights during his three-month stay at the University of Twente. I want to thank Matthew for proofreading my work. Lastly, I want to thank Janusz Meylahn for taking the time to be a member of the committee and reading through this (rather lengthy) report.

Equilibria in the Two-Stage Facility Location Game With Unsplittable Clients

M.C. Vos

July 21, 2023

Abstract

We consider a non-cooperative facility location game where facility and client players strategically interact on a given host graph. This is represented as a sequential game comprised of two stages, where each stage is a simultaneous game with disjoint player sets. In the first stage, each facility player chooses a vertex of the host graph on which to place their facility. In the second stage, each client player strategically chooses a probability distribution over the facilities in their neighborhood, denoting the likelihood of visiting each facility. Facility players aim to maximize the expected attracted purchasing power. Clients aim to minimize the expected congestion they encounter, where we assume that the congestion at each facility is proportional to the total purchasing power of clients patronizing the facility. In contrast to recent publications on similar games, we assume that clients cannot split up their purchasing power over multiple facilities. We demonstrate the existence of instances of this game that admit no subgame perfect equilibria and show that determining the existence of such equilibria is NP-hard. Additionally, we establish sufficient conditions for the existence of such equilibria, notably the condition of all clients possessing equal purchasing power. For this unweighted game, we present an algorithm to compute subgame perfect equilibria and analyze their efficiency by providing bounds on the price of anarchy and stability. Lastly, we examine the existence of approximate subgame perfect equilibria.

Contents

1 Introduction

1.1 History and Context

Facility location games have been a common topic of research in the fields of economics, mathematics, and theoretical computer science for several decades. In the broadest sense, facility location games deal with a set of facilities that must be placed on some underlying space to serve a set of clients. The chief distinction between different settings within the field of facility location games is whether it is cooperative or non-cooperative/competitive.

In cooperative settings, there is some central authority that chooses the location for all facilities. The goal in such games is to determine what placements are optimal with respect to some function, usually related to the distance from customers to the facilities. Examples include the geographical distribution of hospitals, police, or fire stations by a government. Techniques from optimization and combinatorics are well-suited to solve such problems.

In this thesis, we focus on competitive facility location games. Competitive facility location games deal with the placement of facilities (e.g., ice-cream trucks, shops, bars) by competing players. These players selfishly place their facilities to maximize some individual payoff function. These games are often analyzed using game-theoretic techniques.

The origin of this field is Hotelling's [14] seminal paper from 1929. This work concerns a model of a one-dimensional space (e.g., a street) on which multiple competing entrepreneurs want to open a facility. Like most subsequent work, Hotelling's primary focus was developing insight into the equilibrium patterns within these games. He showed the importance of the geographical location of a facility in relation to that of the competitor(s) in competitive markets. He proved what we now know as the *principle of minimum differentiation*, the tendency for competing businesses to make their product and geographical location as similar as possible. His model was refined to the *Hotelling-Downs model* by Downs [7] when he applied it in a more abstract way to the field of political science. He represents the political spectrum as a line, on which all voters are located according to their political ideology. The location chosen by a party represents its political position. In this model, the clients (i.e., the customers/voters) choose the facility (i.e., the store/party) closest to them on the line.

The majority of subsequent literature in the field results from changing this basic model in various ways. Examples are: relaxing different assumptions on this basic model, adding more decision variables (e.g., price), changing the space on which the game is played (e.g., d-dimensional spaces, networks), and changing the behavior of the clients. Much of this was influenced by the wide range of applications of the basic model. For a broad overview of results in different dimensions of choice, we refer to survey papers ([13], [20]).

The client behavior in the Hotelling-Downs model is very basic. In most applications, clients determine which facility is most attractive based on more than just the travel distance. Modeling this gets especially involved when this depends on the choices of other clients. Kohlberg [16] assumed that the clients choose a facility based on both the distance and the congestion at each facility, the latter depending on the market share of the facility. One could view this as the clients competing with each other and having to strategize accordingly. In Kohlberg's model, equilibria do not exist for games with three or more players.

Another interesting assumption regarding client behavior is considered by Feldman et al. [10]. They assume that clients have some maximum distance that they are willing to travel and that if there is no facility within this range, they simply choose not to patronize any facility. For this model, equilibria always exist, independent of the number of facility players.

Recently, Krogmann et al. [17, 18] published two papers on a model which combines the models of Kohlberg and Feldman. They move the game to a graph network, with a discrete, weighted client located on each vertex. The directed edges of the graph encode which locations can reach other locations, to easily represent the limited ranges of the clients. The game is played in two stages. In the first stage, each facility player selfishly chooses a location (vertex) for their facility. Then, given this facility placement, the clients selfishly distribute their purchasing power over the facilities they can reach. The facility and client players thus both face strategic decisions, resulting in a sequential game of two stages, where each stage is a simultaneous game. In their first paper [17], the clients aim to minimize their maximum waiting time. In their second paper [18], the clients instead aim to minimize their total waiting time. The authors show that subgame perfect equilibria always exist in the former case, but not necessarily in the latter case.

In this thesis, we present a third type of client behavior for this two-stage facility location game. We consider unsplittable clients: clients who cannot distribute their purchasing power over different facilities and aim to minimize the expected waiting time (congestion) at the one facility they patronize. As we allow clients to randomize their behavior, i.e., play mixed strategies, we consider the expected waiting time of each client, which is related to the expected load on each facility. We show that for this client behavior, subgame perfect equilibria do not exist in general, and find classes of problems where their existence is guaranteed.

1.2 Further Related Work

There is ongoing research in various fields within the broader topic of competitive facility game analysis. One area that has seen significant activity is the study of (strategy-proof) mechanism design for facility location problems. Strategy-proof mechanisms are rules designed to discourage players from misrepresenting their intended actions to gain an advantage in situations where information asymmetry is a factor. The objective of such rules is to maximize the social welfare of the clients, often measured as some function of their distance to the closest facility. Procaccia and Tennenholtz [26] studied this problem on a line, providing tight bounds on the approximation ratio for certain mechanisms. Subsequently, researchers explored different underlying spaces and objective functions [2, 19, 11, 10]. Moreover, there has been growing interest [4] in games involving "obnoxious" facilities, where clients aim to maximize their distance from facilities due to factors like noise or pollution. We refer to the survey by Chan [3] for an extensive overview of mechanism design for facility location games.

Another type of facility location game is the "Voronoi game". In this game, two competing players take turns placing facilities in a given space. The clients then patronize the closest facility. The goal for the facility players is then to maximize the portion of the clients that patronizes one of their facilities. Ahn et al. [1] studied the game played on a line segment or a circle, while Cheong et al. [5] examined the "One-Round" Voronoi game, where one player places all their facilities first, followed by their opponent. Both publications analyze optimal strategies in such games. Dürr et al. [8] analyzed Nash equilibria in a variation of Voronoi games played on a graph with more than two players, where each player is only allowed to place one facility.

Lastly, numerous publications focus on different types of "pure" Hotelling games. Núñez and Scarsini [23] demonstrated that in a model where all clients have strict preferences over a finite number of possible facility locations, pure equilibria exist when the number of facilities is sufficiently large. Fournier [12] analyzed approximate equilibria in pure

Hotelling games on a graph, proving that such equilibria always exist when the number of facilities is large. Peters et al. [25] iterated on Kohlberg's [16] model, and derived conditions under which subgame perfect equilibria exist for an even number of facility players.

1.3 Outline of the Thesis

The main goal of this thesis is to analyze equilibria for two-stage facility location games with unsplittable clients, with a primary focus on subgame perfect equilibria. In Chapter 2, we formally introduce the mathematical model and examine both stages of the two-stage game as independent, single-stage games. We discuss the concept of Nash equilibria for these single-stage games. Finally, we introduce the concept of subgame perfect equilibria within the context of the two-stage model and show that these subgame perfect equilibria may not exist in general.

In Chapter 3, we evaluate some of our model choices. We argue that certain instances may be interpreted as different representations of the same underlying problem, and show how to map instances in one representation to those in another. These different representations are useful for the construction of examples. The chapter serves two additional purposes: firstly, it allows the reader to become familiar with the model before getting to the more complex results. Secondly, it provides meaningful discussion on which properties of an instance are fundamental to the underlying problem, and which are only consequences of the way we choose to represent the problem. Although the results of this chapter are interesting, they do not directly pertain to the main topics of this thesis. Therefore, readers can choose to skip this chapter without any loss of comprehension in the subsequent chapters.

In Chapters 4 and 5, we examine sufficient conditions for the existence of subgame perfect equilibria. Chapter 4 contains most of the theoretical foundations necessary for these results. It also demonstrates that subgame perfect equilibria are guaranteed to exist for instances that admit a so-called balanced client equilibrium profile. Additionally, we present an algorithm that finds a specific client strategy profile, which is utilized in later results. In Chapter 5, we employ the developed theory to show that unweighted instances always admit subgame perfect equilibria. The chapter also includes counterexamples for some classes of problems which one may expect to always admit subgame perfect equilibria.

In Chapter 6, we investigate the efficiency of subgame perfect equilibria (when they exist) by establishing bounds on the price of anarchy and the price of stability.

Chapter 7 concerns the problem of deciding whether a given instance of the two-stage facility location game admits a subgame perfect equilibrium. We show that this decision problem is generally NP-hard.

In Chapter 8, we consider approximate (subgame perfect) equilibria, which are states out of which facility players may be able to improve, but so that no player can improve by a factor greater than a certain value α. We show the existence of instances that do not admit any ϕ-approximate equilibria, even when there are only two facility players, with ϕ denoting the golden ratio. We show that the problem of deciding whether a given instance of the two-stage facility location game admits an α-approximate subgame perfect equilibrium is NP-hard, for any given $\alpha < \phi$. Moreover, we show that a k-approximate equilibrium is guaranteed to exist, where k denotes the number of facility players. Additionally, we outline our efforts to narrow the gap between these two numbers for games with two facility players.

2 The Two-Stage Facility Location Game

2.1 Model and Notation

We consider a model similar to the one formulated by Krogmann et al. [17, 18], with some changes and additions to enable the description of the different client behavior. We consider the *two-stage facility location game* (2-FLG), where two types of players, k *facilities* and n *clients*, strategically interact on a given vertex-weighted directed host graph $H = (V, A, w)$ where $V = \{v_1, \dots, v_n\}$ is the vertex set, A is the arc set and $w : V \to \mathbb{Q}_+$ is the vertex weight function. Each vertex $v_i \in V$ represents a client with weight $w(v_i)$, while simultaneously functioning as a possible facility location. We denote the set of facility players as $\mathcal{F} = \{f_1, \dots, f_k\}$. The location function $\mathcal{U} : \mathcal{F} \to 2^V$ denotes the *location set* $\mathcal{U}(f_j)$ of each facility player $f_j \in \mathcal{F}$. The location set $\mathcal{U}(f_j) \subseteq V$ denotes the set of vertices that facility player f_j may choose as their location. An instance of 2-FLG is denoted using the tuple $(H, \mathcal{U}, k)$.

The two-stage facility location game consists of two stages, which are played sequentially by disjoint sets of players. Each of these stages is a simultaneous game. In the first stage, each facility player f_j selects a single location vertex $s_j \in \mathcal{U}(f_j)$ to place their facility, where multiple players can select the same vertex. We call the resulting vector of location vertices $\mathbf{s} = (s_1, \dots, s_k)$ the *facility placement profile* (FPP), and denote with $S \subseteq V^k$ the set of all possible facility placement profiles.

After all facility players have chosen a location, the second stage of the game starts. In this stage, each client may consider patronizing any facility in their shopping range $N(v_i) := \{v_i\} \cup \{z \mid (v_i, z) \in A\}$. Similarly, we define the *attraction range* $A_\mathbf{s}(f_j)$ of a facility f_j as the set of clients it can serve for the given facility placement profile. That is: $A_\mathbf{s}(f_j) := s_j \cup \{v_i \mid (v_i, s_j) \in A\}$. For a facility placement profile $\mathbf{s} \in S$, we denote with $V^{cov}(\mathbf{s}) \subseteq V$ the set of *covered clients*: clients with at least one facility in their shopping range. Every covered client $v_i \in V^{cov}(\mathbf{s})$ must patronize exactly one facility and strategically decides on a probability distribution representing the likelihoods of patronizing each facility in their shopping range. Let $\sigma(\mathbf{s}) : V \to [0, 1]^k$ denote the *client subgame (strategy) profile* resulting from the decisions of the clients. Here, $\sigma(\mathbf{s})_{i,j}$ denotes the probability that client v_i patronizes facility f_j. The term *subgame* refers to the fact that $\sigma(\mathbf{s})$ characterizes the behavior of the clients for a specific part of the two-stage game: the subgame related to the facility placement profile $\mathbf{s}$. We call $\sigma(\mathbf{s})$ feasible if all covered clients choose a probability distribution over the facilities in range, and all uncovered clients are guaranteed not to patronize any facility:

Definition 2.1. For some instance of 2-FLG and some facility placement profile $\mathbf{s} \in S$, a client subgame profile $\sigma(\mathbf{s})$ is feasible if for each client $v_i \in V$, we have:

$$
\begin{aligned}
\sigma(\mathbf{s})_{i,j} &= 0 \quad \forall\, f_j \notin N_\mathbf{s}(v_i) \\
\sigma(\mathbf{s})_{i,j} &\in [0, 1] \quad \forall\, f_j \in N_\mathbf{s}(v_i) \\
\sum_{f_j \in N_\mathbf{s}(v_i)} \sigma(\mathbf{s})_{i,j} &= \begin{cases} 1 \text{ if } N_\mathbf{s}(v_i) \neq \emptyset, \\ 0 \text{ else.} \end{cases}
\end{aligned}
$$

In sequential games, the *strategy* of a player characterizes their behavior for every possible situation. The strategy of a client player in the two-stage facility location game thus describes their behavior in response to every facility placement profile. A *strategy profile* characterizes the strategies of all players. As such, let $\sigma : S \times V \to [0, 1]^k$ denote the *client (strategy) profile*, which is characterized by a client subgame profile corresponding to *every* facility placement profile $\mathbf{s} \in S$. Naturally, we call a client profile σ feasible if $\sigma(\mathbf{s})$ is feasible for each $\mathbf{s} \in S$. Let Φ denote the set of feasible σ, and let $\Phi_{\mathbf{s}}$ denote the set of feasible client subgame profiles for each $\mathbf{s} \in S$. Then, $\Phi = (\Phi_{\mathbf{s}})_{\mathbf{s} \in S}$.

A *state* $(\mathbf{s}, \sigma)$ of 2-FLG is determined by a facility placement profile $\mathbf{s} \in S$ and a client profile $\sigma \in \Phi$. A state induces *expected (facility) loads* on all facilities. The expected load on facility f_j for state $(\mathbf{s}, \sigma)$ is $\ell_j(\mathbf{s}, \sigma) := \sum_{i=1}^{n} \sigma(\mathbf{s})_{i,j} \cdot w(v_i)$. Hence, $\ell_j(\mathbf{s}, \sigma)$ naturally models the total expected congestion at facility f_j, given client profile σ. Facility players aim to maximize their expected load, and will strategically place their facilities to achieve this. We further assume that the quality of the facilities, e.g., the waiting time, deteriorates with increasing congestion. Hence, for a client v_i, the expected facility load corresponds to the waiting time of v_i at the facility. However, the client v_i considers the expected facility loads, *conditional* on v_i patronizing that facility. This is generally different from the unconditional expected facility load considered by the facility player. Specifically, the client always incurs the full cost of their own weight, independent of their chosen strategy. The client, therefore, exclusively considers the load caused by other clients when deciding which facilities to patronize. To this end, we define the *v_i-excluded* facility load $\ell_{-i,j}(\mathbf{s}, \sigma)$ as the expected load on f_j caused by clients other than v_i:

$$\ell_{-i,j}(\mathbf{s}, \sigma) := \sum_{\substack{i'=1 \\ i' \neq i}}^{n} \sigma(\mathbf{s})_{i',j} \cdot w(v_{i'})$$

The expected load experienced by client v_i, referred to as the *cost* of client v_i, is then given by:

$$L_i(\mathbf{s}, \sigma) = w(v_i) + \sum_{j=1}^{k} \sigma(\mathbf{s})_{i,j} \cdot \ell_{-i,j}(\mathbf{s}, \sigma). \tag{1}$$

Each client aims to minimize their cost and picks their strategy accordingly. Note that the expected facility loads and the costs incurred by the clients for state $(\mathbf{s}, \sigma)$ are independent of the behavior of the clients for facility placements other than $\mathbf{s}$. The tuple $(\mathbf{s}, \sigma(\mathbf{s}))$ contains sufficient information to determine the payoffs for all players. As such, we sometimes denote a state $(\mathbf{s}, \sigma)$ as $(\mathbf{s}, \sigma(\mathbf{s}))$, when we are only interested in the payoffs resulting from FPP $\mathbf{s}$.

When a client patronizes a facility f_j with probability 1 (i.e., the client plays a pure strategy), the cost of this client is simply the expected load on f_j. When $\sigma(\mathbf{s})$ is a mixed client subgame profile, the cost of the client for $(\mathbf{s}, \sigma(\mathbf{s}))$ naturally equals the weighted average of the costs of the pure client subgame profiles in the support of $\sigma(\mathbf{s})$.

In this report, we often consider two specific classes/types of 2-FLG: *unweighted 2-FLG* and *unrestricted 2-FLG*.

Definition 2.2. An instance of 2-FLG is *unweighted* if all clients have equal weight.

We assume that all client weights are one for unweighted 2-FLG, as scaling all client weights by some nonzero number does not meaningfully change an instance.

Definition 2.3. An instance $(H, \mathcal{U}, k)$ of 2-FLG with $H = (V, A, w)$ is *unrestricted* if each facility player is allowed to select every vertex in V as their location. That is, if $\mathcal{U}(f_j) = V$ for each facility $f_j \in \mathcal{F}$.

We commonly denote unrestricted instances as (H, k) instead of $(H, \mathcal{U}, k)$.

2.2 Equilibria and Subgame Perfection

Both stages of the two-stage facility location game can be considered independent single-stage games, if the behavior of the players in the other stage is fixed. Every facility placement $\mathbf{s} \in S$ induces a (single stage) *client game*, where the client players seek to minimize their cost by strategically picking which facilities in their neighborhood to patronize. Similarly, every $\sigma \in \Phi$ induces a (single stage) *facility game*, where the facility players strategically pick locations to place their facilities.

The Client Game

Single-stage client games are an example of the well-studied *congestion games* (see Rosenthal [27]). Specifically, it is a *weighted singleton congestion game*. The client game is *weighted* as every client v_i has a weight $w(v_i)$, and it is *singleton* as every client ends up patronizing exactly one facility, unlike for more general non-singleton congestion games where players (clients) select subsets of the total set of available resources (facilities).

For the client game induced by $\mathbf{s} \in S$, the set of feasible strategy profiles is $\Phi_{\mathbf{s}}$. A strategy profile for the client game is thus a client subgame profile for the two-stage game. A client subgame profile $\sigma_{\mathbf{s}} \in \Phi_{\mathbf{s}}$ is a Nash equilibrium for this client game if no client can decrease their cost by unilaterally changing their strategy from $\sigma(\mathbf{s})$. To formalize this, let $\sigma(\mathbf{s})_{-i}$ denote the profile describing the strategies for $\sigma(\mathbf{s})$ of all clients except v_i. Then $\sigma(\mathbf{s})$ is a Nash equilibrium for the client game if for all clients $v_i \in V$, we have:

$$L_i(\mathbf{s}, (\sigma(\mathbf{s})_i, \sigma(\mathbf{s})_{-i})) \leq L_i(\mathbf{s}, (\sigma'(\mathbf{s})_i, \sigma(\mathbf{s})_{-i})) \text{ for all } (\sigma'(\mathbf{s})_i, \sigma(\mathbf{s})_{-i})) \in \Phi_{\mathbf{s}} \tag{2}$$

We call a Nash equilibrium for a client game a *client equilibria*, and denote with $\Omega_{\mathbf{s}} \subseteq \Phi_{\mathbf{s}}$ the set of all client equilibria for the client game induced by $\mathbf{s}$.

The definition in (2) is cumbersome to work with. Therefore, we provide a different characterization of client equilibria in Theorem 1, for which we require some new definitions. We say that a client v_i *considers* facility f_j for $\sigma(\mathbf{s})$ if v_i patronizes this f_j with a nonzero probability. Furthermore, we denote with $F_i(\mathbf{s}, \sigma) \subseteq N_s(v_i)$ the set of facilities considered by v_i for $(\mathbf{s}, \sigma)$.

Theorem 1. *A client subgame profile $\sigma(\mathbf{s}) \in \Phi_s$ is a Nash equilibrium for the client game induced by $\mathbf{s}$ if and only if for all clients $v_i \in V$, the set F_i exclusively contains facilities with minimal v_i-excluded load among the facilities in the attraction range of v_i.*

Proof. Assume by contradiction that there is a client v_i that considers a facility that does *not* have minimal v_i-excluded load. That is, there exist $f_j, f_x \in N_{\mathbf{s}}(v_i)$ such that $\sigma(\mathbf{s})_{i,j} > 0$ and $\ell_{-i,j}(\mathbf{s}, \sigma(\mathbf{s})) > \ell_{-i,x}(\mathbf{s}, \sigma(\mathbf{s}))$. Then v_i can improve by playing strategy $\sigma'(\mathbf{s})_i$ instead of $\sigma(\mathbf{s})_i$, where $\sigma'(\mathbf{s})_{i,j} = 0$, $\sigma'(\mathbf{s})_{i,x} = \sigma(\mathbf{s})_{i,x} + \sigma(\mathbf{s})_{i,j}$ and $\sigma'(\mathbf{s})_{i,y} = \sigma(\mathbf{s})_{i,y}$ for all $y \notin \{j, x\}$. As an improving move exists, $\sigma(\mathbf{s})$ cannot be a Nash equilibrium for the client game. □

Nash equilibria for the client game always exist. Even the existence of a pure Nash equilibrium is guaranteed, as shown by Rosenthal [27]. However, client equilibria are not generally unique. The following trivial example shows that the number of client equilibria may be infinite, and that each of these may induce different facility loads.

Theorem 2. *The number of client equilibria of a client game is not generally bounded, and different client equilibria for the same client game may induce different facility loads.*

Proof. Consider the instance of 2-FLG with host graph $H = (\{v_1\}, \emptyset, (w(v_1) = 1))$, $k = 2$ and facility placement profile $\mathbf{s} = (v_1, v_1) \in S$. Then, for every $\gamma \in [0, 1]$, the client subgame profile $\sigma(\mathbf{s})$ with $\sigma(\mathbf{s})_1 = (\gamma, 1-\gamma)$ is a client equilibrium. Furthermore, the expected facility loads are $\ell_1(\mathbf{s}, \sigma(\mathbf{s})) = \gamma$ and $\ell_2(\mathbf{s}, \sigma(\mathbf{s})) = 1 - \gamma$. These loads are thus different for every value of γ. The number of such client equilibria is unbounded. □

Theorem 2 shows that even if we assume that the clients' behavior always results in a client equilibrium, we do not generally know the facility loads resulting from a facility placement profile $\mathbf{s}$ when $\sigma(\mathbf{s})$ is not given.

The Facility Game

In the single-stage facility game, the facility players each choose a location to place their facility and aim to maximize their expected facility load. A (single-stage) facility game is induced by a client profile $\sigma \in \Phi$. Thus, the client behavior resulting from each facility placement profile is known. That is, the facility players know the loads resulting from each facility placement profile $\mathbf{s} \in S$.

A facility placement profile $\mathbf{s} \in S$ is a Nash equilibrium (*facility equilibrium*) for the single-stage facility game induced by σ if no facility player can improve their load by unilaterally changing their strategy. Formally, $\mathbf{s} \in S$ is a facility equilibrium if for all facilities $f_j \in \mathcal{F}$:

$$\ell_j(\mathbf{s}, \sigma) \geq \ell_j(\mathbf{s}', \sigma) \quad \forall \mathbf{s}' \in S \text{ s.t. } s'_i = s_i \text{ for } i \neq j.$$

The facility game is a normal-form game with a finite number of pure strategies and players. Nash [22] showed that mixed Nash equilibria always exist for such games. However, we are only interested in pure facility equilibria, which do not generally exist. Analyzing under which conditions pure facility equilibria exist is the primary focus of this thesis.

Subgame Perfect Equilibria

Recall that client games may admit multiple client equilibria, and that these may induce different facility loads. It is infeasible to determine the facility loads knowing only a facility placement profile. This complicates our analysis of equilibria with respect to the facility players, as a procedure similar to that used by Krogmann et al. in [17] and [18] is infeasible. We must consider the behavior of both types of players simultaneously. This leads us to analyze the subgame perfect equilibria (SPE) of the two-stage game.

In line with the literature [24], we call a state $(\mathbf{s}, \sigma) \in S \times \Phi$ a subgame perfect equilibrium for the two-stage facility location game if and only if the following two conditions hold:

1. $\mathbf{s}$ is a facility equilibrium for the facility game induced by σ.
2. For all $\mathbf{s}' \in S$, $\sigma(\mathbf{s}')$ is a client equilibrium for the client game induced by $\mathbf{s}'$.

We call client profiles $\sigma \in \Phi$ that satisfy the second condition *client equilibrium profiles* and denote the set of all client equilibrium profiles as Ω, with $\Omega \subseteq \Phi$.

Since single-stage games (both client and facility games) always admit mixed Nash equilibria, two-stage games always admit a *mixed* SPE. However, we only consider subgame perfect equilibria where the facility players play pure strategies. Client players are allowed to randomize their behavior, however. For the rest of this report, all mentions of subgame perfect equilibria implicitly refer to subgame perfect equilibria that are pure with respect to the facility strategies. Lemma 1 shows that the existence of such an SPE is not guaranteed, not even for unrestricted 2-FLG.

Lemma 1. *Instances of the two-stage facility location game might not admit any subgame perfect equilibria. This is even true for unrestricted 2-FLG.*

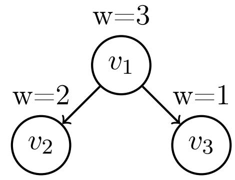


Figure 1: Host graph H

Proof. Consider instance $(H, 2)$ of unrestricted 2-FLG where H is the host graph shown in Figure 1. We claim that no subgame perfect equilibrium exists. To show this, we determine the set of client equilibrium profiles Ω by finding the set of client equilibria for each induced client game. An exhaustive list of all client equilibria is given in Table 1.

Table 1: Possible client equilibria for every facility placement profile

$\mathbf{s} = (s_1, s_2)$	$\Omega_{\mathbf{s}}$	induced loads (ℓ_1, ℓ_2)
(1, 1)	$\{((\gamma, 1-\gamma), (0,0), (0,0)) \mid \gamma \in [0,1]\}$	$(\gamma, 3-\gamma), \gamma \in [0,3]$
(1, 2)	$\{((1,0), (0,1), (0,0))\}$	(3, 2)
(1, 3)	$\{((1,0), (0,0), (0,1))\}$	(3, 1)
(2, 1)	$\{((0,1), (1,0), (0,0))\}$	(2, 3)
(2, 2)	$\{((0,1), (1,0), (0,0)),$ $((1,0), (0,1), (0,0)),$ $((0.5,0.5), (0.5,0.5), (0,0))\}$	(2, 3) (3, 2) (2.5, 2.5)
(2, 3)	$\{((0,1), (1,0), (0,1))\}$	(2, 4)
(3, 1)	$\{((0,1), (0,0), (1,0))\}$	(1, 3)
(3, 2)	$\{((1,0), (0,1), (1,0))\}$	(4, 2)
(3, 3)	$\{((0,1), (0,0), (1,0)),$ $((1,0), (0,0), (0,1)),$ $((0.5,0.5), (0,0), (0.5,0.5))\}$	(1, 3) (3, 1) (2, 2)

Now consider the facility game induced by some $\sigma \in \Omega$. This game can be written as a bimatrix game with the following payoff matrices:

	v_1	v_2	v_3
v_1	$a_{1,1}, b_{1,1}$	$3, 2$	$3, 1$
v_2	$2, 3$	$a_{2,2}, b_{2,2}$	$2, 4$
v_3	$1, 3$	$4, 2$	$a_{3,3}, b_{3,3}$

Here, the diagonal entries depend on the choice of σ. We do not need to know the payoffs on the diagonal to conclude that none of the off-diagonal strategy profiles are Nash equilibria. For each of these six strategy profiles, some player can improve by moving to another off-diagonal strategy profile. Next, consider the diagonal strategy profiles:

- $\mathbf{s} = (v_1, v_1)$: for this to be a Nash equilibrium, we require: $a_{1,1} \geq 2, b_{1,1} \geq 2$. This cannot hold as $a_{1,1} + b_{1,1} = 3$ for all $\sigma \in \Omega$.
- $\mathbf{s} = (v_2, v_2)$: for this to be a Nash equilibrium, we require: $a_{2,2} \geq 4, b_{2,2} \geq 4$. No $\sigma \in \Omega$ satisfies this.

- $\mathbf{s} = (v_3, v_3)$: for this to be a Nash equilibrium, we require: $a_{3,3} \geq 3, b_{2,2} \geq 3$. No $\sigma \in \Omega$ satisfies this.

Thus, there is no $\sigma \in \Omega$ for which the induced facility game admits a pure Nash equilibrium. We conclude that the instance does not admit a subgame perfect equilibrium. □

In Chapter 4, we start the search for classes of 2-FLG that are guaranteed to admit subgame perfect equilibria. Before that, however, we take a detour to gain more insight into our model choices and investigate which parts of the representation of a problem instance are fundamental. The reader may skip this chapter to get to the main results of the thesis without risking the subsequent loss of comprehension.

3 Different Problem Representations

This chapter introduces two classes of 2-FLG: *bipartite* and *uniformly restricted* 2-FLG. We argue that for any instance of bipartite, uniformly restricted, or unrestricted 2-FLG, there are instances of the other two classes that are equivalent for all practical purposes. These classes of instances may thus be interpreted as different representations of the same underlying problems. We refine this notion of equivalence by introducing the concept of *essentially equivalent* instances. However, we refrain from formally stating the consequences of this definition, as this would necessitate excessively cumbersome notation and handling several technicalities.

One goal of this chapter is to show that all results for *unrestricted* 2-FLG also hold for the other two classes. Additionally, unrestricted instances are particularly well-suited to visualizations, as their host graphs are comparatively small, and the lack of location sets makes them easier to understand at a glance. However, the property that the facility locations are represented with the same vertices as the clients, complicates the construction of (counter)examples in proofs. Bipartite 2-FLG avoids this problem. We get the best of both worlds by constructing bipartite examples and mapping them to *essentially equivalent* unrestricted instances. We introduce the new classes first and define the concept of *essential equivalence* later.

Definition 3.1. We call an instance $(H,\mathcal{U},k)$ of 2-FLG with $H = (V, A, w)$, *uniformly restricted* if the location set of every facility is the same. That is, if there is a set $U \subseteq V$ such that $\mathcal{U}(f_j) = U$ for all facilities $f_j \in \mathcal{F}$.

Note that every instance of unrestricted 2-FLG is uniformly restricted (with $U = V$).

Next, we define *Bipartite* 2-FLG. One may intuitively interpret *bipartite* 2-FLG as a version of unrestricted 2-FLG where each vertex of the host graph represents either a client or a possible facility location, but not both. The class derives its name from its bipartite host graphs, where all edges are incident to one "client vertex" and one "location vertex". Of course, this interpretation is inconsistent with our definition of 2-FLG, where every vertex represents a client. We solve this using clients with zero weight. Such clients cannot influence the behavior of other players, and can thus mostly be ignored.

Definition 3.2. We call an instance $(H,\mathcal{U},k)$ of 2-FLG with $H = (V, A, w)$ *bipartite* if the location set $U \subseteq V$ of every facility is the same, all clients in U have zero weight, and all edges in A are of the form (v_i, v_j) with $v_i \in V \setminus U$ and v_i in U.

Note that every instance of bipartite 2-FLG is uniformly restricted. We often denote instances of bipartite or uniformly restricted 2-FLG with a tuple (H, U, k) instead of $(H,\mathcal{U}, k)$, where U denotes the location set of all facilities.

3.1 Defining Equivalence

Before defining *essential equivalence*, we provide some arguments for why this definition implies "equivalence for all practical purposes". We argue that the location of a facility matters only insofar as it affects the attraction range of the facility. That is, we implicitly assume that clients can or do not distinguish between two distinct facility placement profiles if each client has identical shopping ranges for both profiles. As an example, consider instance $(H, 2)$ of unrestricted 2-FLG with host graph H as shown in Figure 2. We argue that all four facility placement profiles in S induce the same client game since both clients can reach both facilities for each FPP. Additionally, if the client subgame profile corresponding to a set of facility placement profiles is the same, then these FPPs are equivalent from the perspective of the facilities. We argue that all such facility placement profiles are equivalent for all practical purposes and that two location vertices corresponding to identical attraction ranges are therefore redundant (for the purpose of facility placement). Therefore, refining the location set $\mathcal{U}(f_j)$ of a facility f_j by removing such redundant vertices does not change the instance.

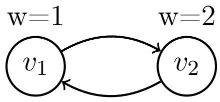


Figure 2: Host graph H

Additionally, we argue that client subgame profiles are only determined by the choices of clients with nonzero weight. Clients with zero weight cannot affect the behavior of any other players (both facilities and other clients). Lastly, we argue that clients that are not covered for any facility placement profile in S do not matter, since such clients cannot affect the behavior of other players. We argue that adding or removing such clients does not meaningfully change an instance. These arguments bring us to the following definition of *essential equivalence*.

Definition 3.3. We call two client games *essentially equivalent* if the sets of covered clients with nonzero weight are identical (including their weights), and the shopping ranges of all of these clients are identical for both games.

Definition 3.4. We call two facility placement profiles *essentially equivalent* if they induce *essentially equivalent* client games.

Definition 3.5. We call two instances of 2-FLG *essentially equivalent* if the following conditions hold:

1. Both instances have the same set of facility players
2. Both instances have the same set of client players (including weights), excluding clients with zero weight and clients that are uncovered for all facility placement profiles.
3. For every facility placement profile in either instance, there is an essentially equivalent facility placement profile in the other.

This definition ensures that any two *essentially equivalent* instances are identical for all practical purposes. Intuitively, a state $(\mathbf{s}, \sigma)$ of one instance can be mapped to an equivalent one in the other. However, we would need to define equivalence between client subgame profiles, client profiles, and facility games to formally prove this property. Seeing as this is not the primary focus of this report, we refrain from doing so.

Note that essential equivalence is transitive. That is, if an instance (H, U, k) is essentially equivalent to both (H', U', k) and (H^*, U^*, k), then (H', U', k) and (H^*, U^*, k) are also essentially equivalent to each other.

3.2 Essentially Equivalent Instances of Bipartite 2-FLG

To prove the main result of this chapter, we need to develop some theory on augmenting instances of bipartite 2-FLG to ones that are essentially equivalent. Lemma 2 shows that applying certain operations to bipartite instances results in new bipartite instances that are essentially equivalent to the original.

Lemma 2. *Augmenting an instance (H, U, k) of bipartite 2-FLG with $H = (V, A, w)$ by performing any of the following operations, results in a new instance that is essentially equivalent to the original instance.*

1. *Adding or removing vertices with degree 0 from $V \setminus U$.*

2. *Adding or removing edges (v_i, u_j) with $v_i \in V \setminus U, u_j \in U$ and $w(v_i) = 0$.*

3. *Removing a redundant location vertex. That is, removing a vertex $u_i \in U$ from the host graph if there exists another location vertex $u_j \in U, u_j \neq u_i$ with an identical neighborhood $N(u_j)$ to that of u_i.*

4. *Adding a redundant location vertex. That is, adding a vertex u_j with $w(u_j) = 0$ to V and U, and edges $\{(v, u_j) \mid v \in N(u_i)\}$ for some $u_i \in U$ to A.*

Proof. 1. These vertices correspond to clients that are not covered for any facility placement profile. The existence of these clients does not affect the essential equivalence of instances. 2. These edges only affect the shopping ranges of clients with zero weight. By definition, these shopping ranges do not affect the essential equivalence of instances. 3. The removed vertex u_i corresponds to a client with zero weight, which does not affect the essential equivalence of instances. Furthermore, any FPP in the original instance which made use of the removed vertex u_i is essentially equivalent to an FPP in the new instance where every occurrence of u_i is replaced by u_j. 4. The added vertex u_i corresponds to a client with zero weight, which does not affect the essential equivalence of instances. Any FPP in the new instance which makes use of the added vertex u_j is essentially equivalent to an FPP in the original instance where every occurrence of u_j is replaced by u_i. □

We use Lemma 2 and the transitivity of essential equivalence to show that every instance of bipartite 2-FLG is essentially equivalent to a *non-degenerate* instance of bipartite 2-FLG.

Definition 3.6. We call an instance (H, U, k) of bipartite 2-FLG *non-degenerate* if every client $v_i \in V \setminus U$ has nonzero weight and degree, and every location vertex $u \in U$ has a unique neighborhood. An instance of bipartite 2-FLG that is not non-degenerate is called *degenerate.*

Lemma 3. *Every degenerate instance of bipartite 2-FLG is equivalent to a non-degenerate instance of the same problem.*

Proof. Follows immediately from Lemma 2. □

An application of this concept of non-degeneracy is finding a contradiction on the graph structure of certain instances. It permits one to consider only non-degenerate instances when proving that instances with some property cannot exist, as long as this property is shared between essentially equivalent instances. If one shows that there are no non-degenerate instances with this property, then there are no bipartite instances with this property in general (by Lemma 3). We apply this idea in Chapter 8.

3.3 Equivalence of Different Classes

We now show how to map bipartite, unrestricted, and uniformly restricted instances to essentially equivalent ones of the other classes. First, we show how to map instances of uniformly restricted instances to bipartite ones. The general idea is to add a new vertex with weight zero to the graph for every vertex in the location set of the original instance. Then, we add edges so that each new vertex can be reached by all clients that could reach the corresponding original vertex in the original instance. Figure 3 shows an example of this mapping, where we represent U and U' using square vertices. We omit the weight label for vertices in the location set of bipartite instances, as these are zero by definition.

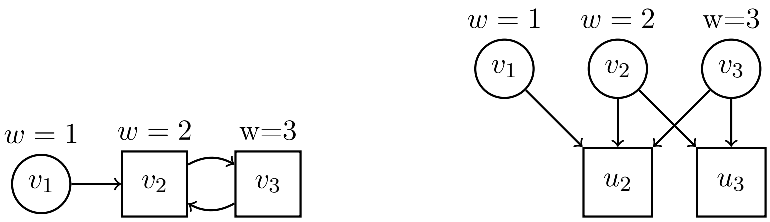


Figure 3: Equivalent instances of uniformly restricted and bipartite 2-FLG

Lemma 4. *For every instance of uniformly restricted 2-FLG, there is an essentially equivalent instance of bipartite 2-FLG.*

Proof. Let (H, U, k) denote an instance of uniformly restricted 2-FLG, with $H := (V, A, w)$ and $V := \{v_1, v_2, \ldots, v_n\}$. We construct an instance (H', U', k) of bipartite 2-FLG that is essentially equivalent to (H, U, k). We define $U' := \{u_i \mid v_i \in U\}$, where all vertices in U' are new. Next, we define host graph $H' := (V', A', w')$ with:

$$V' = V \cup U'$$
$$A' = \big\{(v_i, u_j) \mid v_i \in V, u_j \in U', (v_i, v_j) \in A\big\} \cup \big\{(v_i, u_i) \mid i = 1, 2, \ldots, |U|\big\}$$
$$w'(x) = \begin{cases} w(x) \text{ if } x \in V \\ 0 \text{ if } x \in U'. \end{cases}$$

We consider the same set of facility players as in (H, U, k). By construction, (H', U', k) is an instance of bipartite 2-FLG. We claim that (H, U, k) and (H', U', k) are essentially equivalent. To see this, note that the sets of facility players and the sets of client players with nonzero weights are the same for both instances. Furthermore, the attraction range (in terms of clients with nonzero weight) of a facility located on some $v_i \in V$ in (H, U, k) is the same as for a facility located on $u_i \in U'$ in (H', U', k). As such, any facility placement profile for either instance can be mapped to an essentially equivalent one for the other. We conclude that the two instances are essentially equivalent. □

Next, we show how to map instances of bipartite 2-FLG to unrestricted ones. The general idea is to first construct a new bipartite instance of 2-FLG, essentially equivalent to the original, that admits a perfect matching of the graph. Then, we construct the unrestricted instance by merging every pair of matched vertices in a specific way. We provide an example of the host graphs at different points in this process in Figure 4. For the bipartite instances, we indicate a maximum matching by bold edges. Be mindful that the procedure relabels the vertices of U.

Lemma 5. *For every instance of bipartite 2-FLG, there is an essentially equivalent instance of unrestricted 2-FLG.*

Proof. Let (H, U, k) denote an instance of bipartite 2-FLG with host graph $H := (V, A, w)$. We first construct another instance (H', U', k) of bipartite 2-FLG that is essentially equivalent to (H, U, k). In particular, we construct an instance that admits a perfect matching of the graph. We construct this instance by applying the operations described in Lemma 2.

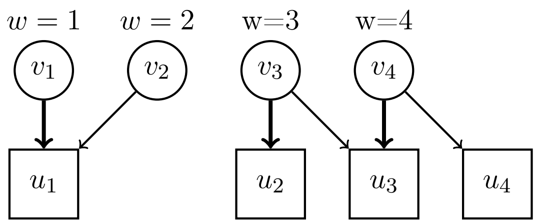
First, we find a maximal matching M in (H, U, k). Then, for every vertex $u_i \in U$ that is not matched by M, we augment H by adding a vertex v_i with weight zero to V and an edge (v_i, u_i) to A. Let M' denote the matching obtained from M by adding all newly added edges. By construction, M' is a maximum matching of the augmented host graph $H' = (V', A', w')$. Next, we consider vertices in $V' \setminus U$ that are unmatched by M'. For each such (client) vertex v_j we consider some vertex $u_x \in U$ that is adjacent to v_j. We "duplicate" this vertex u_x and match it to v_j. That is, we augment H' by adding a new vertex u_y to U and adding edges $\{(v, u_y) \mid v \in N(u_x)\}$ to A'. Additionally, we augment M' by adding edge (v_j, u_y).

Each augmentation of H is an application of Lemma 2. As such, the augmented instance (H', U', k) is essentially equivalent to (H, U, k). Furthermore, M' is a perfect matching of the graph for this instance. For ease of notation, we now relabel the vertices of U' such that $M' := \{(v_i, u_i) \mid i = 1, 2, \ldots, |U|\}$. Since all these vertices in U have weight zero, the instance obtained by this relabeling is essentially equivalent to the original.

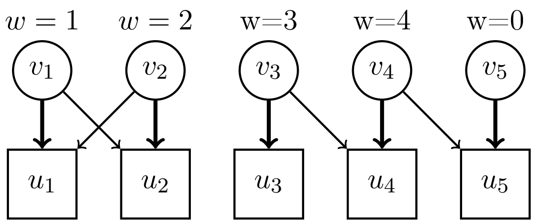
We use the perfect matching to construct an instance (H^*, k) of unrestricted 2-FLG that is essentially equivalent to (H', U', k). We define $H^* := (V^*, A^*, w^*)$ as follows:

$$\begin{aligned} V^* &= V' \setminus U' \\ A^* &= \{(v_i, v_j) \mid (v_i, u_j) \in A', i \neq j\} \\ w^*(v) &= w'(v) \quad \text{for all } v \in V^* \end{aligned}$$

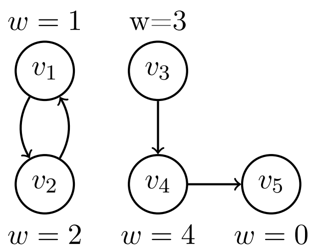
Let (H^*, k) use the same facility set as (H', U', k). By construction, the sets of facilities and the sets of clients with nonzero weights are the same for both instances. Furthermore, the attraction range of a facility located on any $v_i \in V^*$ in (H^*, k) is the same as for a facility located on u_i in (H', U', k). As such, the two instances are essentially equivalent. We conclude by transitivity that (H^*, k) and (H, U, k) are essentially equivalent. □



(a) Original host graph H, maximal matching in **bold**



(b) Augmented host graph H', perfect matching in **bold**



(c) Host graph H^* for unrestricted 2-FLG

Figure 4: Essentially equivalent instances for bipartite an unrestricted 2-FLG

We now have all necessary results for the main theorem of this chapter.

Theorem 3. *For every instance of either bipartite, uniformly restricted or unrestricted 2-FLG, there are essentially equivalent instances in both other classes.*

Proof. Recall that every instance of unrestricted 2-FLG is also uniformly restricted, and note that every instance of unrestricted 2-FLG is essentially equivalent to itself. For every instance of unrestricted 2-FLG, there is thus an essentially equivalent instance of uniformly restricted 2-FLG. Combining this with Lemmas 4 and 5 completes the proof. □

Theorem 3 shows the modeling power of unrestricted 2-FLG: every problem that can be modeled as a bipartite or uniformly restricted instance can also be modeled as an unrestricted instance. We find this especially striking for uniformly restricted instances, as one would assume that the addition of location sets increases modeling power. Lastly, we note that the results of this chapter do *not* hold for unweighted 2-FLG, as the proofs make heavy use of clients with zero weight.

4 Balancedness and the Class Set

In this chapter, we define the concept of *balanced* client (subgame) profiles. The theory we develop around this type of client behavior provides us with the tools necessary to show that two different conditions are sufficient for the existence of subgame perfect equilibria. The first of these conditions, the existence of a *balanced* client equilibrium profile, is shown to be sufficient in this chapter. The second condition, that of equal client weights, is discussed in Chapter 5.

4.1 Balanced Client Profiles

We define a specific type of client behavior, which we call *balanced*. Intuitively, one may consider a balanced client subgame profile $\sigma(\mathbf{s})$ as the fairest possible assignment of the clients to the facilities for the client game induced by $\mathbf{s}$. This notion of fairness is sometimes referred to as the *egalitarian rule* or *max-min fairness* [28]. Formally, we define *balancedness* as follows:

Definition 4.1. Consider some instance of 2-FLG and facility placement profile $\mathbf{s} \in S$. We call a client subgame profile $\sigma(\mathbf{s})$ *balanced* if for any pair of (distinct) facilities $f_i, f_j \in \mathcal{F}$, one of the following holds:

1. $\ell_i(\mathbf{s}, \sigma(\mathbf{s})) = \ell_j(\mathbf{s}, \sigma(\mathbf{s}))$
2. $\ell_i(\mathbf{s}, \sigma(\mathbf{s})) > \ell_j(\mathbf{s}, \sigma(\mathbf{s}))$ and none of the clients in $A_{\mathbf{s}}(f_i) \cap A_{\mathbf{s}}(f_j)$ consider f_i.
3. $\ell_j(\mathbf{s}, \sigma(\mathbf{s})) > \ell_i(\mathbf{s}, \sigma(\mathbf{s}))$ and none of the clients in $A_{\mathbf{s}}(f_i) \cap A_{\mathbf{s}}(f_j)$ consider f_j.

Definition 4.2. For the two-stage facility placement game, we call a client profile $\sigma \in \Phi$ *balanced* if for every FPP $\mathbf{s} \in S$, $\sigma(\mathbf{s})$ is a balanced client subgame profile.

Mind that a client subgame profile being balanced, and it being a client equilibrium, are largely unrelated. Balancedness relates to how the behavior of the clients affects the facilities, while being a client equilibrium relates to how the behavior of the clients affects the clients themselves. Some balanced client subgames profiles are not client equilibria, and some client equilibria are not balanced.

As an example, consider instance $(H, 3)$ of unrestricted, unweighted 2-FLG, where H is the host graph shown in Figure 5, and FPP $\mathbf{s} = (v_1, v_2, v_3)$.

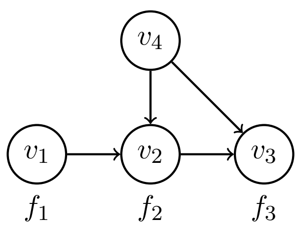


Figure 5: Host graph H with facility placement profile $\mathbf{s} = (v_1, v_2, v_3)$

There are (infinitely) many balanced client subgame profiles for the client game induced by $\mathbf{s}$. Two such profiles are:

- $\sigma^1(\mathbf{s}) = \Big((1,0,0),(0,\frac{1}{2},\frac{1}{2}),(0,0,1),(0,1,0)\Big)$
- $\sigma^2(\mathbf{s}) = \Big((1,0,0),(0,\frac{3}{4},\frac{1}{4}),(0,0,1),(0,\frac{3}{4},\frac{1}{4})\Big)$

There are a few things to note. First, the induced facility loads are the same for both of these client subgame profiles (1, 1.5 and 1.5 for f_1, f_2 and f_3 respectively). This is no coincidence; we show in Lemma 6 that all balanced client subgame profiles induce identical facility loads. Next, note that a balanced client subgame profile can be a client equilibrium, but is not necessarily one; $\sigma^1(\mathbf{s})$ is a client equilibrium since no client can improve, but $\sigma^2(\mathbf{s})$ is not, since both v_2 and v_4 can improve.

We mentioned earlier that balanced client subgame profiles may be seen as the fairest possible assignment of the clients over the facilities. We formalize this notion by introducing the *vector of sorted facility loads* ℓ_{sort}.

Definition 4.3. Consider some instance of 2-FLG and some state $(\mathbf{s},\sigma) \in S \times \Phi$. Then we denote with $\ell_{sort}(\mathbf{s},\sigma)$ the vector of *sorted facility loads*. That is, the sequence of induced facility loads for state $(\mathbf{s},\sigma)$ in increasing order.

Theorem 4. *For the client game induced by s, a client subgame profile $\sigma(s)$ is balanced if and only if it maximizes ℓ_{sort} lexicographically.*

Proof. We consider an instance $(H,\mathcal{U},k)$ of 2-FLG with $H = (V,A,w)$ and some facility placement profile $\mathbf{s} \in S$. Let $\sigma(\mathbf{s})$ denote a client subgame profile for the client game induced by $\mathbf{s}$.

We assume that $\sigma(\mathbf{s})$ is balanced, and show that it maximizes ℓ_{sort} lexicographically. Assume by contradiction that there is a some other FPP $\sigma'(\mathbf{s}) \in \Phi_{\mathbf{s}}$ such that $\ell_{sort}(\mathbf{s},\sigma'(\mathbf{s}))$ is lexicographically greater than $\ell_{sort}(\mathbf{s},\sigma(\mathbf{s}))$. Let f_i denote a facility with minimal load for state $\sigma(\mathbf{s})$ among those facilities whose load for $\sigma(\mathbf{s})$ is strictly smaller than for $\sigma'(\mathbf{s})$. Let $\mathcal{F}' := \{f_j \in \mathcal{F} \mid \ \ell_j(\mathbf{s},\sigma) \leq \ell_i(\mathbf{s},\sigma)\}$ denote the set of facilities that had smaller or equal load than f_i for $\sigma(\mathbf{s})$. As ℓ_{sort} increases lexicographically when moving from $\sigma(\mathbf{s})$ to $\sigma'(\mathbf{s})$, we find that $\ell_x(\mathbf{s},\sigma) \leq \ell_x(\mathbf{s},\sigma')$, for all $f_x \in \mathcal{F}'$. Thus, the sum of the loads of the facilities in $\mathcal{F}'$ has strictly increased. It follows that some client in $A_{\mathbf{s}}(\mathcal{F}')$ must have shifted probability density from a facility in $\mathcal{F}\backslash\mathcal{F}'$ to one in $\mathcal{F}'$. But by the balancedness of $\sigma(\mathbf{s})$, clients in $A_s(\mathcal{F}')$ could not have patronized any facility in $\mathcal{F} \setminus \mathcal{F}'$ for $\sigma(\mathbf{s})$. We conclude that $\sigma'(\mathbf{s})$ cannot exist.

Now assume that $\sigma(\mathbf{s})$ is not balanced. By definition, there is a client $v_i \in V$ and facilities $f_j, f_x \in A_{\mathbf{s}}(v_i)$ with $\ell_j(\mathbf{s},\sigma(\mathbf{s})) < \ell_x(\mathbf{s},\sigma(\mathbf{s}))$, but v_i considers f_j for $\sigma(\mathbf{s})$. Consider client subgame profile $\sigma'(\mathbf{s})$ identical to $\sigma(\mathbf{s})$ except that the probability that v_i patronizes f_j is ϵ higher, and the probability that they patronize f_x is ϵ lower. For $\epsilon > 0$ sufficiently small, we find:

$$\min(\ell_j(\mathbf{s},\sigma'(\mathbf{s})),\ell_x(\mathbf{s},\sigma'(\mathbf{s}))) > \ell_j(\mathbf{s},\sigma(\mathbf{s})). \tag{3}$$

Since the load on all facilities except f_j and f_x is the same for $\sigma(\mathbf{s})$ and $\sigma(\mathbf{s})$, we find that $\ell_{sort}(\mathbf{s},\sigma'(\mathbf{s})) > \ell_{sort}(\mathbf{s},\sigma(\mathbf{s}))$ lexicographically. Thus, $\sigma(\mathbf{s})$ does not maximize ℓ_{sort} lexicographically. □

One may interpret Theorem 4 as follows: a balanced client equilibrium maximizes the minimum facility load of any facility as the highest priority. Among such profiles, it maximizes the second lowest facility load, and so on. Theorem 4 trivially implies that the vector of sorted facility loads is identical for every balanced client subgame profile. Moreover, all induced facility loads are identical for a pair of balanced client subgame profiles, as proven in Lemma 6.

Lemma 6. *Consider some instance of 2-FLG and some* $\mathbf{s} \in S$*. Each balanced client subgame profile for the client game induced by* $\mathbf{s}$ *induces the same facility loads.*

Proof. Assume by contradiction that there is a pair of balanced client subgame profiles $\sigma(\mathbf{s}), \sigma'(\mathbf{s})$ that *do* induce different facility loads. We denote with F the set of facilities that have different loads for $\sigma(\mathbf{s})$ and $\sigma'(\mathbf{s})$. By assumption, F is nonempty. Let f_j denote a facility in F that has minimum load for $\sigma(\mathbf{s})$, and define $\alpha := \ell_j(\mathbf{s}, \sigma(\mathbf{s}))$. We claim that all facilities in F have a load of at least α for both profiles. To see this, note that since the vector of sorted loads is the same for both profiles (Lemma 4), F contains the same number of facilities whose load for $\sigma'(\mathbf{s})$ is smaller than α as facilities whose load for $\sigma(\mathbf{s})$ is smaller than α. Since the latter number is zero by the choice of f_j, so is the former.

Now consider client subgame profile $\sigma^*(\mathbf{s})$, which we define as the "average" of $\sigma(\mathbf{s})$ and $\sigma'(\mathbf{s})$. That is, $\sigma^*(\mathbf{s})_{i,j} = \frac{1}{2}(\sigma(\mathbf{s})_{i,j} + \sigma'(\mathbf{s})_{i,j})$ for all i, j. Note that the number of facilities whose load for $\sigma^*(\mathbf{s})$ is strictly less than α is the same as for both original profiles. Furthermore, facilities with load exactly α for $\sigma^*(\mathbf{s})$ have load exactly α for both original profiles as well. Since f_j has load α for $\sigma(\mathbf{s})$, but not for $\sigma'(\mathbf{s})$, the number of facilities with load α for $\sigma^*(\mathbf{s})$ is strictly smaller than for $\sigma(\mathbf{s})$. It follows that $\ell_{sort}(\mathbf{s}, \sigma^*(\mathbf{s})) > \ell_{sort}(\mathbf{s}, \sigma(\mathbf{s}))$ lexicographically. Applying Lemma 4 shows that $\sigma(\mathbf{s})$ is not balanced; a contradiction. □

Theorem 4 has another important consequence: it implies the existence of a balanced client subgame profile for every client game.

Theorem 5. *Consider some instance of 2-FLG. For every* $\mathbf{s} \in S$*, the client game induced by* $\mathbf{s}$ *admits a balanced client subgame profile.*

Proof. Every client game admits a feasible client subgame profile. As such, there is always one that maximizes the vector of sorted facility loads lexicographically. By Theorem 4, this client subgame profile is balanced. We conclude that a balanced client subgame profile exists for every client game. □

The existence guarantee of balanced profiles is what makes them such a powerful tool. We later provide an algorithm for finding a balanced client subgame profile in any client game. Next, we discuss the implications of the existence of a client subgame profile that is not only balanced, but also a client equilibrium. Such balanced client equilibria possess several useful properties.

Lemma 7. *Let* $\sigma \in \Omega$ *be a balanced client equilibrium profile, and let* $F_i(\mathbf{s}, \sigma)$ *denote the set of facilities considered by client* v_i *for state* $(\mathbf{s}, \sigma)$*. Then for any* $\mathbf{s} \in S$ *and* $v_i \in V$*, client* v_i *patronizes each facility in* $F_i(\mathbf{s}, \sigma)$ *with equal probability in state* $(\mathbf{s}, \sigma)$*.*

Proof. As $\sigma(\mathbf{s})$ is balanced: every facility in $F_i(\mathbf{s}, \sigma)$ has equal load. If not, we can select a pair of facilities with different loads and contradict the definition of a balanced client subgame profile. Furthermore, as $\sigma(\mathbf{s})$ is a client equilibrium, the v_i-excluded loads of all these facilities are equal. It follows that the contribution of v_i to each induced facility load is equal. That is, v_i patronizes each facility with the same probability. □

Corollary 5.1. *For any instance of 2-FLG, the number of balanced equilibria is finite.*

Proof. Let (H, k) denote some instance of 2-FLG. We show that for every $\mathbf{s} \in S$, the number of balanced client equilibria is finite. By Lemma 7, every balanced client equilibrium can be characterized by a *unique* tuple $(F_1(\mathbf{s}, \sigma(\mathbf{s})), F_2(\mathbf{s}, \sigma(\mathbf{s})), \ldots, F_n(\mathbf{s}, \sigma(\mathbf{s})))$. The set of possible values attainable by $F_i(\mathbf{s}, \sigma(\mathbf{s})$ is bounded above by the size of the power set of $N_{\mathbf{s}}(v_i)$, which is finite. As such, there is only a finite number of balanced client equilibria $\sigma(\mathbf{s})$ for each $s \in S$, and thus only a finite number of balanced client equilibria $\sigma \in \Omega$. $\square$

We now show that the existence of a balanced client equilibrium profile is a sufficient condition for the existence of a subgame perfect equilibrium. To prove this, we show that the facility game induced by any balanced client profile (not just equilibria) always admits a pure Nash equilibrium. To this end, we prove that the vector of sorted facility loads is a lexicographical potential function for facility games induced by balanced client profiles. A lexicographical potential function is a function that increases lexicographically whenever one of the facility players performs an improving move. The existence of a (lexicographical) potential function proves that any sequence of improving moves leads to a pure Nash equilibrium, as cycling is impossible:

Theorem 6. *An instance of 2-FLG admits a subgame perfect equilibrium if there exists a balanced client equilibrium profile $\sigma \in \Omega$.*

Proof. Let $\sigma \in \Omega$ denote a balanced client equilibrium profile. We consider the facility game induced by σ and show that ℓ_{sort} increases lexicographically whenever a facility player performs an improving move.

Consider some facility placement profile $\mathbf{s} \in S$ and let f_{im} denote the facility player which performs an improving move. Let $\mathbf{s}'$ denote the FPP that would result from this move, and assume by contradiction that $\ell_{sort}(\mathbf{s}', \sigma) \leq \ell_{sort}(\mathbf{s}, \sigma)$. Then there must be at least one facility whose load was decreased. Construct the set $\mathcal{F}' \subseteq \mathcal{F}$ as follows:

1. Let $\mathcal{F}'$ denote the set of facilities with smaller load for $(\mathbf{s}', \sigma)$ than for $(\mathbf{s}, \sigma)$.
2. Refine $\mathcal{F}'$ by only keeping the facilities with minimal load for $(\mathbf{s}', \sigma)$.
3. Refine $\mathcal{F}'$ again, by only keeping the facilities with maximal load for $(\mathbf{s}, \sigma)$.

Since at least one facility's load was decreased, $\mathcal{F}'$ cannot be empty. Thus, there is some facility $f_{dec} \in \mathcal{F}'$. If $\ell_{dec}(\mathbf{s}', \sigma) \geq \ell_{im}(\mathbf{s}', \sigma)$, then ℓ_{sort} has increased, and we are done. Now assume that $\ell_{dec}(\mathbf{s}', \sigma) < \ell_{im}(\mathbf{s}', \sigma)$. Since the load on f_{dec} decreased, there must be some client v_i that patronizes f_{dec} with a strictly smaller probability for state $(\mathbf{s}', \sigma)$ than for $(\mathbf{s}, \sigma)$. Let $F_i(s', \sigma)$ denote the set of facilities considered by v_i for $(\mathbf{s}, \sigma)$. Then, since σ is balanced and $\ell_{dec}(\mathbf{s}', \sigma) < \ell_{im}(\mathbf{s}', \sigma)$, we find $f_{im} \notin F_i(s', \sigma)$. Since f_{im} was the only facility that moved, we find that $F_i(s', \sigma) \subseteq N_s(v_i)$.

Since σ is balanced, we find that $\ell_j(\mathbf{s}', \sigma) \leq \ell_{dec}(\mathbf{s}', \sigma)$ for all facilities $f_j \in F_i(\mathbf{s}', \sigma)$. Since each such facility is also in $N_{\mathbf{s}}(v_i)$, and v_i considered f_{dec} for state $(\mathbf{s}, \sigma)$, this implies that $\ell_j(\mathbf{s}, \sigma) \geq \ell_{dec}(\mathbf{s}, \sigma)$. By the construction of $\mathcal{F}'$, we know every facility with a strictly larger load than f_{dec} for $(\mathbf{s}, \sigma)$, also has a strictly larger load for $(\mathbf{s}', \sigma)$. We conclude that $f_j(\mathbf{s}, \sigma) = f_{dec}(\mathbf{s}, \sigma)$ and $f_j(\mathbf{s}', \sigma) = f_{dec}(\mathbf{s}', \sigma)$ for all $f_j \in F_i(\mathbf{s}', \sigma)$. In other words: $F_i(\mathbf{s}', \sigma) \subseteq \mathcal{F}'$.

Next, let $V_{\mathcal{F}'}(\mathbf{s}, \sigma) \subseteq V$ denote the set of clients that consider at least one facility in $\mathcal{F}'$ for state $(\mathbf{s}, \sigma)$. Since f_{dec} and v_i were chosen arbitrarily, $F_i(\mathbf{s}', \sigma) \subseteq \mathcal{F}'$ for all $v_i \in V_{\mathcal{F}'}(\mathbf{s}', \sigma)$. Furthermore, any client in $V_{\mathcal{F}'}(\mathbf{s}, \sigma)$ is also in $V_{\mathcal{F}'}(\mathbf{s}', \sigma)$. This gives:

$$\sum_{f_j \in \mathcal{F}'} \ell_j(\mathbf{s}', \sigma) = w(V_{\mathcal{F}'}(\mathbf{s}', \sigma)) \geq w(V_{\mathcal{F}'}(\mathbf{s}, \sigma)) \geq \sum_{f_j \in \mathcal{F}'} \ell_j(\mathbf{s}, \sigma). \tag{4}$$

As the load of the facilities in $\mathcal{F}'$ all have an equal load for state $(\mathbf{s}', \sigma)$, none of these facilities' loads were decreased by the move; a contradiction.

Thus, $\ell_{dec}(\mathbf{s}', \sigma) \geq \ell_{im}(\mathbf{s}', \sigma)$. It follows that $\ell_{sort}(\mathbf{s}', \sigma) > \ell_{sort}(\mathbf{s}, \sigma)$ lexicographically, which shows that every improving move increases ℓ_{sort} lexicographically. As $|S|$ is finite, any sequence of improving moves reaches some pure facility equilibrium $\mathbf{s}^*$ in a finite number of moves. Since $\mathbf{s}^*$ is a facility equilibrium for the facility game induced by client equilibrium profile σ, $(\mathbf{s}^*, \sigma)$ is a subgame perfect equilibrium. We conclude that a subgame perfect equilibrium exists. □

We have shown that the vector of sorted loads works as a lexicographical potential function. This raises the question of whether an exact potential function also exists. Monderer and Shapley [21] showed that such *potential games* posses many useful properties. Lemma 8 shows that no such function exists for our model, not even for *unweighted* 2-FLG.

Lemma 8. *Facility games induced by a client equilibrium profile for an instance of 2-FLG are not generally exact potential games. This is independent of the chosen equilibrium profile, and also holds for unrestricted and unweighted games.*

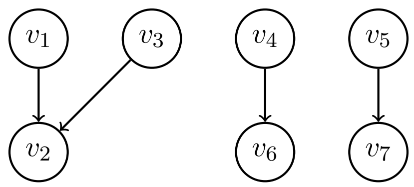


Figure 6: Host graph H with all client weights equal to 1

Proof. Consider instance $(H, 2)$ of unweighted. unrestricted 2-FLG, where H is the graph shown in Figure 6. Next, consider the facility game induced by some equilibrium $\sigma \in \Omega$. Note that for each of the facility placement profiles $(v_2, v_1), (v_6, v_1), (v_2, v_7)$ and (v_6, v_7), only one client equilibrium exists. The facility loads for each of these profiles can be found in Table 2.

Table 2: Facility loads for different facility placement profiles

$\mathbf{s} = (s_1, s_2)$	$\ell_1(\mathbf{s}, \sigma)$	$\ell_2(\mathbf{s}, \sigma)$
(v_2, v_1)	2	1
(v_6, v_1)	2	1
(v_2, v_7)	3	2
(v_6, v_7)	2	2

Now we check what happens to the load of the moving facility when moving from (v_1, v_2) to (v_2, v_7) in two different ways: through (v_6, v_1) and through (v_6, v_7). We sum the changes in load resulting from the moves in both cases. When moving through (v_6, v_1), we find:

$$\Big(\ell_1(v_6, v_1) - \ell_1(v_2, v_1)\Big) + \Big(\ell_2(v_6, v_7) - \ell_2(v_6, v_1)\Big) = (2 - 2) + (2 - 1) = 1.$$

When moving through (v_2, v_7), we find:

$$\Big(\ell_2(v_2, v_7) - \ell_2(v_2, v_1)\Big) + \Big(\ell_1(v_6, v_7) - \ell_1(v_2, v_7)\Big) = (2 - 1) + (2 - 3) = 0.$$

As these two values are different, an exact potential function cannot exist. As we made no assumptions on the client equilibrium profile, this result holds for all possible client equilibria. □

We conclude that the use of a lexicographical potential function is warranted.

The existence of a balanced client equilibrium profile $\sigma \in \Omega$ is thus a sufficient condition for the existence of an SPE. This warrants some research into what the class of instances that admit balanced equilibria looks like. Lemma 9 provides a nontrivial class of instances that always admit a subgame perfect equilibrium.

Lemma 9. *Every instance $(H, \mathcal{U}, 2)$ of unweighted 2-FLG with two facility players admits a balanced client equilibrium profile.*

Proof. Consider some instance $(H, \mathcal{U}, 2)$ of unweighte, 2-FLG. We provide an algorithm that greedily constructs a balanced client equilibrium for any FPP $\mathbf{s} \in S$. We will fix the strategies for the clients sequentially and consider only the facility loads caused by the already assigned clients. First, fix the strategies of all clients that have at most one facility in their shopping range. These clients only have one feasible strategy, so no decision can be made yet. This leaves only the clients in $A_s(f_1) \cap A_s(f_2)$. If this set is empty, we may terminate and the resulting client subgame profile is clearly balanced and a client equilibrium. If it is not, we consecutively *assign* these clients to the facility with lesser load, until we either run out of clients or the loads on the facilities become equal. Here, assigning a client to a facility means that this client patronizes the facility with probability one.

If we run out of clients, we terminate the algorithm, and claim that the resulting client subgame profile is balanced and a client equilibrium. If the load on the facilities is equal, both of these are trivially true. If not, then the facility which is considered by all clients in $A_s(f_1) \cap A_s(f_2)$ is the facility with the smaller load. To see this, note that the facility to which all clients in $A_s(f_1) \cap A_s(f_2)$ were assigned must have had a strictly smaller load than the other facility prior to the assignment of the last client. Furthermore, as the load on both facilities is integer at every step of the algorithm, the difference must be at least 1. We conclude that the former facility has a load at most equal to that of the other facility at termination. As all clients in $A_s(f_1) \cap A_s(f_2)$ consider the facility with a smaller load exclusively, the client subgame profile is both balanced and a client equilibrium.

If the loads ever become equal, we fix the strategies of all remaining unassigned clients to $(\frac{1}{2}, \frac{1}{2})$. It follows that the load on both facilities is equal for the resulting client subgame profile. The client subgame profile is thus balanced. It is also clear that none of the clients can improve. Thus, The resulting client subgame profile is a balanced client equilibrium. We can construct a balanced client equilibrium profile $\sigma \in \Omega$ by applying this algorithm for every FPP $\mathbf{s} \in S$. □

Neither the assumption of equal client weights, nor the assumption of exactly two facility players may be dropped, as shown in Lemmas 10 and 11.

Lemma 10. *An instance $(H, \mathcal{U}, k)$ of unweighted 2-FLG with $k > 2$, Ω might admit no balanced client equilibrium profile.*

Proof. Consider the unweighted, unrestricted instance $(H, 3)$ and facility placement profile $\mathbf{s} = (v_1, v_2, v_3) \in S$, with H as shown in Figure 7.

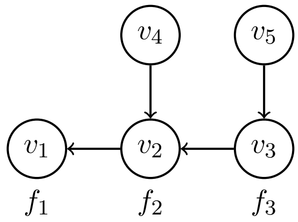


Figure 7: Host graph H with facility placement $\mathbf{s}$

For the client game induced by $\mathbf{s}$, the unique balanced client subgame profile is:

$$\sigma(\mathbf{s}) = \Big((1,0,0),(\frac{2}{3},\frac{1}{3},0),(0,\frac{1}{3},\frac{2}{3}),(0,1,0),(0,0,1)\Big)$$

This induces facility loads $\ell_1(\mathbf{s},\sigma(\mathbf{s})) = \ell_2(\mathbf{s},\sigma(\mathbf{s})) = \ell_3(\mathbf{s},\sigma(\mathbf{s})) = 1\frac{2}{3}$. As all facility loads are equal, ℓ_{sort} must be lexicographically maximal, which proves that $\sigma(\mathbf{s})$ is balanced. Note that no other client subgame profile induces these facility loads. However, $\sigma(\mathbf{s})$ is not a client equilibrium, as clients v_2 and v_3 can both improve. □

Lemma 11. *An instance $(H, 2)$ of 2-FLG might admit no balanced client equilibrium.*

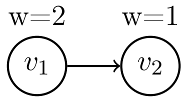


Figure 8: Host graph H

Proof. Consider instance $(H, 2)$ of unrestricted 2-FLG, with H as shown in Figure 8. Then for the client game induced by $\mathbf{s} = (v_1, v_2)$, the only client equilibrium is $\sigma(\mathbf{s}) = ((1,0),(0,1))$. However, $\sigma(\mathbf{s})$ is not balanced, as $\ell_1(\mathbf{s},\sigma(\mathbf{s})) = 2 > 1 = \ell_2(\mathbf{s},\sigma(\mathbf{s}))$ but client $v_1 \in A_{\mathbf{s}}(f_1) \cap A_s(f_2)$ patronizes f_1 with a nonzero probability. We conclude that no balanced client equilibrium profile exists. □

Note that while neither of the instances used in the proofs of Lemmas 10 and 11 admits a balanced client equilibrium profile, both *do* admit a subgame perfect equilibrium. The existence of a balanced client equilibrium profile is thus not *necessary* for the existence of an SPE:

Lemma 12. *The existence of a balanced client equilibrium profile is not necessary for the existence of a subgame perfect equilibrium.*

Proof. Consider instance $(H, 2)$ of 2-FLG where H is the host graph shown in Figure 8. In the proof of Lemma 11, we showed that $(H, 2)$ does not admit a balanced client equilibrium profile. However, it does admit a subgame perfect equilibrium $(\mathbf{s}, \sigma)$:

$$\mathbf{s} = (v_1, v_2)$$

$$\sigma(s') = \begin{cases} \{(\frac{1}{2},\frac{1}{2}),(0,0)\} & \text{for } \mathbf{s}' = (v_1, v_1) \\ \{(1,0),(0,1)\} & \text{for } \mathbf{s}' = (v_1, v_2) \\ \{(0,1),(1,0)\} & \text{for } \mathbf{s}' = (v_2, v_1) \\ \{(1,0),(0,1)\} & \text{for } \mathbf{s}' = (v_2, v_2) \end{cases}$$

It is trivial to verify that this state is a subgame perfect equilibrium. □

Lemma 12 suggests that further sufficient conditions for the existence of SPE might exist, besides the existence of a balanced client equilibrium profile. The condition of equal client weights is one such condition; the existence of an SPE for unweighted 2-FLG is guaranteed. We now develop further results concerning balancedness to prove this.

4.2 The Class Set

In many cases, balanced client subgame profiles induce identical loads on many facilities. Furthermore, the sets of clients considering such a group of facilities are the same for all balanced client subgame profiles. In this chapter, we introduce the *class set* of a client game; a great tool for analyzing balanced client subgame profiles for that client game. The *class set* is a partitioning of the facilities and the covered clients into different *classes*, based on their interactions for balanced client subgame profiles.

Definition 4.4. Consider some instance of 2-FLG and a facility placement profile $\mathbf{s} \in S$. For the client game induced by $\mathbf{s}$, we define the *class set* $\mathcal{C} = \{C_1, \ldots, C_m\}$ with classes $C_p := (\mathcal{F}_p, V_p)$ for $p = 1, \ldots, m$ as the set of maximum cardinality satisfying the following conditions:

1. $\{\mathcal{F}_p \mid p = 1, \ldots, m\}$ is a partition of $\mathcal{F}$
2. $\{V_p \mid p = 1, \ldots, m\}$ is a partition of the set of clients covered by $\mathbf{s}$.
3. For every balanced client subgame profiles $\sigma(\mathbf{s}) \in \Phi_{\mathbf{s}}$ and class $(\mathcal{F}_p, V_p) \in \mathcal{C}$, the clients in V_p exclusively consider facilities in $\mathcal{F}_p$.

Theorem 7. *For every instance of 2-FLG and facility placement profile* $\mathbf{s}$*, the client game induced by* $\mathbf{s}$ *has a unique class set* $\mathcal{C}$*.*

Proof. Note that $\{(\mathcal{F}, V_{cov}(\mathbf{s}))\}$ trivially satisfies all conditions in the definition for the class set, except possibly the maximum cardinality requirement. If $\{(\mathcal{F}, V_{cov})\}$ is not the class set, then there is another set satisfying the conditions that is of maximal cardinality, which is then the class set. Thus, the class set always exists.

To see that the class set is unique, assume by contradiction that there exist two distinct class sets $\mathcal{C}$ and $\mathcal{C}'$. Then there are a client $v_i \in V_{cov}$ and a facility $f_j \in \mathcal{F}$ which are in the same class in $\mathcal{C}$ but in different classes in $\mathcal{C}'$. If not, then every class in $\mathcal{C}$ is a subset of a class in $\mathcal{C}'$, which means that either $\mathcal{C} = \mathcal{C}'$ or the class sets have different cardinalities; both contradictions.

Let $C_j = (\mathcal{F}_j, V_j)$ and $C'_j = (\mathcal{F}'_j, V'_j)$ denote the classes containing f_j in $\mathcal{C}$ and $\mathcal{C}'$ respectively. Define $C_\alpha := (\mathcal{F}_\alpha, V_\alpha) = (\mathcal{F}_j \cap \mathcal{F}'_j, V_j \cap V'_j)$, and define $C_\beta := (\mathcal{F}_\beta, V_\beta)$ as the class containing all clients and facilities of C_j that are not in C_α. We claim that replacing C_j with both C_α and C_β in $\mathcal{C}$ does not violate any of the conditions in the definition of the class set. We first show that $\mathcal{F}_\alpha, \mathcal{F}_\beta, V_\alpha$, and V_β are nonempty. Since f_j and v_i are in $\mathcal{F}_\alpha$ and V_β respectively, these sets are both nonempty. V_α is nonempty as it contains all clients that consider f_j for some balanced $\sigma(\mathbf{s})$. Since the balanced load on each facility is nonzero, at least one such client exists. $\mathcal{F}_\beta$ is nonempty as it contains all facilities considered by v_i for some balanced client subgame profile. As every client in V_{cov} considers at least one facility for every client subgame profile, this set is nonempty. To see that condition (3.) holds, consider some $v_i \in V_\alpha$. Since $v_i \in V_j$ and $v_i \in V'_j$, all facilities considered by v_i for some balanced client equilibrium are in both $\mathcal{F}_j$ and $\mathcal{F}'_j$, and thus in $\mathcal{F}_j \cap \mathcal{F}'_j = \mathcal{F}_\alpha$. Similarly, for any client $v_i \in C_\beta$, we know that $v_i \notin V'_j$. As such, v_i is in some other class of $\mathcal{C}'$. All facilities considered by v_i for any balanced client subgame profile are also in this other class of $\mathcal{C}'$, and thus not in $\mathcal{F}_j'$. As such, all these facilities are in $\mathcal{F}_\beta$. We conclude that $\mathcal{C}$ can be refined without violating the conditions for a class set. Thus, $\mathcal{C}$ was not a set of maximal cardinality satisfying these conditions; a contradiction. □

For ease of notation, we define the *balanced load* of a facility for FPP $\mathbf{s}$ as the load on this facility for state $(\mathbf{s}, \sigma(\mathbf{s}))$, where $\sigma(\mathbf{s})$ is some balanced client subgame profile.

Lemma 13. *Consider some instance of 2-FLG and an FPP $\mathbf{s} \in S$, and let $\mathcal{C}$ denote the class set for the client game induced by $\mathbf{s}$. Then, for any class $(\mathcal{F}_p, V_p) \in \mathcal{C}$, all facilities in $\mathcal{F}_p$ have the same balanced load for $\mathbf{s}$.*

Proof. Assume the opposite is true. Then there is a class $C_p := (\mathcal{F}_p, V_p) \in \mathcal{C}$ such that not all facilities in $\mathcal{F}_p$ have equal balanced loads for $\mathbf{s}$. Let $\mathcal{F}_{min}$ denote the set of facilities in $\mathcal{F}_p$ with minimal balanced load for $\mathbf{s}$. By the definition of balancedness (Def. 4.1) the clients in $A_{\mathbf{s}}(\mathcal{F}_p) \cap V_p$ exclusively consider facilities in $\mathcal{F}_{min}$ for any balanced client subgame profile. Thus, we can refine $\mathcal{C}$ by replacing C_p by two new classes: $(\mathcal{F}_{min}, A_{\mathbf{s}}(\mathcal{F}_{min}) \cap V_p)$ and $(\mathcal{F}_p \setminus \mathcal{F}_{min}, V_p \setminus A_{\mathbf{s}}(\mathcal{F}_{min}))$. We conclude that $\mathcal{C}$ is not a class set as it is not of maximum cardinality; a contradiction. □

Since the balanced load is equal for all facilities in a class, we can view this load as a property of the class itself.

Definition 4.5. Consider some instance of 2-FLG and some facility placement profile $\mathbf{s}$. Let $\mathcal{C}$ denote the class set for the client game induced by $\mathbf{s}$. For any class $C_p = (\mathcal{F}_p, V_p) \in \mathcal{C}$, we define the *class load* $\ell(C_p)$ as the balanced load of the facilities in $\mathcal{F}_p$.

Next, we establish two properties of the class set that are useful for later proofs. These properties explain why a class set might contain multiple classes with the same class loads, and are a consequence of the assumption that the class set cannot be refined (i.e., that it has maximum cardinality among all sets satisfying certain conditions).

The first property concerns proper, nonempty subsets of the set of facilities of a class $(\mathcal{F}_p, V_p)$. In particular, the total weight of clients in the intersection of the attraction range of any such subset and V_p is strictly larger than the sum of the balanced load on the facilities in this subset:

Lemma 14. *Let $\mathcal{C}$ be the class set for host graph H and $\mathbf{s} \in S$, and some class $C_p = (\mathcal{F}_p, V_p) \in \mathcal{C}$. Then the total weight of clients in the intersection of V_p and the attraction range of any proper, nonempty subset of $\mathcal{F}_p$ is strictly larger than the total balanced load on that subset of facilities. That is:*

$$w(A_s(F) \cap V_p) > \ell(C_p)|F| \quad \forall F \subsetneq \mathcal{F}_p \text{ s.t. } F \neq \emptyset$$

Proof. Consider some class $C_p = (\mathcal{F}_p, V_p)$ and some set of facilities $F \subsetneq \mathcal{F}_p$, $F \neq \emptyset$. For any balanced $\sigma(\mathbf{s})$, the facilities in F can only be considered by clients in $A_s(F) \cap V_p$. Since the balanced load on each of these facilities is $\ell(C_p)$, we find: $A_{\mathbf{s}}(F) \cap V_p \geq \ell(C_p)|F|$. Now assume by contradiction that this holds with equality. Then, for any balanced client subgame profile $\sigma(\mathbf{s})$, the clients in $A_{\mathbf{s}}(F) \cap V_p$ must exclusively consider facilities in F. But then we can refine $\mathcal{C}$ by splitting C_p into classes $(F, A_{\mathbf{s}}(F) \cap V_p)$ and $(\mathcal{F}_p \setminus F, V_p \setminus A_{\mathbf{s}}(F))$. Thus, $\mathcal{C}$ is not a class set. □

As an example, consider the two unweighted, unrestricted instances $(H, 2)$ and $(H', 2)$ of 2-FLG, where H and H' are the host graphs shown in Figure 9. Let $\mathbf{s} = (v_1, v_2)$ and $\mathbf{s}' = (w_1, w_2)$ denote facility placement profiles for the respective instances, and consider the induced client games. In both games, the balanced load on both facilities is 2. However, the class set $\mathcal{C}$ for $\mathbf{s}$ contains two classes, while the class set $\mathcal{C}'$ for $\mathbf{s}'$ contains just one:

$$\mathcal{C} = \Big\{ \Big(\{f_1\}, \{v_1, v_4\} \Big), \Big(\{f_2\}, \{v_2, v_3\} \Big) \Big\}$$
$$\mathcal{C}' = \Big\{ \Big(\{f_1, f_2\}, \{w_1, w_2, w_3, w_4\} \Big) \Big\}.$$

This difference naturally follows from the definition of the class set. For both graphs, every balanced client subgame profile must induce a load of 2 on both facilities. To achieve this for f_1 in host graph H, we require all clients in $A_\mathbf{s}(f_1) = \{v_1, v_4\}$ to consider f_1 exclusively. As such, there is no balanced client equilibrium where v_1 or v_4 considers f_2. The maximum cardinality requirement ensures that $\mathcal{C}$ is the unique class set. This is exactly what Lemma 14 states; if we were to (incorrectly) assume that all facilities and clients belong to the same class for the client game induced by $\mathbf{s}$, then the lemma is violated for $F = \{f_1\}$, which implies that the "class set" can be refined.

On the other hand, applying this lemma to $\mathcal{C}'$ does not give a contradiction. This is correct, as $\mathcal{C}'$ cannot be refined. To see this, note that both w_1 and w_3 may consider either f_1, f_2 or both, for different balanced client subgame profiles. As such, w_1, w_3, f_1, and f_2 all belong to the same class. Furthermore, w_2 and w_4 are in the same class as f_2 and f_1 respectively. We conclude that $\mathcal{C}'$ is indeed the class set.

(a) Host graph H with facility placement $\mathbf{s} = (v_1, v_2)$

(b) Host graph H' with facility placement $\mathbf{s}' = (w_1, w_2)$

Figure 9: Two similar instances with different class sets

The second property we discuss may be intuited as the condition that each class consists of one "connected component". Formalizing this notion of "connectivity" is somewhat cumbersome, however. We consider a "path" connecting two elements of the class as an alternating sequence of clients and facilities, where every facility in the sequence is in the shopping range of the clients preceding and succeeding it. For every class, such a sequence must exist from every client or facility to every other client or facility. If we can split a class into two or more "components" between which no "paths" exist, then we can refine the class set by replacing that class with its "components".

All connected graphs possess the property that for every partition $(V', V \setminus V')$ of its vertex set V, the cut-set of the cut induced by this partition is nonempty. In Lemma 15, we characterize our notion of the "connectivity" of a class $C_p = (V_p, \mathcal{F}_p)$ somewhat similarly: for every partition $(F, \mathcal{F}_p \setminus F)$ of the class' facilities, we require at least one client in V_p to be in the attraction range of both F and $\mathcal{F}_p \setminus F$.

Lemma 15. *Consider an instance of 2-FLG with host graph H and some facility placement profile $\mathbf{s} \in S$. Let $\mathcal{C}$ denote the class set for client game induced by $\mathbf{s}$. Then for any class $C_p = (\mathcal{F}_p, V_p) \in \mathcal{C}$, we find:*

$$V_p \cap A_\mathbf{s}(F) \cap A_\mathbf{s}(\mathcal{F}_p \setminus F) \neq \emptyset \quad \textit{for every partition } (F, \mathcal{F}_p \setminus F) \textit{ of } \mathcal{F}_p \tag{5}$$

Proof. Consider some class $C_p = (\mathcal{F}_p, V_p) \in \mathcal{C}$. Now assume by contradiction that there exists $F \subsetneq \mathcal{F}_p, F \neq \emptyset$ such that $V_p \cap A_{\mathbf{s}}(F) \cap A_{\mathbf{s}}(\mathcal{F}_p \setminus F) = \emptyset$. Then for any balanced client subgame profile, clients in $A_{\mathbf{s}}(F) \cap V_p$ exclusively consider facilities in F, and clients in $A_{\mathbf{s}}(\mathcal{F}_p \setminus F) \cap V_p$ exclusively consider facilities in $\mathcal{F}_p \setminus F$. As such, we can replace C_p by the classes $(F, A_{\mathbf{s}}(F) \cap V_p)$ and $(\mathcal{F}_p \setminus F, A_{\mathbf{s}}(\mathcal{F}_p \setminus F) \cap V_p)$. Since $\mathcal{C}$ can be refined, it is not a class set; a contradiction. □

4.3 An algorithm for finding balanced profiles and the class set

In this chapter, we provide a basic algorithm for finding a balanced client subgame profile and the class set and investigate its running time. In particular, we show that for unweighted 2-FLG, we can always find a balanced client subgame profile and the class set in time polynomial in the input size. The same is true for general instances of 2-FLG, given some minor assumptions on the representation of the client weights in the input.

Finding Balanced Client Profiles

The key idea for the algorithm is to relate client games to maximum flow networks. For any client game, we construct a flow network for which there is a bijective relation between feasible client subgame profiles and maximum flows. Furthermore, the facility loads induced by a client subgame profile equal the flow through specific edges for the corresponding maximum flow. We can control the capacity of these edges to find balanced client subgame profiles.

Definition 4.6. Consider an instance $(H, \mathcal{U}, k)$ of 2-FLG with host graph $H = (V, A, w)$ and some $\mathbf{s} \in S$, and let $W := \sum_{v \in V^{cov}(\mathbf{s})} w(v_i)$ denote the sum of the weights of the clients covered by $\mathbf{s}$. Then, flow network $G = (V', E'_1 \cup E'_2 \cup E'_3)$ is the *client game flow network* for the client game induced by $\mathbf{s}$ if:

$$
\begin{aligned}
V' &:= V^{cov}(\mathbf{s}) \cup \mathcal{F} \cup \{s, t\} \\
E'_1 &:= \{(s, v_i, w(v_i)) \mid v_i \in V^{cov}(\mathbf{s})\} \\
E'_2 &:= \{(v_i, f_j, \infty) \mid f_j \in N_{\mathbf{s}}(v_i)\} \\
E'_3 &:= \{(f_j, t, c_j) \mid f_j \in \mathcal{F}\},
\end{aligned}
$$

where we denote edges as *(tail, head, capacity).*

The capacities of the edges in E'_3 are not fixed, as these are the aforementioned edges that can be used to control which maximum flows may be found. As an example, consider the instance of 2-FLG with 3 facility players and host graph H as shown in Figure 10. Further assume that facility placement profile $\mathbf{s} := (v_2, v_4, v_5)$ is in S. Figure 11 shows the client game flow network G for the client game induced by $\mathbf{s}$.

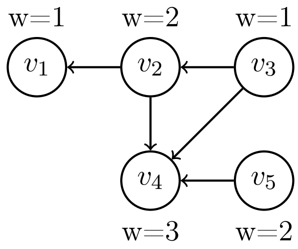


Figure 10: Host graph H

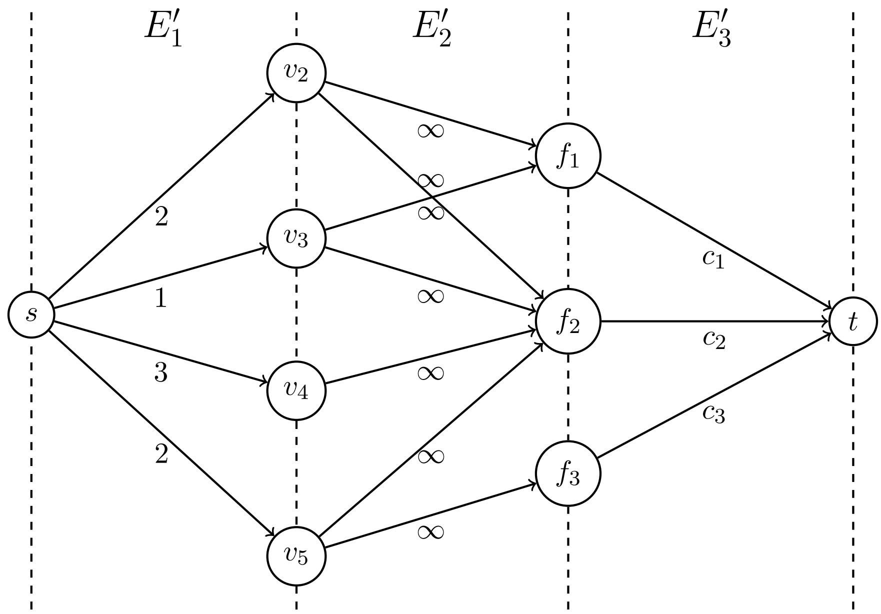


Figure 11: Client game flow network G for $\mathbf{s} = (v_2, v_4, v_5)$

The dashed lines indicate which edges belong to E'_1, E'_2, and E'_3 respectively, and do not represent edges themselves. Note that v_1 does not appear in the network, as it is not covered by $\mathbf{s}$. Further note that any (maximum) s,t-flow φ has flow value at most $W = 8$, since the flow value cannot exceed the sum of the capacities of edges in E'_1. We now show how to relate each (maximum) s,t-flow in G to a specific client subgame profile for the client game induced by $\mathbf{s}$.

Definition 4.7. Consider some instance of 2-FLG and let $G = (V', E'_1 \cup E'_2 \cup E'_3)$ denote the client game flow network for the client game induced by some FPP $\mathbf{s} \in S$, with some given capacities on the edges of E'_3. Let φ denote some s,t-flow in G. Then, the client subgame profile $\sigma^\varphi(\mathbf{s})$ defined by:

$$\sigma^\varphi(\mathbf{s})_{i,j} = \begin{cases} \frac{\varphi(v_i f_j)}{w(v_i)} & \text{if } v_i \text{ is covered by } \mathbf{s} \\ 0 & \text{else} \end{cases} \tag{6}$$

is called the *profile representation* of φ. Similarly, φ is called the *flow representation* of $\sigma^\varphi(\mathbf{s})$.

We claim that the profile representation of some flow φ is a feasible client subgame profile if and only if φ has flow value W:

Lemma 16. *Consider some instance of 2-FLG and let $G' = (V', E'_1 \cup E'_2 \cup E'_3)$ denote the client game flow network for the client game induced by some FPP $\mathbf{s} \in S$. Let φ denote some s,t-flow in G. Then, the profile representation $\sigma^\varphi(\mathbf{s})$ of φ is a feasible client subgame profile if and only if φ has flow value W.*

Proof. φ saturates each edge in E'_1 if and only if it has flow value W. By the flow conservation constraints on the vertices of $V^{cov}(\mathbf{s})$:

$$\sum_{f_j \in A_{\mathbf{s}}(v_i)} \sigma^{\varphi}_{i,j} = \sum_{f_j \in A_{\mathbf{s}}(v_i)} \frac{\varphi(v_i f_j)}{w(v_i)} = \frac{c_{(s,v_i)}}{w(v_i)} = 1$$

is satisfied for all clients $v_i \in V^{cov}(\mathbf{s})$ if and only if φ has flow value W. This a necessary condition for feasibility (see Definition 2.1). It is self-evident that the other conditions for feasibility are satisfied for any s,t-flow. We conclude that $\sigma^\varphi(\mathbf{s})$ is feasible if and only if φ has flow value W. □

Finding the (unique) flow representation φ for any feasible client subgame profile $\sigma^\varphi(\mathbf{s})$ is trivial. The flow through the edges of E'_2 is given by (6). Furthermore, the flow conservation constraints ensure that every s,t-flow is uniquely characterized by the flow through the edges of E'_2. By Lemma 16, φ has flow value W, which requires all edges in E'_1 to be saturated by φ. The flow through an edge (f_j, t) in E'_3 equals the load on facility f_j for $(\mathbf{s}, \sigma^\varphi(\mathbf{s}))$:

$$\varphi_{f_j,t} = \sum_{\substack{v_i \in V \text{s.t.} \\ (v_i, f_j) \in E'_2}} \varphi_{v_i,f_j} = \sum_{\substack{v_i \in V \text{s.t.} \\ (v_i, f_j) \in E'_2}} w(v_i)\sigma^\varphi(\mathbf{s})_{i,j} = \sum_{v_i \in A_{\mathbf{s}}(f_j)} w(v_i)\sigma^\varphi(\mathbf{s})_{i,j} = \ell_j(\mathbf{s}, \sigma^\varphi(\mathbf{s})).$$

As such, the relation between feasible client subgame profiles and s,t-flows in G with flow value W is bijective. For the flow representation of some client subgame profile to be feasible in G, we require the capacities on E'_3 to be sufficiently large. This can be used to our advantage: to find a client subgame profile for which the loads on certain facilities are bounded from above, we set the capacities of E'_3 in G accordingly and find a maximum flow. If the maximum flow has flow value W, then the profile representation of this flow is a feasible client subgame profile satisfying the given bounds on the facility loads. If the flow value is less than W, then no client subgame profile satisfying the given bounds exists. In particular, if the total capacity on E'_3 is W, then this procedure finds a client subgame profile for which the induced loads on all facilities equal the capacities exactly (assuming such a profile exists). Recall that, by Theorem 4 and Lemma 6, a client subgame profile is balanced if and only if it induces a unique vector of facility loads. If this vector of facility loads is known, the procedure can find a balanced client subgame profile by finding a maximum flow.

To find this vector, we consider the client games as resource allocation problems. We aim to find an allocation of the clients (resources) to the facility players satisfying max-min fairness. Our algorithm is based on the *bottle-neck algorithm* [15] for finding max-min fair resource allocations. We first provide the mathematical representation of the algorithm and prove its correctness, and then apply it to an example. The reader is encouraged to consider the mathematical representation and the example side by side.

Algorithm 1 ComputeBalancedProfile($H = (V, A, w), \mathcal{F}, \mathbf{s}$)

1: construct directed graph $G = (V', E'_1 \cup E'_2 \cup E'_3)$
2: $V' \leftarrow V^{cov}(\mathbf{s}) \cup \mathcal{F} \cup \{s, t\}$ ▷ Construct client game flow network
3: $E'_1 \leftarrow \{(s, v_i, w(v_i)) \mid v_i \in V^{cov}(\mathbf{s})\}$ ▷ We denote edges as (tail, head, capacity)
4: $E'_2 \leftarrow \{(v_i, f_j, \infty) \mid f_j \in N_{\mathbf{s}}(v_i)\}$
5: $E'_3 \leftarrow \{(f_j, t, c_j) \mid f_j \in \mathcal{F}\}$
6: $J \leftarrow \{1, 2, \ldots, |\mathcal{F}|\}$, $\hat{J} \leftarrow \emptyset$ ▷ Initialize other variables
7: $W \leftarrow \sum_{v_i \in V^{cov}(\mathbf{s})} w(v_i)$
8: $\epsilon \leftarrow$ sufficiently small strictly positive number
9: **for** $j \in J$ **do**
10:     $c_j \leftarrow W$ ▷ Set capacities to guarantee existence of flow with value W
11: **end for**
12: **while** $J \neq \emptyset$ **do** ▷ Uniformly decrease the capacities of edges in E'_3
13:     $\gamma \leftarrow$ FindNextThreshold(G, J)
14:     **for** $j \in J$ **do**
15:         $c_j \leftarrow \gamma$
16:     **end for**
17:     **for** $j \in J$ **do** ▷ Check for each edge in E'_3 if we can decrease capacity
18:         $c_j \leftarrow \gamma - \epsilon$
19:         $\hat{\varphi} \leftarrow$ maximum (s, t)-flow in G
20:         $c_j \leftarrow \gamma$
21:         **if** $|\hat{\varphi}| < W$ **then**
22:             move j from J to $\hat{J}$
23:         **end if**
24:     **end for**
25: **end while**
26: $\varphi \leftarrow$ maximum (s, t)-flow in G ▷ Find max. flow and profile representation
27: **for** $v_i \in V$ **do**
28:     **for** $f_j \in \mathcal{F}$ **do**
29:         $\sigma^{\varphi}(\mathbf{s})_{i,j} \leftarrow \begin{cases} \frac{\varphi(v_i f_j)}{w(v_i)} & \text{if } v_i \text{ is covered by } \mathbf{s} \\ 0 & \text{else} \end{cases}$
30:     **end for**
31: **end for**
32: **return** $\sigma^{\varphi}(\mathbf{s})$

The function *FindNextThreshold* returns the smallest value γ, such that the network with $c_j = \gamma$ for all $j \in J$, still admits a s, t-flow with flow value W. We discuss the implementation of this function at a later point. For now, we simply assume that this function exists.

Theorem 8. *Assuming a working implementation of the function FindNextThreshold is used, Algorithm 1 computes a balanced client subgame profile for any client subgame profile given an instance of 2-FLG and FPP **s**.*

Proof. We first show that the algorithm terminates. Note that $|J|$ is finite, and nothing is added to J after initialization. To prove that the central while loop (line 17) has a finite number of iterations, it thus suffices to show that at least one element is moved from J to $\hat{J}$ in each iteration. To this end, consider an iteration with *threshold value* γ and assume by contradiction that for every $j \in J$, decreasing c_j to $\gamma - \epsilon$ does not lower the maximum attainable flow value. Then, for each $j \in J$, let φ^j denote the maximum flow found in line 19 of the algorithm (related to that j). By assumption, all of these flows have flow value W. Consider flow $\varphi^* = \frac{1}{|J|}\sum_{j\in J}\varphi^j$ and the network with $c_j = \gamma$ for all $j \in J$. In this network, φ^* has flow value W, but does not saturate (f_j, t) for any $j \in J$. But then γ was incorrectly chosen as a threshold value; a contradiction. It is self-evident that every other loop in the algorithm also only has a finite number of iterations. By assumption, "FindNextThreshold" always terminates, and so does a reasonable algorithm for finding maximum s,t-flows (e.g., the Edmonds-Karp algorithm[9]). We conclude that Algorithm 1 terminates when applied to any client game.

By Lemma 16, $\sigma^\varphi(\mathbf{s})$ is a feasible client subgame profile. To prove that is also balanced, we need an intermediate result. We claim that, at any point in the algorithm, any flow with flow value W saturates all edges (f_j, t) for $j \in \hat{J}$. To see this, note that j is moved from J to $\hat{J}$ only if decreasing c_j would lower the maximum attainable flow value over the network (so flow value W is no longer attainable). At that iteration of the algorithm, every flow with flow value W thus saturates (f_j, t). Since any flow that is feasible at a later iteration of the algorithm is also feasible at every preceding iteration, this saturation property still holds at every later iteration. Since $\hat{J} = \{1, 2, \ldots, k\}$ at termination, φ saturates all edges in E'_3.

We now show that $\sigma^\varphi(\mathbf{s})$ is balanced. Assume by contradiction that it is not. Then there are facilities $f_i, f_j \in \mathcal{F}$ such that $\ell_i(\mathbf{s}, \sigma^\varphi(\mathbf{s})) < \ell_j(\mathbf{s}, \sigma^\varphi(\mathbf{s}))$, and there is a client v in $A_\mathbf{s}(f_i) \cap A_\mathbf{s}(f_j)$ that considers f_j. Since $\ell_i(\mathbf{s}, \sigma^\varphi(\mathbf{s})) < \ell_j(\mathbf{s}, \sigma^\varphi(\mathbf{s}))$ and φ saturates all edges in E'_3, we find that $c_j > c_i$. Consider G at the iteration when j was moved to $\hat{J}$. Since i was moved to $\hat{J}$ at some later iteration (since $c_i < c_j$), φ does not saturate (f_i, t) at this iteration. Let $\epsilon > 0$ denote some sufficiently small, strictly positive number, and let φ' denote the flow obtained from φ by decreasing the flow along (v, f_j) and (f_j, t) by ϵ, and increasing the flow along (v, f_i) and $f_i, t)$ by ϵ. Then, φ' has load W and does not saturate (f_j, t). But then j would not have been moved to $\hat{J}$ in this iteration; a contradiction.

We conclude that Algorithm 1 always finds a balanced client subgame profile for the client game induced by $\mathbf{s}$. □

We now apply Algorithm 1 to an example. In particular, we revisit the instance of 2-FLG with the host graph as shown in Figure 10, and again consider the client game induced by $\mathbf{s} := \{v_2, v_4, v_5\}$. We apply Algorithm 1 with arguments $(H, \{f_1, f_2, f_3\}, \mathbf{s})$. At initialization, we construct the flow network G as shown in Figure 11 and assign $c_1 = c_2 = c_3 = W = 8$, $J = \{1, 2, 3\}$ and $\hat{J} = \emptyset$.

Next, we iterate through the main part of the algorithm. We uniformly decrease c_1, c_2 and c_3 until we reach the first threshold value: $\gamma = 3$. Ways to find this value will be discussed later, although we do verify its correctness now. To see that the threshold value cannot be larger, apply a max-flow algorithm for $\gamma = 3$ to obtain a (s, t)-flow with flow value $W = 8$. To see that it cannot be smaller, note that the (s, t)-cut (Q, R) with $Q = \{s, v_2, v_3, v_4, f_1, f_2\}, R = (v_5, f_3, t\}$ has a capacity of $2 + c_1 + c_2 = 2 + 2\gamma$. By the max-flow min-cut theorem, no s,t-flow with flow value $W = 8$ can exist for $\gamma < 3$.

Next, we determine which facility indices to move from J to $\hat{J}$. A facility index j is moved if and only if (f_j, t) is saturated for every maximum flow. This is true for 1 and 2, but not for 3. Note the relation to the client game, where clients $\{v_2, v_3, v_4\}$ have only f_1 and f_2 in range. For any client subgame profile, the combined load on f_1 and f_2 is therefore at least $w(v_2) + w(v_3) + w(v_4) = 6$. By construction, the sum of the flows through (f_1, t) and (f_2, t) is thus also at least 6. We fix $c_1 = c_2 = 3$ and move 1 and 2 from J to $\hat{J}$. This gives $J = \{3\}$ and $\hat{J} = \{1, 2\}$. In the next iteration, we decrease c_3 until another threshold is reached. Since the next threshold value is $\gamma = 2$, we fix $c_3 = 2$ and move 3 from J to $\hat{J}$.

After the second iteration, J is empty. The final step of the algorithm is to find a maximum flow φ and construct its profile representation $\sigma^{\varphi}(\mathbf{s})$. In this example, φ is the (unique) maximum flow shown in Figure 12. In general, the maximum flow is not unique, however. $\sigma^{\varphi}(\mathbf{s})$ is the pure, balanced client subgame profile where v_2 and v_3 consider f_1, v_4 considers f_2, and v_5 considers f_3. One can easily verify that this client subgame profile is balanced.

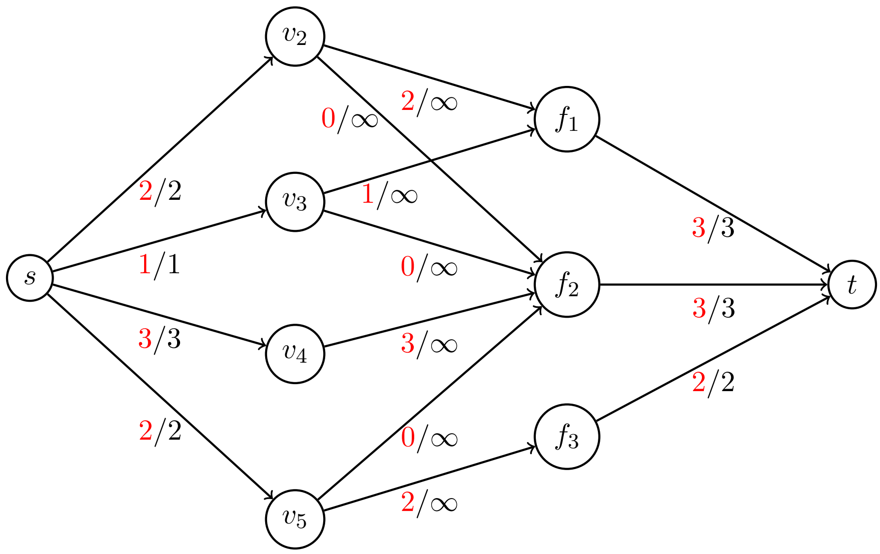


Figure 12: Network G at termination, flow φ is shown in *red*

Thresholds and Complexity

So far, we refrained from describing methods for finding threshold values. We now establish some naive implementations that are quite straightforward. However, proving polynomial running time for these methods is highly technical. As such, we seek to find a balance between providing a convincing argument and keeping the technical details to a minimum.

Lemma 17 shows how to construct a finite set X of *candidate* threshold values.

Lemma 17. *Consider an instance of 2-FLG with host graph $H := (V, A, w)$ and k facility players, and some FPP $\boldsymbol{s} \in S$. Let*

$$X := \left\{ \frac{1}{y} \sum_{v_i \in V'} w(v_i) \;\middle|\; y \in \{1, 2, \dots, k\}, V' \subseteq V^{cov}(\boldsymbol{s}) \right\}$$

denote the threshold candidate set. Then, all threshold values used when applying Algorithm 1 for H and $\boldsymbol{s}$ belong to the threshold candidate set X.

Proof. The threshold value γ corresponding to some iteration of Algorithm 1 equals the balanced load of each facility that is moved from J to $\hat{J}$ at that iteration. Recall that the balanced load of any facility equals the class load $\ell(C_p)$ of the class C_p containing that facility. Furthermore, for a class $C_p := (\mathcal{F}_p, V_p)$, all facilities in $\mathcal{F}_p$ are considered exclusively by clients in V_p and clients in V_p exclusively consider facilities in $\mathcal{F}_p$. We find:

$$\ell(C_p) = \frac{1}{|\mathcal{F}_p|} \sum_{v_i \in V_p} w(v_i).$$

Note that the number of facilities in a class $|\mathcal{F}_p|$ is an integer between 1 and k. Since a threshold value γ equals the class load of some class in the class set, we conclude that γ is in X. ☐

Since the cardinality of the candidate set is finite, we have shown how to construct a finite set that contains all threshold values. A naive implementation of the function $FindNextThreshold$ is to determine the candidate set, sort it, and perform a binary search to find the smallest value $\gamma \in X$ for which setting $c_j = \gamma$ for all $j \in J$ admits a flow on G with flow value W. Note, however, that X may contain up to $k \cdot 2^n$ elements, which is exponential in the number of players, and thus in the input size. Constructing and sorting the candidate set may thus take time exponential in the input size. For instances of unweighted 2-FLG, this problem is easily solved. For unweighted 2-FLG, the sum of the weights of the clients in any subset of V is integer and at most $W = |V^{cov}(\mathbf{s})|$. As such, X contains at most kn elements. The size of X is thus polynomial in the number of players.

Lemma 18. *For unweighted instances 2-FLG, we can find threshold values in time polynomial in the input size.*

Proof. Consider some instance of unweighted 2-FLG and a facility placement profile $\mathbf{s} \in S$. Define $W = |V^{cov}(\mathbf{s})| \leq n$. Then, the threshold candidate set X defined in Lemma 17 is:

$$X = \{\frac{y}{z} \mid y \in \{1, 2, \ldots, W\}, z \in \{1, 2, \ldots, k\}\}. \tag{7}$$

Constructing and sorting this set can then be done in time $O(|X| \log(|X|)) = O(kn \log(kn)$. Finding the next threshold value may be done by checking all values of X in descending order. This requires at most $O(kn)$ verifications (this is the maximum number of verifications over the whole of Algorithm 1, not just for one function call). Performing a binary search over X instead of checking all values further speeds up this process. A verification step requires finding a maximum flow in G, which was shown to be possible in time $O(k+n)(k^2n^2))$. We conclude that finding the next threshold value can be done in time $O((k+n)k^3n^3)$, which is polynomial in the input size. □

To use a similar argument for general instances, we must consider the binary representation of client weights in the input. We argue that for any reasonable binary representation of the client weights in the input, there exists a $\lambda \in \mathbb{N}^+$ such that scaling all client weights by λ results in an instance where all client weights are integer, and the maximum client weight is at most exponential in the size of the input. We refrain from discussing the technical details of this argument. Given this instance with integer weights, we further scale all loads by "k!" to guarantee that the candidate set X includes only integer values. The resulting client weights, and thus W, are still at most exponential in the size of the input. The threshold candidate set X for the scaled client game is a subset of $X' = \{1, 2, \ldots, W-1, W\}$, where W denotes the sum of the weights of the scaled client game. Performing a binary search on X' allows us to find threshold values in polynomial time with respect to the input size. The threshold values for the original game and the scaled game are proportional. We conclude that finding threshold values can be done in polynomial time with respect to the input size, given a reasonable binary representation of the input. Theorem 9 shows that Algorithm 1 runs in polynomial time for all instances under this assumption, and runs in polynomial time for unweighted instances either way.

Theorem 9. *Algorithm 1 runs in time polynomial in the input size, given that the threshold values can be found in polynomial time.*

Proof. To find the balanced client subgame profile, we apply Algorithm 1. Each loop in the algorithm has a polynomial number of iterations. The only operation that may still exceed polynomial time in the input size are running the $FindNextThreshold$ function and finding a maximum (s,t)-flow in G. The latter can be done in $O(|V'||E(G)|^2)$ time using Edmonds-Karp [9]. Since $|V'| \leq k + n + 2$ and $|E(G)| \leq n + k + kn$, maximum flows can thus be found in time $O\big((k+n)k^2n^2\big)$, which is polynomial in the input size. We conclude that if *FindNextThreshold* runs in polynomial time, so does Algorithm 1. $\square$

Finding the Class Set

We now show how to use Algorithm 1 to efficiently find the class set. We use the flow representation φ of a known balanced client subgame profile $\sigma^{\varphi}(\mathbf{s})$ to accomplish this. We also use the network G, where the capacities on E_3' are those found by the algorithm. That is, $c_j = \ell_j(\mathbf{s}, \sigma^{\varphi}(\mathbf{s}))$ for each $f_j \in \mathcal{F}$. Lemma 19 shows how to use the residual graph of φ in G to efficiently determine whether two distinct facilities belong to the same class. This lemma is applied multiple times to obtain the partition of the facilities into classes.

Lemma 19. *Consider some instance of 2-FLG and $\mathbf{s} \in S$, to which Algorithm 1 was applied. Let G and φ denote the network and maximum flow found by the algorithm. Then, two distinct facilities f_i and f_j belong to the same class if and only if the residual graph of φ contains an (f_i, f_j)- and an (f_j, f_i)-path.*

Proof. Assume that for a pair of facilities $f_i, f_j \in \mathcal{F}$, the residual graph of φ contains a (f_i, f_j)-path and a (f_j, f_i)-path. Then there is a tour T in the residual graph where f_i and f_j are both in the vertex sequence of T. This vertex sequence cannot contain s or t since the residual graph contains no edges with s as the tail vertex or t as the head vertex. The vertex sequence of T is thus an alternating sequence of client vertices and facility vertices. Consider flow φ' obtained from φ by sending some strictly positive amount of flow along T, and let $\sigma^{\varphi'}(\mathbf{s})$ denote the profile representation of φ'. Then, each client v_i in the vertex sequence of T considers the facility preceding it in the sequence for client subgame profile $\sigma^{\varphi}(\mathbf{s})$, and the facility succeeding it in the sequence for $\sigma^{\varphi'}(\mathbf{s})$. Since both are balanced subgame profiles, all consecutive facilities in the sequence belong to the same class. We conclude that all facilities in the vertex sequence of T belong to the same class, which includes f_i and f_j.

Now assume that f_i and f_j belong to the same class $C_p = (\mathcal{F}_p, V_p)$. We iteratively construct a set $F^* \subseteq \mathcal{F}_p$ with the property that for all $f \in F^*$, either $f = f_i$, or there exists a (f_i, f)-path in the residual graph. It suffices to give a procedure that adds f_j to F^* while maintaining the aforementioned property of F^* throughout this procedure. We initialize $F^* = \{f_i\}$ and repeatedly apply Lemma 14 with $F = \mathcal{F}_p \setminus F^*$. The lemma states that there is some client $v_i \in A_{\mathbf{s}}(F)$ that does not exclusively consider facilities in F for $\sigma^{\varphi}(\mathbf{s})$. This means that v_i considers some facility f_x in F^* and has a facility $f_y \in F$ in their shopping range. This implies that the residual graph contains edges (f_x, v_i) and (v_i, f_y). By the induction hypothesis, the residual graph contains a (f_i, f_x)-path and thus a (f_i, f_y)-path. We add f_y to F^* and continue to the next iteration. Since the number of facilities in a class is finite, this procedure eventually adds f_j to F^*. We conclude that the residual graph contains a (f_i, f_j)-path. By the same argument, the residual graph contains a (f_j, f_i)-path, as well. $\square$

If a balanced client subgame profile $\sigma(\mathbf{s})$ is known, but was not obtained from Algorithm 1, then it is possible to use Lemma 19 without running the entire algorithm. In this case, the client game flow network G is constructed as usual, where the capacities on E_3' follow immediately from the (known) balanced facility loads. As a maximum flow in G, we use the flow representation of $\sigma(\mathbf{s})$, which is trivial to find. Independent of how the balanced client subgame profile was found, we can use it to find the class set in polynomial time.

Theorem 10. *Given a balanced client subgame profile $\sigma^{\varphi}(\boldsymbol{s})$ for the client game induced by $\boldsymbol{s}$, we can find the class set $\mathcal{C}$ for this client game in polynomial time.*

Proof. As discussed, the flow network G and flow φ defined in Algorithm 1 for $\sigma^{\varphi}(\mathbf{s})$ can be found in polynomial time. We apply Lemma 19 to every pair of facilities to find the partition of $\mathcal{F}$ into classes. As there are $\frac{k(k-1)}{2}$ distinct pairs of facilities, this requires $O(k^2)$ applications of Lemma 19. Each application requires checking whether two directed paths between the facility vertices exist. This takes $O(kn)$ time using a BFS/DFS algorithm. Partitioning the facilities is thus done in $O(k^3n)$ time. To partition the covered clients into classes, simply add each client $v_i \in V^{cov}(\mathbf{s})$ to the class containing the facilities considered by v_i for $\sigma^{\varphi}(\mathbf{s})$. This is the correct partition since each client is in the same class as every facility they consider for any balanced client subgame profile. Partitioning the clients is done in time $O(kn)$. Using this procedure, the class set is found in $O(k^3n)$ time, which is polynomial in the input size. □

In practice, one would not apply Lemma 14 to every pair of facilities, but only those with equal balanced loads. After all, facilities with different balanced loads cannot belong to the same class.

5 Subgame Perfect Equilibria in Different Types of 2-FLG

In the previous chapter, we defined the concept of balanced client subgame profiles and showed that the existence of a balanced client equilibrium profile is a sufficient condition for the existence of SPE. In this chapter, we find further types of instances which are guaranteed to admit SPE. In particular, we show that every instance of *unweighted* 2-FLG admits a subgame perfect equilibrium. Additionally, we consider some types of instances that do not generally admit SPE, although one may expect them to.

5.1 Subgame Perfect Equilibria in Unweighted Games

We need several intermediate results to show that unweighted instances always admit subgame perfect equilibria. We first apply the theory of balanced client subgame profiles and the class set to show the existence of a specific type of client equilibria: *rounded* client equilibria. Next, we show how to find a pure, rounded client equilibrium profile that consistently favors the same facilities in tie-breaker scenarios, which we call *favoring* client equilibrium profiles. We prove the existence of a sequence of states with different *favoring* client equilibria and facility placement profiles that leads to a subgame perfect equilibrium. This result follows from the observation that the vector of sorted facilities loads increases lexicographically along this sequence of states and that it can only terminate at an SPE. As this proof is constructive, it also provides the outline for an algorithm for finding subgame perfect equilibria.

The types of client equilibria discussed in this chapter are based on balanced client subgame profiles, and thus on the class set. However, the definition of the class set is too constraining for these purposes. As such, we define the *tier set* $\mathcal{T}$, a coarser version of the class set. The tier set groups classes by their class loads. Even-indexed tiers contain all classes that have a specific integer load, while odd-indexed tiers contain all classes with loads in the *open* interval between two specific consecutive integers.

Definition 5.1. Consider an instance of unweighted 2-FLG with n clients and FPP $\mathbf{s} \in S$, and let $\mathcal{C}$ denote the class set for the client game induced by $\mathbf{s}$. We define the *tier set* $\mathcal{T}$ as the set $\{T_1, T_2, \ldots, T_{2n}\}$ with:

$$T_i = \begin{cases} \{C \in \mathcal{C} \mid \ell(C) \in (\frac{i-1}{2}, \frac{i+1}{2})\} & \text{for } i \text{ odd} \\ \{C \in \mathcal{C} \mid \ell(C) = \frac{i}{2}\} & \text{for } i \text{ even} \end{cases}$$

With slight abuse of notation, we will say that some client $v \in V$ or some facility $f \in \mathcal{F}$ is in some tier T_i when the client or facility belongs to a class that is, in turn, part of T_i. This is allowed since, like with classes, every facility and every covered client is in exactly one tier. Note that, due to the choice of indexing, the tier set may contain empty tiers. We now define the concept of *rounded* client equilibria

Definition 5.2. Consider some instance of unweighted 2-FLG and $\mathbf{s} \in S$. Let $\mathcal{T}$ denote the tier set for the client game induced by $\mathbf{s}$, and let $\sigma^b(\mathbf{s})$ denote some balanced client subgame profile. Then, we call a client subgame profile $\sigma(\mathbf{s}) \in \Phi_s$ *rounded* if for all facilities $f_j \in \mathcal{F}$:

$$\ell_j(\mathbf{s}, \sigma(\mathbf{s})) \in \{\lfloor \ell_j(\mathbf{s}, \sigma^b(\mathbf{s})) \rfloor, \lceil \ell_j(\mathbf{s}, \sigma^b(\mathbf{s})) \rceil\}$$

and clients exclusively consider facilities in their own tier. Naturally, we call a client profile σ *rounded* if $\sigma(\mathbf{s})$ is *rounded* for each $\mathbf{s} \in S$.

As their name suggests, rounded client subgame profiles may be interpreted as profiles obtained by "rounding" a balanced client subgame profile to guarantee integer facility loads. Like balanced profiles, rounded client subgame profiles are easy to work with, especially so when they are also pure. We will show that pure, rounded client equilibria exist for every client game for unweighted instances of 2-FLG. This holds even if we require clients to exclusively consider facilities in their own class, not just in their own tier. We later provide an algorithm for finding a pure, rounded client equilibrium that always finds one also satisfying this stricter condition. To construct and prove the correctness of this algorithm, certain preliminaries are necessary. First, Lemma 20 gives a characterization of pure client equilibria specific to unweighted instances.

Lemma 20. *Consider some instance of unweighted 2-FLG and facility placement profile $\boldsymbol{s} \in S$, and let $\sigma(\boldsymbol{s})$ denote some pure client subgame profile. Then, $\sigma(\boldsymbol{s})$ is a client equilibrium if and only if the following holds for each client $v_i \in V$:*

$$\ell_{j'}(\boldsymbol{s}, \sigma(\boldsymbol{s})) \geq \ell_j(\boldsymbol{s}, \sigma(\boldsymbol{s})) - 1 \quad \forall f_{j'} \in N_{\boldsymbol{s}}(v_i),$$

where f_j denotes the (unique) facility considered by v_i.

Proof. Since v_i exclusively considers f_j, it is trivial to find the v_i-excluded loads for all facilities:

$$\ell_{-i,j'}(\mathbf{s}, \sigma(\mathbf{s})) = \begin{cases} \ell_{j'}(\mathbf{s}, \sigma(\mathbf{s})) & \text{for } j' \neq j \\ \ell_{j'}(\mathbf{s}, \sigma(\mathbf{s})) - 1 & \text{for } j' = j. \end{cases}$$

Substituting these values in Theorem 1 gives the desired result. □

For some of the upcoming proofs, it is useful to characterize client games as bipartite graphs, where pure client subgame profiles are represented by the set of edges we call saturated. To this end, we define *client graphs*, *client class graphs*, and *client tier graphs.*

Definition 5.3. Consider instance $(H, \mathcal{U}, k)$ of unweighted 2-FLG and facility placement profile $\mathbf{s} \in S$. The *client graph* G for the client game induced by $\mathbf{s}$ is the bipartite, undirected graph with vertex set $V(G)$ and edge set $E(G)$ with:

$$\begin{aligned} V(G) &:= V^{cov}(\mathbf{s}) \cup \mathcal{F} \\ E(G_p) &:= \{(v_i, f_j) \in V^{cov}(\mathbf{s}) \times \mathcal{F} \mid v_i \in A_{\mathbf{s}}(f_j)\} \end{aligned}$$

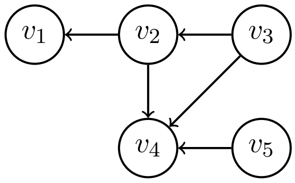
That is, a client vertex and a facility vertex in the client graph are connected by an edge if the facility is in the clients' shopping range.

Definition 5.4. Consider instance $(H, \mathcal{U}, k)$ of unweighted 2-FLG and facility placement profile $\mathbf{s} \in S$, and let $\mathcal{C}$ denote the class set for the client game induced by $\mathbf{s}$. Then, for each class $(\mathcal{F}_p, V_p) \in \mathcal{C}$, we define the *client class graph* $G_p := (V(G_p), E(G_p))$ as the subgraph induced by $\mathcal{F}_p \cup V_p$ in the *client graph.* That is,

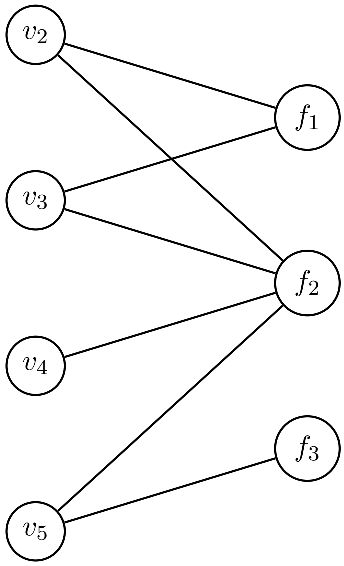
$$\begin{aligned} V(G_p) &:= V_p \cup \mathcal{F}_p \\ E(G_p) &:= \{(v_i, f_j) \in V_p \times \mathcal{F}_p \mid v_i \in A_{\mathbf{s}}(f_j)\} \end{aligned}$$

We define the *client tier graph* for a tier $T_i \in \mathcal{T}$ analogously to the client class graph.

As an example, consider the instance of unweighted 2-FLG with facility players $\mathcal{F} := \{f_1, f_2, f_3\}$ and the host graph shown in Figure 13 (left). The client graph for the client game induced by $\mathbf{s} := \{v_2, v_4, v_5\} \in S$ is shown in Figure 13 (right). Note the similarities to the (weighted) instance shown in Figure 10 and its accompanying client game flow network in Figure 11: the client graph is the "E_2'-part" of the flow network.



(a) Host graph H



(b) Client graph G for $\mathbf{s} = (v_2, v_4, v_5)$

Figure 13: Example for client graph

Theorem 11. *For every instance of unweighted 2-FLG and facility placement profile $\mathbf{s} \in S$, there exists a pure, rounded client equilibrium $\sigma(\mathbf{s})$ for the client game induced by $\mathbf{s}$.*

Proof. Consider some instance of unweighted 2-FLG and client equilibrium $\mathbf{s} \in S$, and let $\mathcal{C}$ denote the class set for the client game induced by $\mathbf{s}$. For each class $C_p \in \mathcal{C}$, let $G_p = (V(G_p), E(G_p))$ denote the client class graph (see Definition 5.4) for C_p. We now show how to construct a pure, rounded client equilibrium $\sigma(\mathbf{s})$.

We initialize $\sigma(\mathbf{s})_{i,j} = 0$ for all $v_i \in V$, $f_j \in \mathcal{F}$ and repeatedly augment $\sigma(\mathbf{s})$ by assigning clients sequentially, on a class-by-class basis. At any step in the procedure, we call an edge (v_i, f_j) in $E(G_p)$ *saturated* if $\sigma(\mathbf{s})_{i,j} = 1$, and *unsaturated* otherwise. Furthermore, we call a trail between two elements of $V(G_p)$ *odd-alternating* if all odd-numbered edges are unsaturated and all even-numbered edges are saturated.

For every class $C_p = (V_p, \mathcal{F}_p) \in \mathcal{C}$, we apply a two-phase procedure. In the first phase, we assign clients in V_p to facilities in $\mathcal{F}_p$ until each of these facilities has a load of $\lfloor \ell(C_p) \rfloor$. This is achieved with the following steps:

1. Choose some $f_j \in \mathcal{F}_p$ with current load 0. If no such facility exists, terminate.
2. Find an odd-alternating (f_j, v_i)-trail in G_p for a yet unassigned client $v_i \in V_p$.
3. Augment $\sigma(\mathbf{s})$ such that the saturation of all edges in the trail is swapped. If $\ell_j(\mathbf{s}, \sigma(\mathbf{s})) = \lfloor \ell(C_p) \rfloor$, return to step 1. Else, return to step 2.

It remains to be shown that we can always find a trail in step 2. Assume by contradiction that at some point, there is no trail satisfying our criterion for some facility $f_j \in \mathcal{F}_p$. Let $\tilde{\mathcal{F}}_j$ denote the set of facilities in $\mathcal{F}_p$ that can be reached from f_j by some odd-alternating trail in G_p for $\sigma(\mathbf{s})$ at that point in the algorithm. Then, every client in $\tilde{V}_j := A_{\mathbf{s}}(\tilde{\mathcal{F}}_j) \cap V_p$ is connected to $\tilde{\mathcal{F}}_j$ by some saturated edge. Furthermore, none of these clients are adjacent to a facility in $\mathcal{F}_p \setminus \tilde{\mathcal{F}}_j$, as otherwise the edge connecting them would necessarily be unsaturated, meaning that the facility belongs to $\tilde{\mathcal{F}}_j$ by definition. Thus, $N_{\mathbf{s}}(\tilde{V}_j) \cap \mathcal{F}_p = \tilde{\mathcal{F}}_j$. But since the maximum load on any facility for the current $\sigma(\mathbf{s})$ is $\lfloor \ell(C_p) \rfloor$, and strictly less on f_j, we conclude that $|A_{\mathbf{s}}(\tilde{\mathcal{F}}_p) \cap V_p| < |\tilde{\mathcal{F}}_j| \cdot \lfloor \ell(C_p) \rfloor$, which contradicts the existence of a balanced flow with class load $\ell(C_p)$. We conclude that we can always find a trail in step 2.

Next, note that applying step 3 increments the load on f_j by one, and does not affect the load of any other facility. As both the balanced load and the number of facilities in a class are finite, the algorithm must terminate. At termination, $\ell_j(\mathbf{s}, \sigma(\mathbf{s})) = \lfloor \ell(C_p) \rfloor$, for each $f_j \in \mathcal{F}_p$, and $\sigma(\mathbf{s})$ is pure.

If $\ell(C_p)$ is integer, this first phase of the procedure assigns all clients in V_p. In this case, we are done with class C_p. If $\ell(C_p)$ is non-integer, we apply the second phase of the procedure, where we consecutively assign the remaining clients. We repeatedly perform the following operations until all clients in V_p are assigned:

1. Choose an unassigned client $v_i \in V_p$.
2. Find a odd-alternating (v_i, f_j)-trail in G_p, where f_j is a facility with $\ell_j(\mathbf{s}, \sigma(\mathbf{s})) = \lfloor \ell(C_p) \rfloor$.
3. Augment $\sigma(\mathbf{s})$ such that the saturation of all edges in the trail is swapped.

Again, we must show that we can always find a trail in step 2. Assume by contradiction that at some point, there is no trail satisfying the criterion for client $v_i \in V_p$. Let $\tilde{V}_i$ denote the set of clients reachable from v_i by odd-alternating trails. By assumption, each facility in $N_{\mathbf{s}}(\tilde{V}_i) \cup \mathcal{F}_p$ has load $\lceil \ell(C_p) \rceil > \ell(C_p)$. Since these facilities are exclusively considered by clients in $\tilde{V}_i$ for any balanced client subgame profile, the average load on these facilities is strictly larger than $\ell(C_p)$ for any balanced client subgame profile; a contradiction. We conclude that we can always find a trail in step 2.

Next, note that applying step 3 increases the number of assigned clients by one. Since the number of clients is finite, the algorithm always terminates. Furthermore, every application of step 3 increases the load on f_j from $\lfloor \ell(C_p) \rfloor$ to $\lceil \ell(C_p) \rceil$ and does not affect the other facility loads. After applying the algorithm for C_p, the load on every facility in $\mathcal{F}_p$ is therefore either $\lfloor \ell(C_p) \rfloor$ or $\lceil \ell(C_p) \rceil$.

The client subgame profile $\sigma(\mathbf{s})$ obtained by applying the two-phase procedure to every class is thus pure and rounded. Since all covered clients are assigned to some facility in their shopping range, $\sigma(\mathbf{s})$ is also feasible. To show that $\sigma(\mathbf{s})$ is a client equilibrium, we apply Lemma 20. Assume by contradiction that some client $v_i \in V$ considers facility f_j, and that there is another facility $f_{j'} \in N_{\mathbf{s}}(v_i)$ with $\ell_{j'}(\mathbf{s}, \sigma(\mathbf{s})) < \ell_j(\mathbf{s}, \sigma(\mathbf{s})) - 1$. Since all loads are integer, this implies $\ell_{j'}(\mathbf{s}, \sigma(\mathbf{s})) \leq \ell_j(\mathbf{s}, \sigma(\mathbf{s})) - 2$. We find:

$$\lceil \ell_{j'}(\mathbf{s}, \sigma^b(\mathbf{s})) \rceil < \ell_{j'}(\mathbf{s}, \sigma(\mathbf{s})) + 1 \leq \ell_j(\mathbf{s}, \sigma(\mathbf{s})) - 1 < \lfloor \ell_j(\mathbf{s}, \sigma^b(\mathbf{s})) \rfloor,$$

where $\sigma^b(\mathbf{s})$ denotes some balanced client subgame profile. By definition, f_j and $f_{j'}$ are in different tiers and classes, with f_j belonging to the higher tier. As such, v_i cannot consider f_j for $\sigma(\mathbf{s})$ by construction; a contradiction. We conclude that $\sigma(\mathbf{s})$ is a pure, rounded client equilibrium. □

There are generally many ways to decide which facilities get their load rounded up and down respectively. This means that pure, rounded client equilibria are not generally unique and that different client equilibria might induce distinct facility loads. However, Lemma 21 shows that the vector of sorted facility loads is identical for each pure, rounded client subgame profile, and thus for each pure, rounded client equilibrium.

Lemma 21. *Consider some instance of unweighted 2-FLG and let* $\mathbf{s} \in S$ *denote some facility placement profile. Let* $\sigma(\mathbf{s})$ *and* $\sigma'(\mathbf{s})$ *denote two pure, rounded client subgame profiles for the client game induced by* $\mathbf{s}$*. Then,* $\ell_{sort}(\mathbf{s}, \sigma(\mathbf{s})) = \ell_{sort}(\mathbf{s}, \sigma'(\mathbf{s}))$*.*

Proof. Let $\mathcal{T}$ denote the tier set for this client game, and let $V(T_p)$ and $\mathcal{F}(T_p)$ denote the sets of clients and facilities in tier $T_p \in \mathcal{T}$, respectively. Note that each client in $V(T_p)$ considers exactly one facility in $\mathcal{F}(T_p)$ and that the tier set is independent of the chosen client subgame profile. This implies that for every pair of pure, rounded client subgame profiles $\sigma(\mathbf{s}), \sigma'(\mathbf{s})$, we have:

$$\sum_{f_j \in \mathcal{F}(T_p)} \ell_j(\mathbf{s}, \sigma(\mathbf{s})) = \sum_{f_j \in \mathcal{F}(T_p)} \ell_j(\mathbf{s}, \sigma'(\mathbf{s})) \text{ for all } T_p \in \mathcal{T}.$$

This, combined with the definition of rounded client subgame profiles, implies that the number of facilities in $\mathcal{F}(T_p)$ whose load got rounded down (and thus also the number of facilities whose load got rounded up) must be the same for both client subgame profiles. Since this holds for all $T_p \in \mathcal{T}$, the vector of sorted loads is identical for both profiles. □

Unfortunately, ℓ_{sort} is not a lexicographical potential function for facility games induced by (arbitrary) pure, balanced, client equilibria. Cases exist where ℓ_{sort} decreases lexicographically as a consequence of an improving move. However, we can prove that for every facility placement profile $\mathbf{s} \in S$, there exists *a* pure, rounded client equilibrium profile σ such that every improving move away from $\mathbf{s}$ in the facility game induced by σ strictly increases ℓ_{sort} lexicographically. Using this fact and Lemma 6, we can prove the existence of SPE in unweighted instances. The main idea behind this client equilibrium profile is to control which facilities get their load rounded up or down. In particular, we want to prevent the situation where some facility f_j in tier T_p that gets rounded down by for $\mathbf{s}$, can perform an improving move resulting in FPP $\mathbf{s}'$, where f_j stays in the same tier (that is, the index of its tier in the tier set for both profiles is the same), but now gets rounded up, while one or more facilities in T_p get rounded down for $\mathbf{s}'$ while they did not for $\mathbf{s}$. This situation is undesirable, as such a move might cause ℓ_{sort} to remain constant or decrease. To accomplish this, we choose a pure, rounded client equilibrium profile that consistently rounds the same set of facilities up and down over different facility placement profiles. In other words, it *favors* the facilities based on some ordering of the facilities.

Definition 5.5. Consider some instance of 2-FLG with k facility players, and let β denote some permutation of the integers 1 up to k. We define the *β-ordering* of the facilities as the permutation of $\mathcal{F}$, where $f_{\beta(j)}$ is the jth element, for each $j \in [1, k]$. Let $I^\beta : \mathcal{F} \to \mathbb{N}$ denote the function which maps each facility to its position in the *β-ordering* of $\mathcal{F}$. For any state $(\mathbf{s}, \sigma)$, we define the vector of β-ordered facility loads $\ell^\beta_{ord}(\mathbf{s}, \sigma) := \big(\ell_{\beta(1)}(\mathbf{s}, \sigma), \ell_{\beta(2)}(\mathbf{s}, \sigma), \ldots, \ell_{\beta(k)}(\mathbf{s}, \sigma)\big)$.

We now define *favoring* client equilibria: a type of pure, rounded client equilibria that *favors* the facilities according to some ordering β.

Definition 5.6. Consider an instance of 2-FLG with k facility players. Let $\mathbf{s}$ denote some facility placement profile, and β some permutation of the integers from 1 to k. A client equilibrium $\sigma(\mathbf{s})$ is said to be *β-favoring* if it is a pure, rounded, client equilibrium that minimizes the vector of β-ordered facility loads l^β_{ord} lexicographically. Similarly, a client equilibrium profile σ is said to be β-favoring if $\sigma(\mathbf{s}')$ is a β-favoring client equilibrium for every $\mathbf{s}' \in S$.

β-favoring client equilibria are useful for several reasons. Firstly, for any choice of β, a β-favoring equilibrium always exists, since the set of pure, rounded client equilibria is non-empty. Secondly, β-favoring client equilibria are not significantly harder to find than arbitrary pure, rounded client equilibria. We can adapt the algorithm for finding arbitrary pure, rounded client equilibria to one for finding β-favoring ones. We describe the algorithm and its correctness later, as we are only concerned with their existence for now. Lastly, for any facility placement profile $\mathbf{s} \in S$, there exists a permutation β such that for the facility game induced by a β-favoring client equilibrium profile, every improving move away from $\mathbf{s}$ strictly increases ℓ_{sort} lexicographically. To prove this, we define the *assignment graph*, which characterizes any state $(\mathbf{s}, \sigma)$ as a bipartite graph.

Definition 5.7. Consider an instance of 2-FLG and a state $(\mathbf{s}, \sigma)$ with σ pure, and let G' denote the client graph for the client game induced by $\mathbf{s}$. Then, we define the *assignment graph* for $(\mathbf{s}, \sigma)$ as the subgraph of G' with the same vertex set, but which only contains edges $(v_i, f_j) \in E(G')$ if v_i considers f_j for $\sigma(\mathbf{s})$. The *tier assignment graph* for a tier T_p is analogously obtained from the corresponding client tier graph.

Lemma 22. *Consider some instance of 2-FLG with k facility players, some facility placement profile $\boldsymbol{s}$, and some pure, rounded client equilibrium $\sigma(\boldsymbol{s})$. Let β denote a permutation of the integers from 1 to k that satisfies the following two properties:*

1. *for all $f_i, f_j \in \mathcal{F}$ where f_j is in a strictly higher tier than f_i, we have $I^\beta(f_i) < I^\beta(f_j)$*
2. *for all $f_i, f_j \in \mathcal{F}$ with $\ell_i(\boldsymbol{s}, \sigma'(\boldsymbol{s})) < \ell_j(\boldsymbol{s}, \sigma'(\boldsymbol{s}))$, we have $I^\beta(f_i) < I^\beta(f_j)$*

Let σ denote a β-favoring client equilibrium profile, where $\sigma(\boldsymbol{s})$ is the previously defined β-favoring client subgame profile. Then, in the facility game induced by σ, every improving move away from $\boldsymbol{s}$ strictly increases ℓ_{sort} lexicographically.

Proof. Let β denote a permutation of the integers from 1 to k satisfying the conditions of the lemma, and let σ denote a β-favoring client equilibrium profile. We consider the facility game induced by σ and show that every improving move away from $\mathbf{s}$ increases ℓ_{sort} lexicographically. Assume by contradiction that some facility f_{im} can improve by moving to $\mathbf{s}'$, but that this move does not increase ℓ_{sort} lexicographically. Then, there exists some facility player $f_d \in \mathcal{F}$, for which the following hold:

$$\begin{aligned} &\ell_d(\mathbf{s}', \sigma) < \ell_d(\mathbf{s}, \sigma) \\ &\ell_d(\mathbf{s}', \sigma) \leq \ell_{im}(\mathbf{s}, \sigma) \end{aligned} \tag{8}$$

Let $G = (V_G, E_G)$ and $G' = (V_{G'}, E_{G'})$ denote the assignment graphs for $(\mathbf{s}, \sigma)$ and $(\mathbf{s}', \sigma)$ respectively. Consider the bipartite graph $G^* = (V^*, E^*) := (V_G \cup V_{G'}, E_G \triangle E'_G)$, where $\triangle$ denotes the symmetric difference. Our aim is to find an *alternating* (f_d, f_u)-trail Q in G^*, where f_u is a facility player whose load was increased by the move, i.e., $\ell_u(\mathbf{s}', \sigma) > \ell_u(\mathbf{s}, \sigma)$. Here, *alternating* means that all odd-numbered edges belong to E_G and all even-numbered edges belong to $E_{G'}$.

We greedily construct Q by starting at f_d and adding edges from E_G and E'_G alternately until we reach a facility vertex f_u with $\ell_u(\mathbf{s}', \sigma) > \ell_u(\mathbf{s}, \sigma)$. Since the number of edges is finite, the procedure finds a valid trail Q unless it gets stuck at some facility or client vertex (i.e., there is no viable edge to add). This cannot occur at a client vertex, since each client vertex either has degree zero or is incident to exactly one edge from E_G and one edge from $E_{G'}$. The only client vertices that may pose an exception are those exclusively adjacent to f_{im}. However, it is impossible to reach these client vertices without first visiting f_{im}, and the procedure would have terminated upon visiting f_{im} since the load on f_{im} increased by the move. Hence, the procedure cannot "get stuck" at these client vertices either. The same is true for the facility vertices. It cannot occur at f_d, since f_d is incident to strictly more edges from E_G than from $E_{G'}$. Likewise, all facility vertices f_j with $\ell_j(\mathbf{s}', \sigma) \leq \ell_u(\mathbf{s}, \sigma)$ are incident to at least as many edges from E_G as from $E_{G'}$. We conclude that the procedure always finds an alternating (f_d, f_u)-trail with $\ell_u(\mathbf{s}', \sigma) > \ell_u(\mathbf{s}, \sigma)$.

Let $\mathcal{T}'$ denote the tier set for $\mathbf{s}'$ and let F^Q denote the sequence of facilities in the vertex sequence of Q. That is, the subsequence of the vertex sequence of Q containing only the odd-numbered elements. We claim that the facilities in F^Q belong to non-ascending tiers in $\mathcal{T}'$. To see this, assume by contradiction that there exists a pair of facilities f_i, f_j, consecutive in F^Q, such that f_j belongs to a higher tier in $\mathcal{T}'$ than f_i. Then, the client positioned between these two facilities in the vertex sequence of Q considers a facility that does not belong to the minimal tier among the facilities in its range. As clients always belong to the same tier as the lowest-tiered facility in their shopping range, it follows that there is a client considering a facility outside their tier. We conclude that $\sigma(\mathbf{s}')$ is not rounded; a contradiction.

Next, we consider two distinct cases and show that in either case, all facilities in F^Q belong to the same tier in $\mathcal{T}'$. In the first case, we assume that f_{im} is not in F^Q. Then, an analogous argument to the earlier one proves that the facilities in F^Q belong to non-descending tiers in $\mathcal{T}$, where $\mathcal{T}$ denotes the tier set for $\mathbf{s}$. Next, note that the move decreased and increased the loads on f_d and f_u respectively. As such, f_d cannot be in a strictly higher tier in $\mathcal{T}'$ than in $\mathcal{T}$, and f_u cannot be in a strictly lower tier in $\mathcal{T}'$ than in $\mathcal{T}$. It follows that all facilities in F^Q belong to the same tier in $\mathcal{T}'$.

In the second case, we assume that f_{im} is in F^Q. That is, $f_{im} = f_u$. As the location of f_{im} is not the same between $\mathbf{s}$ and $\mathbf{s}'$, we cannot use the same argument as before. However, by assumption, $\ell_d(\mathbf{s}', \sigma) \leq \ell_{im}(\mathbf{s}, \sigma) < \ell_{im}(\mathbf{s}', \sigma)$. This implies that f_{im} is not in a strictly lower tier in $\mathcal{T}$ than f_d. Since the facilities in F^Q are in non-ascending tiers in $\mathcal{T}'$, we conclude that they are all in the same tier in $\mathcal{T}'$. In either case:

$$\ell_d(\mathbf{s}', \sigma) = \ell_u(\mathbf{s}', \sigma) - 1. \tag{9}$$

This equation, the integrality of the facility loads, and the fact that moving to $\mathbf{s}'$ decreased and increased the loads on f_d and f_u respectively, prove that $\ell_d(\mathbf{s}, \sigma) > \ell_u(\mathbf{s}, \sigma)$. By the choice of β, we conclude that $I^\beta(f_d) > I^\beta(f_u)$

Next, consider the assignment graph obtained from G' by replacing the edges in $Q \cap E_{G'}$ with the edges in $Q \cap E_G$. Let $\sigma^*(\mathbf{s}')$ denote the pure client equilibrium characterized by this assignment graph. We find that:

$$\ell_j(\mathbf{s}', \sigma^*(\mathbf{s}')) = \begin{cases} \ell_j(\mathbf{s}', \sigma(\mathbf{s}')) & \text{for } j \notin \{d, u\} \\ \ell_j(\mathbf{s}', \sigma(\mathbf{s}')) + 1 & \text{for } j = d \\ \ell_j(\mathbf{s}', \sigma(\mathbf{s}')) - 1 & \text{for } j = u. \end{cases} \tag{10}$$

(9) and (10) show that the only difference between $\sigma(\mathbf{s}')$ and $\sigma^*(\mathbf{s}')$ is that the loads of f_u and f_d are swapped. Since all facilities in F^Q are in the same tier of $\mathcal{T}'$, $\sigma^*(\mathbf{s})$ is a pure, rounded client equilibrium. Since $I^\beta(f_d) > I^\beta(f_u)$, it follows that $l^\beta_{ord}(\mathbf{s}', \sigma^*(\mathbf{s}'))$ is bigger than $l^\beta_{ord}(\mathbf{s}', \sigma(\mathbf{s}'))$ lexicographically. We conclude that $\sigma(\mathbf{s}')$ is not β-favoring; a contradiction.

Thus, f_d cannot exist. We conclude that $\ell_{sort}(\mathbf{s}', \sigma)$ is lexicographically strictly larger than $\ell_{sort}(\mathbf{s}, \sigma)$. □

Lemma 22 only holds for improving moves away from $\mathbf{s}$. If some facility improves by moving to $\mathbf{s}'$ from $\mathbf{s}$, ℓ_{sort} increases lexicographically. However, this may greatly affect the facility loads, thereby invalidating the condition on the ordering of the facility loads with respect to β. Consequently, further improving moves do not generally increase ℓ_{sort} lexicographically.

To address this, we consider a different client equilibrium profile following each improving move. This way, we may continue to apply Lemma 22. The key insight that makes this work is Lemma 21. This lemma shows that switching to a different pure, rounded client equilibrium does not affect the progress made in increasing ℓ_{sort}.

Theorem 12. *Every instance of unweighted 2-FLG admits a subgame perfect equilibrium. More so, it always admits one with a pure client equilibrium profile.*

Proof. Consider an instance of unweighted 2-FLG. Let k denote the number of facility players and let S denote the set of facility placement profiles. We provide an algorithm for finding a subgame perfect equilibrium:

1. Choose an arbitrary facility placement profile $\mathbf{s}^1 \in S$ and set $i = 1$. i will denote the iteration of the algorithm.

2. Find some pure, rounded client equilibrium $\sigma^1(\mathbf{s}^1)$ for the client game induced by $\mathbf{s}^1$.

3. Find some permutation β^i of the integers from 1 to k that satisfies the conditions of Lemma 22 for $\sigma^i(\mathbf{s})$ and find a β^i-favoring client equilibrium profile σ^i. Note that σ^i should "include" the previously defined $\sigma^i(\mathbf{s})$, as this client subgame profile is β^i-favoring.

4. Consider the facility game induced by σ^i. If $\mathbf{s}^i$ is a facility equilibrium for this game, then terminate with solution $(\mathbf{s}^i, \sigma^i)$. Else, let $\mathbf{s}^{i+1}$ denote the facility placement resulting from some improving move.

5. Set $\sigma^{i+1}(\mathbf{s}^{i+1}) = \sigma^i(\mathbf{s}^{i+1})$. Increment i by one and return to step 3.

This algorithm returns a subgame perfect equilibrium whenever it terminates. Furthermore, it cannot get stuck at any step, as pure, rounded client equilibria always exist. To prove that it always terminates, it thus suffices to show that the number of iterations is finite. To this end, consider how the vector of sorted facility loads changes over the iterations. For any $i \in \mathbb{N}$, we apply Lemma 22 to find:

$$\ell_{sort}(\mathbf{s}^{i+1}, \sigma^i) > \ell_{sort}(\mathbf{s}^i, \sigma^i), \tag{11}$$

where ">" denotes the lexicographical relation. Furthermore, since σ^i is a pure, rounded client equilibrium for every iteration $i \in \mathcal{N}$, we apply Lemma 21 to find that:

$$\ell_{sort}(\mathbf{s}^j, \sigma^i) = \ell_{sort}(\mathbf{s}^j, \sigma^1), \tag{12}$$

for every $j \in \mathbb{N}$. Note that there is nothing special about σ^1; any pure, rounded client equilibrium profile works as a reference point to establish the relations between other profiles. Applying (11) and (12) with $j = i + 1$ yields:

$$\ell_{sort}(\mathbf{s}^{i+1}, \sigma^1) = \ell_{sort}(\mathbf{s}^{i+1}, \sigma^i) > \ell_{sort}(\mathbf{s}^i, \sigma^i) = \ell_{sort}(\mathbf{s}^i, \sigma^1).$$

Applying transitivity, we find that for $i, j \in \mathcal{N}$ with $j > i$:

$$\ell_{sort}(\mathbf{s}^j, \sigma^1) > \ell_{sort}(\mathbf{s}^i, \sigma^1),$$

which implies $\mathbf{s}^j \neq \mathbf{s}^i$. We conclude that all facility placement profiles visited by the algorithm are unique. Since the number of facility placement profiles $|S| \leq |V|^k$ is finite, the algorithm always terminates in a finite number of iterations. The algorithm finds a subgame perfect equilibrium $(\mathbf{s}, \sigma)$ for every instance of unweighted 2-FLG. Moreover, the found σ is pure (and rounded). We conclude that every instance of unweighted 2-FLG admits a subgame perfect equilibrium $(\mathbf{s}, \sigma)$ where σ is a pure client equilibrium profile. □

The existence of SPE for unweighted instances is our main result for this chapter and one of the main results of this thesis. Theorem 12 is useful for more than just this existence result, however. Since the proof is constructive, it provides an outline of an algorithm for finding subgame perfect equilibria for unweighted instances. The only things that are yet unclear are the procedure for finding $\beta-$favoring client equilibria and the running time of the algorithm.

5.2 Finding Equilibria in the Unweighted Game

We first analyze the complexity of finding balanced client subgame profiles.

Lemma 23. *Consider an instance of 2-FLG with k facility players and a facility placement profile $\mathbf{s} \in S$. Let β denote some permutation of the integers from 1 to k. If the class set $\mathcal{C}$ for the client game induced by $\mathbf{s}$ is known, we can find a β-favoring client equilibrium $\sigma(\mathbf{s})$ in polynomial time with respect to the input size.*

Proof. We describe a procedure for finding a β-favoring client equilibria, establish its correctness, and discuss its running time.

We adapt the procedure outlined in the proof of Lemma 11. First, we execute only the first phase of the procedure for all classes. At this point, each facility receives a load equal to their balanced load rounded down. Subsequently, we execute the second phase per tier, instead of per class. To ensure that we find the pure, rounded client equilibrium $\sigma(\mathbf{s})$ that maximizes $\ell^{\beta}_{ord}(\mathbf{s}, \sigma(\mathbf{s}))$, we select a specific facility f_j in step 2 of the second phase, instead of an arbitrary one satisfying the criteria. In particular, we determine the set facilities reachable via an odd-alternating trail in the client tier graph. From this set, we select the facility with the smallest index in β.

We denote the client equilibrium obtained from this augmented procedure with $\sigma(\mathbf{s})$. By the same arguments as for the original algorithm, $\sigma(\mathbf{s})$ is pure and rounded. To prove that it is β-favoring, we assume by contradiction that it is not. This implies the existence of a β-favoring client equilibrium $\sigma'(\mathbf{s})$.

Let f_j denote the facility with the smallest index in β that has its load rounded down by $\sigma(\mathbf{s})$ but rounded up by $\sigma'(\mathbf{s})$. The existence of such a facility is guaranteed by our assumptions. Let T_p denote the tier containing f_j and let G_p and G'_p denote the tier assignment graphs (see Definition 5.4) for tier T_p for $\sigma(\mathbf{s})$ and $\sigma'(\mathbf{s})$, respectively. Consider bipartite graph $G^* = (V^*, E^*) := (V(G), E(G) \Delta E(G'))$, where Δ denotes the symmetric difference. We greedily construct a trail Q starting from f_j where all odd-indexed edges of Q belong to $E(G')$ and all even-indexed edges belong to $E(G)$, until we reach some facility that gets rounded down by $\sigma'(\mathbf{s})$, but not by $\sigma(\mathbf{s})$. This construction always yields the desired trail, since it is always possible to add an edge unless some facility satisfying the criterion has first been reached. Let f_x denote the facility vertex at the end of this trail. Since $\ell^{\beta}_{ord}(\mathbf{s}, \sigma'(\mathbf{s}))$ is lexicographically larger than $\ell^{\beta}_{ord}(\mathbf{s}, \sigma(\mathbf{s}))$, it follows that $I^{\beta}(f_x) > I^{\beta}(f_j)$.

Next, let v_i denote the client assigned in the iteration of the second phase that increased the load on f_x. Let Q' denote the odd-alternating (v_i, f_x) trail used to assign v_i in the algorithm. Note that Q is an alternating trail in the client tier graph for the iteration which assigned v_i. Thus, there was an odd-alternating (v_i, f_j)-trail in this graph: either $Q + Q'$ or just the parts of Q' and Q until the first edge contained in both trails (this would be the "start" of Q' and the "end" of Q'). Since $I^\beta(f_x) > I^\beta(f_j)$, the algorithm would have increased the load on f_j instead of f_x; a contradiction. We conclude that $\sigma'(\mathbf{s})$ cannot exist and that $\sigma(\mathbf{s})$ is β-favoring.

Each iteration (in either phase) of this algorithm assigns one additional client. As such, the total number of iterations equals the number of clients n. The running time of an iteration of the algorithm is dominated by finding the alternating trails, which can be done using a breadth-first search in the client class/tier graph. Since the number of edges in these graphs is at most kn, each iteration takes $O(k+n+kn) = O(kn)$ time. Assuming that the class set is known, a β-favoring client equilibrium can thus be found in $O(kn^2)$ time, which is polynomial in the input size. □

Finding favoring client equilibria is thus possible in polynomial time. However, it is not possible to find a subgame perfect equilibrium $(\mathbf{s}, \sigma)$ in polynomial time since σ consists of an exponential number of client subgame profiles. In practice, we try to find only $\mathbf{s}$, and $\sigma(\mathbf{s}')$ for those facility placement profiles $\mathbf{s}'$ that differ in at most one element from $\mathbf{s}$. This "partial" client equilibrium profile is sufficient to show that $\mathbf{s}$ is a facility equilibrium for the facility game induced by any extension of this partial client equilibrium profile to a true one. Similarly, the client subgame profiles for "local" facility placement profiles are all that is needed to find improving moves in the algorithm described in the proof of Theorem 12. We claim that, if we only find the necessary client equilibria in step 3 of the procedure, each iteration of this algorithm takes only polynomial time.

Lemma 24. *Each iteration of the algorithm discussed in the proof of Theorem 12 can be done in time polynomial in the input size.*

Proof. Consider an iteration $i \in N$ of the algorithm. First, β^i can be found in $O(k \log(k))$ by simply sorting the facilities based on their known loads and the known tier set. To check whether improving moves exist, we find β-favoring client equilibria for the $k(n-1)+1 \leq kn$ facility placement profiles that differ from $\mathbf{s}^i$ in at most one element. It thus suffices to find a polynomial number of β-favoring client equilibria. By Theorem 10 and Lemma 23, finding the class set and then using this to find a β-favoring client equilibrium can be done in polynomial time. It follows that iteration of step 3 can similarly be done in time polynomial in the input size. In step 4, we check for each of the $k(n-1)$ unilateral moves of the facilities, whether the load on the moving facility increases. Computing the load on a facility for some state can be done in polynomial time. We conclude each iteration can be done in time polynomial in the input size. □

Since the initialization of the algorithm also takes only polynomial time, we can find subgame perfect equilibria for unweighted instances if the number of iterations of the algorithm is polynomially bounded. Note that, since the facility placement profiles corresponding to each iteration are unique, the number of iterations is bounded by $|S| \leq k^n$, which is exponential in the input size. Unfortunately, we were unable to prove that the number of iterations is polynomially bounded, and neither were we able to construct an instance that required an exponential number of iterations. It is currently unknown whether one can find a subgame perfect equilibrium in uweighted 2-FLG in time polynomial in the input size.

5.3 Types of Games that May Admit No Subgame Perfect Equilibrium

So far, we have found two sufficient conditions for the existence of SPE. However, neither of these conditions is *necessary* for the existence of SPE. As such, there might be further sufficient conditions for their existence. We discuss some types of games that do not generally admit subgame perfect equilibria, although one may expect them to. The first of these is the class of (unrestricted) 2-FLG where the host graph is a line (i.e., all vertices have in- and outdegree at most one).

Lemma 25. *There are instances (H, k) of unrestricted 2-FLG, with $H := (V, A, w)$ s.t. $\delta^+(v), \delta^-(v) \leq 1\ \forall v \in V$, that do not admit a subgame perfect equilibrium.*

Proof. We give an example of such an unrestricted instance of 2-FLG. Let H denote the host graph shown in Figure 14, which satisfies the degree constraints. We claim that $(H, 2)$ does not admit a subgame perfect equilibrium.

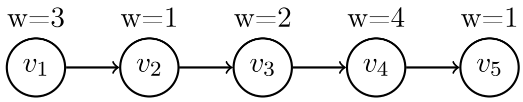


Figure 14: Host graph H for counterexample

The proof requires listing all twenty-five facility placement profiles in S and showing that none of them correspond to an SPE (see the proof of Lemma 1 for an example of this procedure). We omit this proof for the sake of brevity, and only provide some intuition. Note that for any FPP where the facilities choose different locations, a unique client equilibrium exists. The main idea behind the construction of this counterexample is that any sequence of improving moves ends up in the cycle $C := \Big((v_2, v_4), (v_5, v_4), (v_5, v_2), (v_4, v_2), \dots\Big)$. For the FPPs where the two facilities are at the same location, the facility receiving the lesser (or equal) load can improve by moving to either v_2, v_4 or v_5, which again leads to cycle C. □

Another idea is to restrict the client weights. After all, if all weights are equal, the existence of an SPE is guaranteed. Since uniformly scaling weights does not affect the existence of SPE, we may assume that 1 is the smallest weight in any graph. It turns out that even the unrestricted instances where the cient weights are restricted to two distinct values (1 and some $\beta > 1$), do not generally admit an SPE.

Lemma 26. *For every rational number β strictly larger than 1, there exists an instance of unrestricted 2-FLG where all client weights are either 1 or β that admits no subgame perfect equilibrium.*

Proof. We give three families of examples depending on the value of β. If $\beta \in (1, 2)$, let H denote the host graph shown in Figure 15. We claim that $(H, 2)$ does not admit a subgame perfect equilibrium.

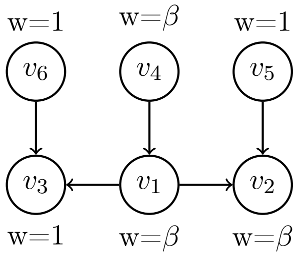


Figure 15: Host graph without SPE for $\beta \in (1, 2)$

If $\beta = 2$, let H denote the host graph shown in Figure 16. We claim that $(H, 2)$ does not admit a subgame perfect equilibrium.

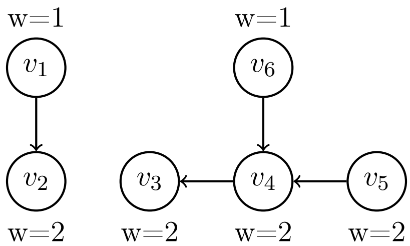


Figure 16: Host graph without SPE

If $\beta > 2$, let H denote the host graph shown in Figure 17. We claim that $(H, 2)$ does not admit a subgame perfect equilibrium.

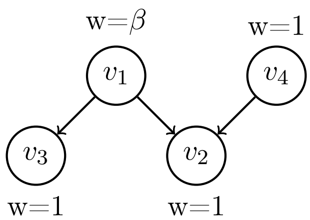


Figure 17: Host graph without SPE for $\beta > 2$

Again, proving that these instances admit no SPE is done by listing all possible facility placement profiles and verifying that none of them correspond to an SPE. We omit this as this process is time-consuming and uninteresting. □

There is one final, somewhat interesting class of 2-FLG for which the existence of SPE might be guaranteed: the class of instances where the number of facility players equals or exceeds the number of client players. We conjecture that such instances always admit SPE. Our brief efforts to prove this conjecture did not yield a proof, however. We did not consider this problem to be of sufficient interest to pursue further.

6 Efficiency of Equilibria

We have shown that certain classes of instances of 2-FLG admit subgame perfect equilibria. The most important of these is the class of unweighted instances. In the following, we analyze the efficiency of subgame perfect equilibria by considering the ratio between the *social payoff* of specific subgame perfect equilibria and the maximum attainable social payoff over all states. Here, the social payoff refers to the sum of all facility loads. That is, the sum of the weights of the set of clients covered by the facilities. We provide bounds on the *price of anarchy* and the *price of stability* for 2-FLG. The price of stability is the ratio between the maximum attainable social payoff over all states and the maximum attainable social payoff over the set of subgame perfect equilibria. The price of anarchy is the ratio between the maximum attainable social payoff over all states and the minimum attainable social payoff over the set of subgame perfect equilibria.

Theorem 13. *For the class of problems in 2-FLG that admit subgame perfect equilibria, the price of anarchy with respect to the facility players is 1 for $k = 1$, and 2 for $k \geq 2$. This result also holds for the class of unrestricted instances, the class of unweighted instances, and their intersection.*

Proof. As stated, the social payoff is simply the weight of the covered clients. For $k = 1$, any SPE maximizes this value, so the price of anarchy is 1. Next, we consider $k \geq 2$. In their paper, Krogmann et al. [17] showed that for arbitrary client behavior, the price of anarchy is at most 2. However, they consider the unrestricted version of the two-stage facility location game. As such, it remains to be shown that this bound also holds for the more general version of 2-FLG considered in this thesis. To see that it does, consider some instance $(H, \mathcal{U}, k)$ of 2-FLG with $k \geq 2$, and let $(\mathbf{s}, \sigma)$ denote some subgame perfect equilibrium.

We say a set of uncovered client vertices $V' \subseteq V$ is a f_j*-cluster* if they can all be covered by facility f_j at the same time. That is, if there exists a vertex $v_i \in \mathcal{U}(f_j)$ such that for all $v \in V'$: $v = v_i$ or $(v, v_i) \in A$. Let $w(V')$ denote the weight of V', which we define as the sum of the weights of all clients in V'. Since $\mathbf{s}$ is a facility equilibrium, the maximum weight of an f_j-cluster is $\ell_j(\mathbf{s}, \sigma)$. After all, if an f_j-cluster V' with larger weight were to exist, then f_j could improve by moving to a vertex that covers all clients in V'. f_j would then be the only facility in the attraction range of the clients in V', meaning that all these clients must exclusively consider f_j after the move.

For each $f_j \in \mathcal{F}$, let V^j denote an f_j-cluster of maximal weight, and let $V^{un}(s)$ denote the set of clients not covered by $\mathbf{s}$. Then the total weight of clients in $V^{un}(s)$ covered by any facility placement profile is at most $\sum_{f_j \in \mathcal{F}} w(V^j)$. There could be an FPP that covers the maximum number of clients in $V^{un}(s)$, while still covering all clients in $V^{cov}(\mathbf{s})$. As such, we can bound the social optimum p_{opt} as follows:

$$p_{opt} \leq \sum_{f_j \in \mathcal{F}} \ell_j(\mathbf{s}, \sigma) + w(V^j) \leq \sum_{f_j \in \mathcal{F}} \ell_j(\mathbf{s}, \sigma) + \ell_j(\mathbf{s}, \sigma) = 2p_{\mathbf{s}},$$

where $p_{\mathbf{s}}$ denotes the social payoff in the subgame perfect equilibrium with FPP $\mathbf{s}$. We conclude that the price of anarchy is at most 2.

Next, we show that this bound is tight for every $k \geq 2$, even for unrestricted, unweighted 2-FLG. To this end, consider instance (H, k) of unrestricted, unweighted 2-FLG with $k \geq 2$, where $H = (V, A)$ is the host graph with:

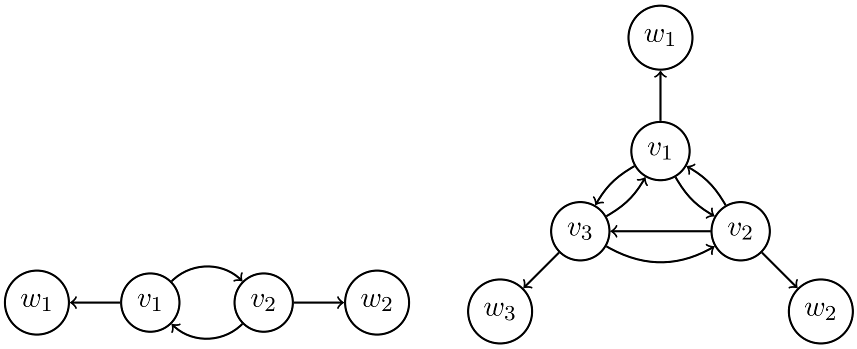
$$V = \{v_1, v_2, \ldots, v_k, w_1, w_2, \ldots, w_k\}$$
$$A = \{(v_i, v_j) \mid i, j \leq k, i \neq j, \} \cup \{v_i, w_i \mid i \leq k\}$$

As examples, the host graphs for $k = 2$ and $k = 3$ are shown in Figure 18.

Consider facility placement profiles $\mathbf{s}_{opt} := (w_1, w_2, \ldots, w_k)$ and $\mathbf{s} = (v_1, v_2, \ldots, v_k)$. Note that $\mathbf{s}_{opt}$ covers all clients, and is thus a social optimum profile. On the other hand, $\mathbf{s}$ covers exactly half as many clients. To prove the tightness of the bound, it suffices to find a client equilibrium profile σ such that $(\mathbf{s}, \sigma)$ is an SPE. To construct such a client equilibrium profile, we find some pure client equilibrium $\sigma(\mathbf{s})$ that induces a load of 1 on all facilities. Next, we consider facility placement profiles that differ from $\mathbf{s}$ in exactly one element. Let $\mathbf{s}'$ denote the facility profile resulting from a move by facility player f_j. If $\mathbf{s}'_j = v_i$ for some $i \leq k$, we set $\sigma(\mathbf{s}') = \sigma(\mathbf{s})$, so $\ell_j(\mathbf{s}, \sigma) = \ell_j(\mathbf{s}', \sigma)$.

If $s'_j = w_i$ for some $i \leq k$, we set $\sigma(\mathbf{s}')$ to some pure client subgame profile where w_i patronizes f_j, v_p patronizes f_x for all $x \neq j$, and v_j patronizes some facility f_y with $y \neq j$. For this choice, $\sigma(\mathbf{s}'$ is a client equilibrium and $\ell_j(\mathbf{s}, \sigma) = 1 = \ell_j(\mathbf{s}', \sigma)$.

For any FPP $\mathbf{s}^*$ that differs from $\mathbf{s}$ in two or more elements, we set $(\sigma(\mathbf{s}))$ to some arbitrary client equilibrium for the client game induced by $\mathbf{s}^*$. These client subgame profiles do not affect whether $\mathbf{s}$ is a facility equilibrium. The constructed client profile σ is a client equilibrium profile, and none of the moves away from $\mathbf{s}$ in the facility game induced by σ are improving moves. We conclude that $(\mathbf{s}, \sigma)$ is a subgame perfect equilibrium and that the upper bound of 2 on the price of anarchy is tight.



(a) Graph for k=2 (b) Graph for k=3

Figure 18: Example graphs for small number of facilities

□

Theorem 14. *For the class of problems in 2-FLG that admit subgame perfect equilibria, the price of stability is at least* $2 - \frac{1}{k}$. *This result also holds for unrestricted 2-FLG, unweighted 2-FLG, and their intersection.*

Proof. This lower bound was established by Krogmann et al. [17] for arbitrary client behavior. We refer to their paper for the proof. Since they use an unweighted and unrestricted instance in their proof, this lower bound on the price of stability also holds for unweighted, unrestricted 2-FLG. □

7 Complexity of the Decision Problem

We wish to show that the problem of deciding whether an instance of 2-FLG admits an SPE is NP-complete. We first show that given the right certificate, we can verify that an SPE exists in polynomial time, which proves that the decision problem is in NP. Note that since the size S is exponential in n and k, so is any client (equilibrium) profile. Therefore, we cannot use an SPE $(\mathbf{s}, \sigma)$ as a (polynomial size) certificate, nor can we verify that σ is a client equilibrium profile in time polynomial in the input size. However, it suffices to show that an SPE exists; we do not need to find one.

Theorem 15. *The problem of deciding whether an instance of 2-FLG admits an SPE is in NP.*

Proof. Consider an instance of 2-FLG with SPE $(\mathbf{s}, \sigma)$. Let $S' \subseteq S$ denote the set of facility placement profiles that differ in at most one element from $\mathbf{s}$. Then, $|S'| \leq k(n-1)+1$, which is polynomial in the number of players. Let $\sigma(S') := \big(\sigma(\mathbf{s}')\big)_{\mathbf{s}' \in S'}$ denote the part of σ defined on S'. As a (polynomial size) certificate, we use $(\mathbf{s}, \sigma(S'))$. We can verify that $\mathbf{s}$ is a facility equilibrium for the game induced by σ knowing only the client subgame profiles $\sigma(\mathbf{s}')$ for $\mathbf{s}' \in S'$. Additionally, since a client equilibrium exists for every client game, it is always possible to extend a "partial" client equilibrium profile $\sigma(S')$ to a "full" client equilibrium profile σ. To prove that an SPE exists, it thus suffices to show the following:

1. $\mathbf{s} \in S$ (i.e., $\mathbf{s}$ is pure and feasible)
2. $\mathbf{s}$ is a facility equilibrium for the facility game induced by any $\sigma \in \Phi$ with $\sigma(S')$ as given by the certificate.
3. For $\mathbf{s}' \in S'$, $\sigma(\mathbf{s}')$ is feasible and a client equilibrium for the client game induced by $\mathbf{s}'$.

Verifying the first condition takes $O(nk)$ time. To verify the second, we consider all unilateral moves and confirm that these do not increase the load on the moving facility. Note that the facility placement profiles resulting from such moves are all in S'. Since finding the facility loads takes $O(nk)$ time, we can verify that $\mathbf{s}$ is a facility equilibrium in $O(n^2k^2)$ time. Verifying the feasibility of a client subgame profile is done in polynomial time using the definition of feasibility. We use Theorem 1 to verify that a client subgame profile is a client equilibrium. To apply the theorem, we compute all v_i-excluded facility loads for each client $v_i \in V$. This process takes $O(nk)$ time for each client subgame profile. Since the number of client subgame profiles in S' is $|S'| = k(n-1)$, we can verify the third condition in time $O(n^3k^2)$.

Verifying all three conditions can thus be done in $O(n^3k^2)$ time, which is polynomial in the input size. We conclude that the decision problem is in NP. □

Next, we show that the decision problem is also NP-hard. We accomplish this using a polynomial time reduction from the boolean satisfiability problem.

Theorem 16. *The problem of deciding whether an instance of 2-FLG admits an SPE is NP-hard. This is even true for unrestricted 2-FLG.*

Proof. To show that the decision problem is NP-hard, it suffices to find a polynomial time reduction from the decision problem to an NP-hard problem of our choice. In this proof, we use the boolean satisfiability problem (SAT), which was shown to be NP-hard by Cook [6]. SAT is the following problem: Given a set of m boolean variables $\mathbf{x} = \{x_1, x_2, \ldots, x_m\}$

and t clauses $C_1, C_2, \ldots, C_t$ (disjunctions on the variables of $\mathbf{x}$ and their negations), decide whether there exists an assignment of the variables $\mathbf{x}$ that satisfies $C_1 \wedge \ldots \wedge C_t$.

Assume that we are given an instance of SAT, consisting of a set of clauses $C_1, \ldots, C_t$, on a set of m boolean variables $\mathbf{x}$. Without loss of generality, we assume $t \geq 4$. We construct an instance of unrestricted 2-FLG with $k = m + 1$ and host graph H. Here, H consists of components $G_1 = (V_1, A_1, w_1)$ and $G_2 = (V_2, A_2, w_2)$. The structure of G_1 is dependent on the given instance of SAT:

- $V_1 := \{v_1, \ldots, v_m, w_1, \ldots, w_m, c_1, \ldots, c_t, b_1, \ldots, b_{(m-1)\cdot t}\}$,
- $w_1(e) = 1 \; \forall e \in A_1$,
- A_1 contains the following edges:

$$\begin{aligned}
&(w_i, v_i) \text{ for } i = 1, \ldots, m \\
&(v_i, w_i) \text{ for } i = 1, \ldots, m \\
&(b_j, v_i) \text{ for } i = 1, \ldots, m \text{ and } j = 1, \ldots, (m-1)\cdot t \\
&(b_j, w_i) \text{ for } i = 1, \ldots, m \text{ and } j = 1, \ldots, (m-1)\cdot t \\
&(c_j, v_i) \text{ for } i = 1, \ldots, m \text{ and all } j \text{ s.t. } x_i \in C_j \\
&(c_j, w_i) \text{ for } i = 1, \ldots, m \text{ and all } j \text{ s.t. } \neg x_i \in C_j.
\end{aligned}$$

Component G_2 is the graph shown in figure 19.

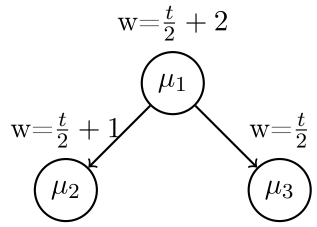


Figure 19: Component G_2

We show that (H, k) admits a subgame perfect equilibrium if and only if there exists a satisfying assignment to the variables of $\mathbf{x}$. Assume that there exists a satisfying assignment to the variables of $\mathbf{x}$, and let $\mathbf{y}$ be one such assignment. We consider facility placement profile $\mathbf{s} \in S$ with

$$s_j = \begin{cases} v_j & \text{if } j \leq m \text{ and } y_j \text{ is true} \\ w_j & \text{if } j \leq m \text{ and } y_j \text{ is false} \\ \mu_2 & \text{if } j = m+1. \end{cases}$$

By construction, $\mathbf{s}$ covers all clients in V_1. We claim that a pure, balanced client equilibrium $\sigma(\mathbf{s})$ exists for the client game induced by $\mathbf{s}$, and show how to construct one. We start by assigning each client in $\{w_1, \ldots, w_m, v_1, \ldots, v_m, c_1, \ldots, c_t\}$ to some arbitrary facility in their shopping range. After this step, the maximum load on any facility f_j is $t+2$; it is $t+2$ if clients v_j, w_j and c_1 up to c_t were all assigned to f_j. Next, we assign the clients in $\{b_1, \ldots, b_{(m-1)\cdot t}\}$ to facilities in such a way that all facility loads, except that of f_{m+1}, become $t+2$. Since each client in $\{b_1, \ldots, b_{(m-1)\cdot t}\}$ can reach all facilities except f_{m+1}, it is trivial to achieve this condition. Finally, we assign μ_1 and μ_2 to f_{m+1}. The client subgame profile constructed this way is pure, balanced, and a client equilibrium.

Let σ denote some pure client equilibrium profile with client equilibrium $\sigma(\mathbf{s})$ balanced. We claim that $(\mathbf{s}, \sigma)$ is an SPE. To prove this, it suffices to show that $\mathbf{s}$ is a facility equilibrium for the facility game induced by σ. First note that since $\sigma(\mathbf{s})$ is balanced, $\ell_j(\mathbf{s}, \sigma(\mathbf{s})) = t+2$ for $j \leq m$ and $\ell_{m+1}(\mathbf{s}, \sigma(\mathbf{s})) = t+3$, since all loads are equal to those induced by the previously constructed balanced client equilibrium. We show that for the facility game induced by σ, no facilities can improve by moving away from $\mathbf{s}$.

- A facility $f_j \in \mathcal{F}$ cannot improve by moving to b_i or c_i for $i \in \mathbb{N}$, since the reach of these vertices is 1, which is less than the current load of all facilities.
- A facility $f_j \in \mathcal{F}$ with $j \leq m$ cannot improve by moving to a vertex in V_2, as this results in a load of at most $t+2$ for the moving facility, which equals their load before the move. f_{m+1} also cannot improve by moving to either μ_1 or μ_3.
- A facility $f_j \in \mathcal{F}$ cannot improve by moving to v_i or w_i for $i \in \mathbb{N}$. Assume by contradiction that f_j can improve by moving to v_i or w_i. Since σ is pure, their new load is at least $t+3$. It follows that some client $b_x \in \{b_1, \ldots, b_{(m-1)\cdot t}\}$ considers f_j after the move. Since $|V_1| = m(t+2)$ and there are at least m facilities located on V_1, there must be some facility located on V_1 with load at most $t+1$. Since b_x can reach this facility, σ is not a client equilibrium profile; a contradiction. Thus, the improving move cannot exist.

We conclude that $\mathbf{s}$ is a facility equilibrium for the facility game induced by σ and that $(\mathbf{s}, \sigma)$ is an SPE.

Now assume that no assignment to the variables of $\mathbf{x}$ is satisfying. Assume by contradiction that (H, k) admits an SPE $(\mathbf{s}, \sigma)$, and let χ denote the number of facilities located on V_2 for $\mathbf{s}$. We show that for each possible value of χ, $\mathbf{s}$ is not a facility equilibrium for the facility game induced by σ.

- $\chi = 0$. Since all $m+1$ facilities are located on V_1, the sum of the facility loads is at most $|V_1| = m(t+2)$. It follows that there is a facility f_j with load strictly less than $t+2$. f_j can improve their load to $t+3$ by moving to μ_2.
- $\chi = 1$. The one facility located on V_2 is located on μ_2. Otherwise, this facility can improve by moving to μ_2, meaning that $\mathbf{s}$ is not a facility equilibrium. Now assume that none of the facilities located on V_1 can improve by moving to μ_3, meaning that the load on each of these facilities is at least $t+2$. It follows that all clients in V_1 are covered, and thus that there is a facility located on v_i or w_i for every $i \in \{1, 2, \ldots, m\}$. As there are exactly m facilities on V_1, this means that there is no i for which both w_i and v_i are used as facility locations. Consider assignment $\mathbf{y}$ of the variables of $\mathbf{x}$ where x_i is *True* if and only v_i is used as a facility location. Since clients $c_1, \ldots c_t$ are all covered by $\mathbf{s}$, $\mathbf{y}$ must satisfy $C_1 \wedge \ldots \wedge C_t$; a contradiction. We conclude that some facility located on V_1 is able to improve by moving to μ_3, and thus that $\mathbf{s}$ is not a facility equilibrium.
- $\chi = 2$. One of the facilities on V_2 can improve by moving to another vertex in V_2, as shown in the proof of Lemma 1. Thus, $\mathbf{s}$ is not a facility equilibrium.
- $\chi \geq 3$. The sum of the loads of the facilities located on V_2 is at most $1.5t+3$. Thus, at least one of these facilities has load at most $0.5t+1 \leq t-1$ (recall the assumption $t \geq 4$). Let f_j denote one such facility. We claim that f_j can improve by moving to v_1, where it will get a load strictly larger than $t-1$. Let $\mathbf{s}'$ denote

the facility placement profile resulting from this move. Assume by contradiction that $\ell_j(\mathbf{s}', \sigma) \leq t - 1$. Then, there exists a client γ in $\{v_1, w_1, b_1, b_2, \ldots, b_{(m-1)\cdot t}\}$ that considers a facility with load strictly larger than t, since all clients in this set are covered and exclusively consider the $m - 1$ facilities located on V_1. Since f_j is in the shopping range of γ, the assumption that the load on f_j is at most $t - 1$ contradicts Theorem 1. We conclude that $\sigma(\mathbf{s}')$ is not a client equilibrium; a contradiction. Thus, f_j can improve by moving to v_1.

Since we made no assumptions on σ besides it being a client equilibrium profile, we have shown that (H, k) does not admit an SPE. □

The problem of deciding whether an instance of the two-stage facility location game admits a subgame perfect equilibrium is both NP-hard and in NP. We conclude that the decision problem is NP-complete.

8 Approximate Equilibria

The existence of a subgame perfect equilibrium is not guaranteed for general instances of 2-FLG. This naturally raises questions about the existence of approximate (subgame perfect) equilibria. In this chapter, we define approximate equilibria in the context of the two-stage facility location game and discuss their existence.

Definition 8.1. Consider an instance of 2-FLG and some client equilibrium profile σ, and some scalar $\alpha \geq 1$. We call $\mathbf{s}$ an *α-approximate facility equilibrium* for the facility game induced by σ if no facility player can increase their load by a factor strictly more than α by moving away from $\mathbf{s}$.

Definition 8.2. Consider an instance of 2-FLG and some scalar $\alpha \geq 1$. We call a state $(\mathbf{s}, \sigma) \in S \times \Phi$ an *α-approximate equilibrium* if σ is a client equilibrium profile and $\mathbf{s}$ is a α-approximate facility equilibrium for the facility game induced by σ.

In this chapter, we aim to establish for which values of α the existence of an α-approximate SPE is guaranteed. We do so for different classes of 2-FLG. We first establish that every unrestricted 2-FLG admits a k-approximate SPE, where k denotes the number of facility players. We proceed by finding instances of 2-FLG that admit no α-approximate SPE for small values of α, where we focus on instances with two facility players. We show how to construct instances of (unrestricted) 2-FLG with two facility players that admit no α-approximate equilibria for α strictly smaller than the golden ratio ϕ (≈ 1.618). Additionally, we prove that for general instances of 2-FLG and $\alpha \in [1, \phi)$, the problem of deciding whether the instance admits an α-approximate SPE is NP-hard. Despite great effort, we were unable to prove whether there are instances of unrestricted 2-FLG that do not admit a α-approximate SPE for $\alpha \in [\phi, 2)$. However, we conjecture that such instances do not exist, and discuss some results that could be used to prove this. We now define the *reach* of a vertex, and use this concept to proof Theorem 17.

Definition 8.3. For host graph (V, A, w) and vertex $u \in V$, we define the *reach* $\rho(u)$ of u as the sum of the weights of the clients in the attraction range of a facility placed on u. That is:

$$\rho(u) = w(u) + \sum_{(v,u) \in A} w(u).$$

Theorem 17. *Every unrestricted two-stage facility location game with k facility players admits an k-approximate SPE.*

Proof. Consider an unrestricted instance (H, k) of 2-FLG with host graph $H = (V, A, w)$. Let v_i denote a vertex with maximum reach among the vertices in V, and let $\mathbf{s} \in S$ denote the facility placement profile where all facilities are located on v_i. Let $\sigma(\mathbf{s})$ denote the client subgame profile where every client patronizes each facility in their shopping range with probability $\frac{1}{k}$. This client subgame profile $\sigma(\mathbf{s})$ is a client equilibrium for the client game induced by $\mathbf{s}$. Let σ denote some client equilibrium profile where $\sigma(\mathbf{s})$ is the previously defined client equilibrium. We claim that $(\mathbf{s}, \sigma)$ is a k-approximate SPE.

Assume by contradiction that some facility player $f_j \in \mathcal{F}$ can improve by a factor strictly larger than k by moving away from $\mathbf{s}$. Since $\ell_j(\mathbf{s}, \sigma) = \frac{\rho(v)}{k}$, the load on f_j after this move is strictly larger than $\rho(v)$. It follows that f_j moved to a vertex $v' \in V$ with $\rho(v') > \rho(v)$. By the choice of v, such a vertex does not exist; a contradiction $\square$

This proof does not work when facilities have different location sets, since the facility placement profile where all facilities are located on the vertex with maximum reach might not exist. As such, we cannot conclude that Theorem 17 holds for general instances of 2-FLG.

8.1 Instances Without α-Approximate SPE for α Smaller Than ϕ

Next, we show that for small values of α, the existence of α-approximate SPE is not guaranteed. For $\alpha \in (1, \phi)$, we find instances of 2-FLG that do not admit an α-approximate SPE, where ϕ denotes the golden ratio. These instances are unrestricted and have two facility players.

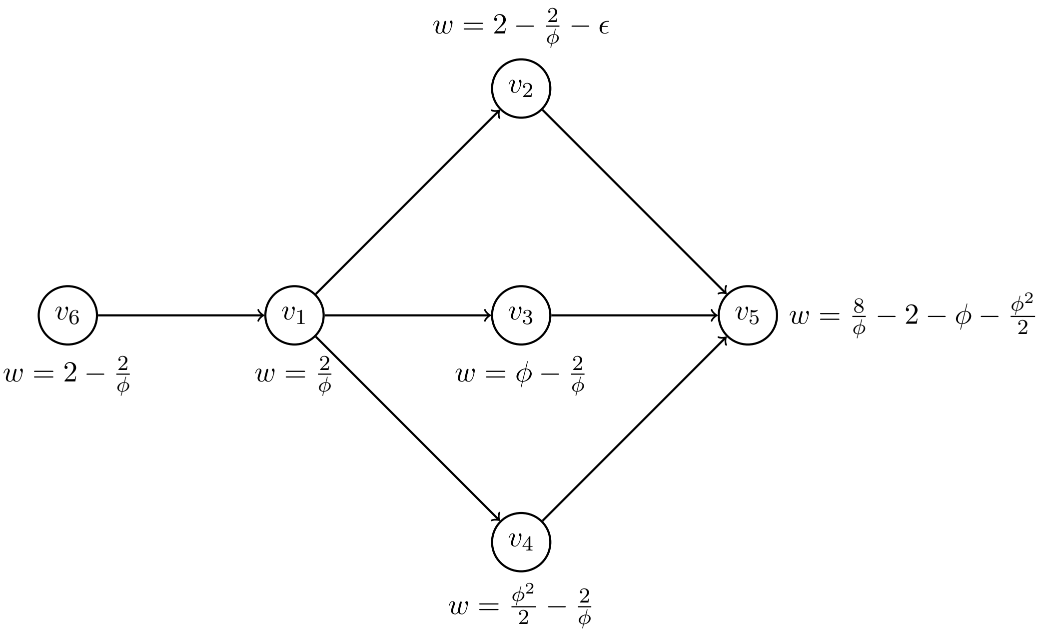


Figure 20: Host graph H

Lemma 27. *Consider the family of instances $(H, 2)$ of unrestricted 2-FLG where H denotes the ϵ-dependent host graph shown in Figure 20, and ϕ denotes the golden ratio. For $\epsilon > 0$ sufficiently small, the instance $(H, 2)$ does not admit a $\phi - \epsilon$-approximate subgame perfect equilibrium.*

Proof. We first establish that the client equilibrium induced by any facility placement profile (s_1, s_2) with $s_1 \neq s_2$, is unique. For any such FPP, the number of clients that can reach both facilities is at most one. Since the reach of every facility location is distinct, the unique client equilibrium is the client subgame profile where the client in the intersection (if there is one) exclusively considers the facility with the smaller reach. If no such client exists, then there is only one feasible client subgame profile, which is the unique client equilibrium. We conclude that for a client game induced by a FPP $\mathbf{s} = (s_1, s_2)$ with $s_1 \neq s_2$, there is a unique client equilibrium. As such, all facilities loads for $(\mathbf{s}, \sigma)$ with $s_1 \neq s_2$ are independent of the chosen client equilibrium profiles $\sigma \in \Omega$. The reach of each vertex of H can be found in Table 3.

Table 3: Reach of each vertex in Figure 20

vertex	**exact reach**	**decimal reach**
v_1	2	2
v_2	$2-\epsilon$	$2-\epsilon$
v_3	ϕ	1.618
v_4	$\frac{\phi^2}{2}$	1.309
v_5	$\frac{2}{\phi}$	1.236
v_6	$2-\frac{2}{\phi}$	0.764

Let σ denote some client equilibrium profile, and consider the facility game induced by σ. We show that for all $\mathbf{s} \in S$, one of the facility players can improve their load by a factor strictly larger than $\phi - \epsilon$ by moving. By symmetry, it suffices to show this for $\{(v_i, v_j) \in S \mid i \leq j\}$. We first consider FPPs where $s_1 \neq s_2$

- For $\mathbf{s} \in \{(v_1, v_2), (v_1, v_3), (v_1, v_4), (v_2, v_3), (v_2, v_4), (v_2, v_4)\}$, the load on f_1 is at most $2 - \frac{2}{\phi} = \frac{2}{\phi^2}$. f_1 can improve by a factor at least ϕ by moving to v_5, since this results in a load of $\rho(v_5) = \frac{2}{\phi}$.
- For $\mathbf{s} \in \{(v_2, v_5), (v_3, v_5), (v_4, v_5)\}$, the load on facility f_1 is exactly $\frac{2}{\phi}$. f_1 can improve by a factor ϕ by moving to v_1, since this results in a load of $\rho(v_1) = 2$.
- For $\mathbf{s} = (v_1, v_5)$, the load on f_2 is $\rho(v_5) = \frac{2}{\phi}$. f_2 can improve by a factor $\phi(1 - \frac{\epsilon}{2}) > \phi - \epsilon$ by moving to v_2, since this results in a load of $\rho(v_2) = 2 - \epsilon$.
- For $\mathbf{s} = (v_i, v_6)$, $i \in \{1, 2, 3, 4, 5\}$, the load on f_2 is $2 - \frac{2}{\phi}$. f_2 can improve by a factor at least ϕ by moving to either v_1 or v_5 (depending on the location of f_1).

Next, we show that for all $v_i \in V$, (v_i, v_i) is not a $(\phi-\epsilon)$-approximate facility equilibrium for the facility game induced by σ. When both facilities are located on the same vertex v_i, one of the facilities has a load of at most $\frac{\rho(v_i)}{2}$. Thus, it suffices to show that for all $v_i \in V$, there is a vertex $v_j \in V$ such that $\ell_1(v_j, v_i) > (\phi - \epsilon)\frac{\rho(v_i)}{2}$. Table 4 lists the vertices v_j corresponding to each profile $\mathbf{s}$ with $s_1 = s_2$.

Table 4: Best-responses for symmetric profiles

v_i	$\frac{\rho(v_i)}{2}$	v_j	$\ell_1(v_j, v_i)$
v_1	1	v_2	$2-\epsilon$
v_2	$1-\frac{\epsilon}{2}$	v_3	ϕ
v_3	$\frac{\phi}{2}$	v_4	$\frac{\phi^2}{2}$
v_4	$\frac{\phi^2}{4}$	v_5	$\frac{2}{\phi} = 1.89 \cdot \frac{\phi^2}{4}$
v_5	$\frac{1}{\phi}$	v_1	2
v_6	$1-\frac{1}{\phi}$	v_2	$2-\epsilon$

□

We conclude that there is no $(\phi - \epsilon)$-approximate facility equilibrium for the facility game induced by σ. Since σ was chosen arbitrarily, this is true for all client equilibrium profiles. We conclude that no $(\phi - \epsilon)$-approximate SPE exists.

Thus, for all $\epsilon > 0$, there exists a host graph H such that the instance $(H, 2)$ of unrestricted 2-FLG admits no $(\phi - \epsilon)$-approximate subgame perfect equilibrium. We conclude that, for any $\alpha < \phi$, the existence of an α-approximate equilibrium is not guaranteed for instances of 2-FLG, and that this is even true for unrestricted instances with two facility players.

8.2 Deciding Whether Approximate SPE Exist

In Theorem 16, we showed that the problem of deciding whether an SPE exists is NP-hard. We now show the same for approximate equilibria.

Theorem 18. *Let $\alpha \in (1, \phi)$ denote some approximation ratio. The problem of deciding whether an instance $(H, \mathcal{U}, k)$ of 2-FLG admits an α-approximate SPE is NP-hard.*

Proof. Let $\alpha \in (1, \phi)$ denote some approximation ratio. We adapt the proof of Theorem 16 to the context of approximate equilibria. As before, we find a polynomial time reduction from the boolean satisfiability problem to the decision problem. Assume we are given an instance of SAT consisting of a set of t clauses $C_1, \ldots, C_t$ and a set of m binary variables $\mathbf{x} = \{x_1, \ldots, x_m\}$, where we assume $t \geq 4$ for simplicity. We construct an instance $(H, \mathcal{U}, k)$ of 2-FLG which depends on the instance of SAT. Host graph H consists of three components: G_1, G_2 and G_3. Component $G_1 = (V_1, A_1, w_1)$ is the same graph as in the proof of Theorem 16, except that the client weights are scaled:

- $V_1 := \{v_1, \ldots, v_m, w_1, \ldots, w_m, c_1, \ldots, c_t, b_1, \ldots, b_{(m-1)\cdot t}\}$,
- $w_1(x) = \frac{m}{m(t+2)-1} \; \forall x \in V_1$,
- A_1 contains the following edges:

$$\begin{aligned}
&(w_i, v_i) \text{ for } i = 1, \ldots, m \\
&(v_i, w_i) \text{ for } i = 1, \ldots, m \\
&(b_j, v_i) \text{ for } i = 1, \ldots, m \text{ and } j = 1, \ldots, (m-1)\cdot t \\
&(b_j, w_i) \text{ for } i = 1, \ldots, m \text{ and } j = 1, \ldots, (m-1)\cdot t \\
&(c_j, v_i) \text{ for } i = 1, \ldots, m \text{ and all } j \text{ s.t. } x_i \in C_j \\
&(c_j, w_i) \text{ for } i = 1, \ldots, m \text{ and all } j \text{ s.t. } \neg x_i \in C_j.
\end{aligned}$$

The other two components, G_2 and G_3, are shown in Figure 21. Note that G_2 is the graph used in the proof of Lemma 27, which was shown not to admit any $(\phi - \epsilon)$-approximate SPE for two facility players for $\epsilon > 0$ sufficiently small. We choose $\epsilon \in \big(0, \min\{2-\phi, 2(1-\frac{\alpha}{\phi})\}\big)$, to ensure that the component does not admit any α-approximate SPE for two facility players.

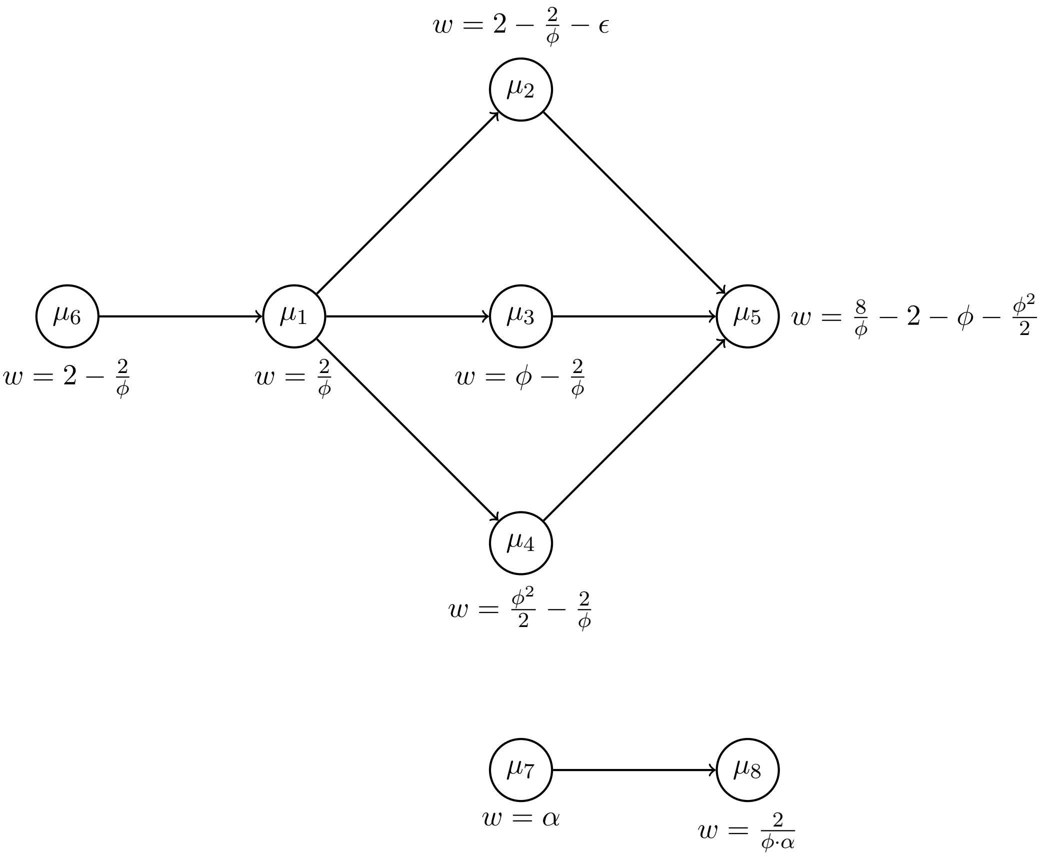


Figure 21: Components G_2 (top) and G_3 (bottom)

We set the number of facility players k to $m+2$, and use the following location function:

$$\mathcal{U}(f_j) = \begin{cases} V_1 \cup \{\mu_7\} & \text{for } j = 1, 2, \ldots, m \\ \{\mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6\} & \text{for } j = m+1 \\ \{\mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6, \mu_8\} & \text{for } j = m+2 \end{cases}$$

We now show that $(H, \mathcal{U}, k)$ admits an α-approximate SPE if and only if there is a satisfying assignment to the variables of $\mathbf{x}$. Assume that there exists a satisfying assignment to the variables of $\mathbf{x}$. Let $\mathbf{y}$ denote one such assignment and let $\mathbf{s}$ denote the following facility placement profile:

$$s_j = \begin{cases} v_j & \text{if } j \leq m \text{ and } y_j \text{ is true} \\ w_j & \text{if } j \leq m \text{ and } y_j \text{ is false} \\ \mu_1 & \text{if } j = m+1 \\ \mu_8 & \text{if } j = m+2. \end{cases}$$

For the same reasons as in the proof of Theorem 16, the client game induced by $\mathbf{s}$ admits a pure, balanced client equilibrium. Let σ denote some pure client equilibrium profile such that $\sigma(\mathbf{s})$ is balanced. To prove that $(\mathbf{s}, \sigma)$ is an α-approximate SPE, it suffices to show that none of the facility players can improve by a factor α by moving away from $\mathbf{s}$ in the facility game induced by σ. First note that f_{m+1} and f_{m+2} cannot improve at all, since the reach of every vertex they can move to is smaller than their current load. Next, note that the sum of the loads of facilities f_1 up to f_m is $m(t+2) \cdot \frac{m}{m(t+2)-1} > m$, since $\mathbf{s}$ covers all clients in V_1. Furthermore, $(\mathbf{s}, \sigma)$ induces equal loads for all facilities on V_1. Thus, the loads on facilities $f_1, \ldots, f_m$ are strictly larger than 1. If any of these facilities moves to μ_7, this will not increase their load by a factor α since $\rho(\mu_7) = \alpha$. For the same reason as in Theorem 16, neither can these facilities improve by moving to a vertex in V_1. Thus, no facility can improve by a factor α by moving away from $\mathbf{s}$. We conclude that $(\mathbf{s}, \sigma)$ is an α-approximate SPE.

Now assume that there exists no satisfying assignment to the variables of $\mathbf{x}$, and assume by contradiction that an α-approximate SPE $(\mathbf{s}, \sigma)$ exits. We consider the facility game induced by σ and show that some facility can improve by a factor α by moving away from $\mathbf{s}$. The proof of Lemma 27 shows that no α-approximate facility equilibrium exists with exactly two facilities located on the vertices of G_2. Since f_{m+1} and f_{m+2} are the only facilities able to pick any vertex on G_2 as their location, and f_{m+1} can *only* pick these vertices, f_{m+2} cannot be located on G_2. This gives $s_{m+2} = \mu_8$.

Next, note that $s_{m+1} \neq \mu_5$, as this would allow f_{m+1} to improve by a factor $\phi > \alpha$ by moving to μ_1. It follows that f_{m+2} can attain a load of $\frac{2}{\phi}$ by moving to μ_5. By assumption, this move does not increase the load on f_{m+2} by a factor α. That is, $\ell_{m+2}(\mathbf{s}, \sigma) > \frac{2}{\phi \cdot \alpha}$. It follows that client μ_7 considers f_{m+2}, which implies that no facility is located on μ_7. We conclude that facilities f_1 up to f_m are all located on V_1.

Since we assumed that there is no satisfying assignment to the variables of $\mathbf{x}$, a set of m facilities cannot cover all clients in V_1. The sum of the loads on facilities f_1 up to f_m is therefore at most $(m(t+2) - 1) \cdot \frac{m}{m(t+2)-1} = m$. Thus, there is some facility f_j with $j \leq m$ with load at most 1. Then, f_j can improve their load by a factor α by moving to μ_7, meaning that $\mathbf{s}$ is not an α-approximate facility equilibrium; a contradiction.

We have found a polynomial time reduction from SAT to the problem of deciding the existence of α-approximate SPE in 2-FLG. We conclude that the problem of deciding whether an instance of 2-FLG admits an α-approximate SPE is NP-hard. □

It should be clear that this decision problem is in NP, as verifying that a facility placement profile is an approximate facility equilibrium is not harder than verifying that it is a (non-approximate) facility equilibrium. As such, given the right certificate, one can verify in polynomial time that some instance admits an approximate SPE. We conclude that the decision problem is NP-complete.

8.3 Towards a Better Bound on the Approximation Constant

So far, we have shown that for all $\alpha < \phi$, there are unrestricted 2-FLG with two facility players that admit no α-approximate SPE. We also proved that all such instances *do* admit a 2-approximate SPE. This leads to the following question: do all unrestricted 2-FLG with $k = 2$ admit an α-approximate SPE, for $\alpha \in [\phi, 2)$? To address this question, we define the *approximation constant* α_{min} for an instance of 2-FLG as the smallest value of α for which an α-approximate equilibrium exists. Establishing upper bounds on α_{min} is quite difficult. As such, we again limit our scope to unrestricted instances with two facility players. This enables the use of the theory developed in Chapter 3. We tried to prove an upper bound on α_{min} for these instances, that is strictly smaller than 2. Unfortunately, we did not succeed in this endeavor. Nevertheless, we believe that our approach has merit and provide an overview of our efforts.

Leveling Client Equilibria

Establishing a better upper bound on α_{min} requires proving that every instance of 2-FLG with two facility players admits an α-approximate equilibrium for some value of α. Theoretically, the type of client behavior needed for states to be α-approximate SPE (for large α) may depend on the instance of 2-FLG. However, we believe this to be unlikely, and this would make establishing upper bounds exceedingly difficult. Therefore, we choose one type of client equilibrium that is easy to work with, and which induces facility games that are likely to admit approximate SPE. If the assumption that there are no α-approximate SPE using this client equilibrium profile, leads to a contradiction, we will have shown that every instance of 2-FLG with two facility players admits an α-approximate SPE. These *leveling* client equilibria are straightforward in concept; when the reaches for both facility locations are the same, both facilities get equal loads. When the reaches are different, a *leveling* client equilibrium is the pure client equilibrium that maximizes the load on the facility with the smaller reach. Formally:

Definition 8.4. For an instance of 2-FLG with $k = 2$, we call a client equilibrium profile $\sigma \in \Omega$ *leveling* if for every facility placement profile $\mathbf{s} := (s_1, s_2) \in S$:

- if $\rho(s_1) = \rho(s_2)$, then $\ell_1(\mathbf{s}, \sigma(\mathbf{s})) = \ell_2(\mathbf{s}, \sigma(\mathbf{s}))$.
- if $\rho(s_1) > \rho(s_2)$, then $\sigma(\mathbf{s}) \in \Omega_{\mathbf{s}}$ is the pure client equilibrium that maximizes $\ell_2(\mathbf{s}, \sigma(\mathbf{s}))$.
- if $\rho(s_2) > \rho(s_1)$, then $\sigma(\mathbf{s}) \in \Omega_{\mathbf{s}}$ is the pure client equilibrium that maximizes $\ell_1(\mathbf{s}, \sigma(\mathbf{s}))$.

Leveling client equilibria have many useful properties to exploit. The first of these is that leveling equilibria are *load-symmetric.*

Definition 8.5. Consider some instance of (bipartite) 2-FLG with k facility players. For any FPP $\mathbf{s} \in S$, let $\mathbf{s}^{(i,j)}$ denote the FPP obtained from $\mathbf{s}$ by interchanging s_i and s_j. We say that a client equilibrium profile $\sigma \in \Omega$ is *load-symmetric* if for all $\mathbf{s} \in S$ and all $i, j \in \{1, 2, \ldots, k\}$, we have:

$$\begin{aligned}
&\ell_i(\mathbf{s}^{(i,j)}, \sigma) = \ell_j(\mathbf{s}, \sigma) \\
&\ell_j(\mathbf{s}^{(i,j)}, \sigma) = \ell_i(\mathbf{s}, \sigma) \\
&\ell_x(\mathbf{s}^{(i,j)}, \sigma) = \ell_x(\mathbf{s}, \sigma) \text{ for all } x \notin \{i, j\}
\end{aligned}$$

Lemma 28. *Every leveling client equilibrium vector is load-symmetric.*

Proof. Let σ denote a leveling client equilibrium profile. By symmetry, it suffices to show that for every $\mathbf{s} = (s_1, s_2) \in S$, we have $\ell_1((s_2, s_1), \sigma) = \ell_2((s_1, s_2), \sigma)$. If $\rho(s_1) = \rho(s_2)$, then since σ is leveling:

$$\ell_1((s_2, s_1), \sigma) = \frac{w\big(A_{\mathbf{s}}(f_1) \cup A_{\mathbf{s}}(f_2)\big)}{2} = \ell_2((s_1, s_2), \sigma).$$

If $\rho(s_2) > \rho(s_1)$, then $\sigma((s_1, s_2))$ is a pure client equilibrium maximizing the load on f_1. Define the pure client equilibrium $\sigma'(s_2, s_1)$ as follows:

$$\begin{aligned}
\sigma'(s_1, s_2)_{i,2} = \sigma(s_2, s_1)_{i,1} \ \forall v_i \in V \\
\sigma'(s_1, s_2)_{i,1} = \sigma(s_2, s_1)_{i,2} \ \forall v_i \in V
\end{aligned}$$

This gives:

$$\ell_1((s_2, s_1), \sigma) = \ell_2((s_1, s_2), \sigma') \geq \ell_2((s_1, s_2), \sigma).$$

To show that $\ell_2((s_1, s_2), \sigma) \geq \ell_1((s_2, s_1), \sigma)$, define $\sigma'(s_1, s_2)$ and apply a similar argument. Thus, $\ell_1((s_2, s_1), \sigma) = \ell_2((s_1, s_2), \sigma)$. A similar argument is used to show load-symmetry holds for $(s_1, s_2) \in S$ with $\rho(s_1) > \rho(s_2)$. □

Leveling equilibria are easy to work with, primarily because these are load-symmetric and unique with respect to the induced facility loads. Additionally, they possess some useful minor properties regarding the behavior of the *contested* clients.

Definition 8.6. Consider some instance of unrestricted 2-FLG with two facility players and some client profile $\sigma \in \Phi$. For $\mathbf{s} \in S$, let $I(\mathbf{s}) := A_{\mathbf{s}}(f_1) \cap A_{\mathbf{s}}(f_2)$ denote the set of *contested clients.* Furthermore, let $I_1(\mathbf{s}, \sigma)$ and $I_2(\mathbf{s}, \sigma)$ denote the sets of contested clients that consider f_1 and f_2 for $\sigma(\mathbf{s})$, respectively.

Note that unlike $I(\mathbf{s})$, both $I_1(\mathbf{s}, \sigma)$ and $I_2(\mathbf{s}, \sigma)$ depend on the client (subgame) profile. We will show that for leveling client subgame profiles, at least half of the contested clients patronize the facility with the smaller reach, assuming that the facilities have distinct reaches. To this end, we establish Lemma 29, which is not meaningful on its own but is needed for several proofs.

Lemma 29. *Consider an instance of unrestricted 2-FLG with $k = 2$ and an FPP $\mathbf{s} \in S$. Suppose that for some pure client subgame profile $\sigma(\mathbf{s})$ we have that:*

1. $\gamma := \ell_2(\mathbf{s}, \sigma(\mathbf{s})) - \ell_1(\mathbf{s}, \sigma(\mathbf{s})) > 0$
2. $w(v) \geq \gamma \quad \forall v \in I_1(\mathbf{s}, \sigma(\mathbf{s}))$.

Then, there exists a pure client equilibrium $\sigma^*(\mathbf{s})$ such that $\ell_1(\mathbf{s}, \sigma^*(\mathbf{s})) - \ell_2(\mathbf{s}, \sigma^*(\mathbf{s})) < \gamma$.

Proof. Consider an instance of 2-FLG with $k = 2$ and let $\mathbf{s} \in S$ denote a facility placement profile. Assume that a pure client subgame profile $\sigma(\mathbf{s})$ satisfies the aforementioned conditions. If $\sigma(\mathbf{s})$ is a client equilibrium, we set $\sigma^*(\mathbf{s}) = \sigma(\mathbf{s})$, and we are done. If not, we can construct a client equilibrium $\sigma^*(\mathbf{s})$ with the desired properties.

Note that any pure client subgame profile $\sigma'(\mathbf{s})$ is uniquely characterized by $I_1(\mathbf{s}, \sigma'(\mathbf{s}))$. We define $A^1 := I_1(\mathbf{s}, \sigma(\mathbf{s}))$ and $\sigma^1(\mathbf{s}) := \sigma(\mathbf{s})$. Client equilibrium $\sigma^*(\mathbf{s})$ can then be obtained with the following procedure.

1. Set $i = 1$
2. Find a client v_i of maximum weight among the set of clients in $I_2(\mathbf{s}, \sigma^i(\mathbf{s}))$ that can improve by moving away from $\sigma^i(\mathbf{s})$ in the client game induced by $\mathbf{s}$. If no such client exists, set $\sigma^*(\mathbf{s}) = \sigma^i(\mathbf{s})$ and terminate.
3. Set $A^{i+1} = A^i \cup \{v_i\}$ and let $\sigma^{i+1}(\mathbf{s})$ denote the pure client subgame profile characterized by $I_1(\mathbf{s}, \sigma(\mathbf{s})) = A^{i+1}$
4. Increment i by one and return to step 2.

As there are only a finite number of clients, this algorithm always terminates. It remains to be shown that the found $\sigma^*(\mathbf{s})$ is a client equilibrium. To see this, first note that none of the clients in $I_2(\mathbf{s}, \sigma^*(\mathbf{s}))$ can improve, since this is the stopping criterion for the algorithm. Next, note that for $j = 1, \ldots, i - 1$, we have:

$$w(v_j) < \ell_2(\mathbf{s}, \sigma^j(\mathbf{s})) - \ell_1(\mathbf{s}, \sigma^j(\mathbf{s})) \leq \gamma,$$

since v_j was able to improve. Thus, v_{i-1} is a client of minimal weight in $I_1(\mathbf{s}, \sigma^*(\mathbf{s}))$. Since v_{i-1} was the last client to move, it does not have an improving move, and thus neither do any of the clients with equal or larger weight. We conclude that $\sigma^*(\mathbf{s})$ is a client equilibrium. Furthermore, $\ell_1(\mathbf{s}, \sigma^*(\mathbf{s})) - \ell_2(\mathbf{s}, \sigma^*(\mathbf{s})) < w(v_{i-1}) < \gamma$. □

Lemma 30. *Consider some instance of (bipartite) 2-FLG with 2 facility players and let σ denote a leveling client equilibrium profile. Then for all $\boldsymbol{s} \in S$ with $\rho(s_1) < \rho(s_2)$, we find $w(I_2(\boldsymbol{s}, \sigma)) \leq w(I_1(\boldsymbol{s}, \sigma))$.*

Proof. Assume by contradiction that there is an $\mathbf{s} \in S$ with $\rho(s_1) < \rho(s_2)$ and $w(I_1(\mathbf{s}, \sigma)) < w(I_2(\mathbf{s}, \sigma))$. Define $A := I_1(\mathbf{s}, \sigma)$ and $B := I_2(\mathbf{s}, \sigma)$. Since $\sigma(\mathbf{s})$ is a pure client equilibrium for the client game induced by $\mathbf{s}$:

$$w(v) \geq \ell_2(\mathbf{s}, \sigma) - \ell_1(\mathbf{s}, \sigma) \ \forall v \in B. \tag{13}$$

Consider pure client subgame profile $\sigma'(\mathbf{s})$ characterized by $I_1(\mathbf{s}, \sigma'(\mathbf{s})) = B$ and $I_2(\mathbf{s}, \sigma'(\mathbf{s})) = A$. We define $\gamma := \ell_2(\mathbf{s}, \sigma'(\mathbf{s})) - \ell_1(\mathbf{s}, \sigma'(\mathbf{s}))$ and find that:

$$|\gamma| < \ell_2(\mathbf{s}, \sigma(\mathbf{s})) - \ell_1(\mathbf{s}, \sigma(\mathbf{s})),$$

since $w(B) > w(A)$ by assumption. Furthermore, $\ell_1(\mathbf{s}, \sigma'(\mathbf{s})) > \ell_1(\mathbf{s}, \sigma(\mathbf{s}))$.

We now claim that $\sigma'(\mathbf{s})$ is a client equilibrium if $\gamma \leq 0$. If $\gamma \leq 0$, none of the clients in $I_1(\mathbf{s}, \sigma'(\mathbf{s}))$ can improve by (13). The clients in $I_2(\mathbf{s}, \sigma'(\mathbf{s}))$ cannot improve since $\ell_1(\mathbf{s}, \sigma'(\mathbf{s})) \geq \ell_2(\mathbf{s}, \sigma'(\mathbf{s}))$. If $\gamma > 0$, we can apply Lemma 29 to show the existence of a client equilibrium $\sigma^*(\mathbf{s})$ with $\ell_1(\mathbf{s}, \sigma^*(\mathbf{s})) \geq \ell_1(\mathbf{s}, \sigma'(\mathbf{s})) > \ell_1(\mathbf{s}, \sigma(\mathbf{s}))$. In either case, we have found a pure client equilibrium that induces a larger load on f_1 than $\sigma(\mathbf{s})$ does. Thus, σ cannot be leveling; a contradiction. □

Another useful property of leveling client equilibria is shown in Lemma 31: if any contested client patronizes the facility with the larger reach, then the load on this facility is less than the reach of the location of the other facility.

Lemma 31. *Consider some instance of (bipartite) 2-FLG, and let σ denote a leveling client equilibrium profile. For all $\boldsymbol{s} \in S$ with $\rho(s_1) > \rho(s_2)$, one of the following holds:*

1. $\ell_2(\boldsymbol{s}, \sigma) = \rho(s_2)$
2. $\ell_1(\boldsymbol{s}, \sigma) < \rho(s_2)$

Proof. Let $\mathbf{s} \in S$ denote a facility placement profile with $\rho(s_1) > \rho(s_2)$. Let $\sigma(\mathbf{s})$ denote the pure client subgame profile where all contested clients exclusively consider f_2, so $I_1(\mathbf{s}, \sigma(\mathbf{s})) = \emptyset$.

If $\ell_2(\mathbf{s}, \sigma(\mathbf{s})) \leq \ell_1(\mathbf{s}, \sigma(\mathbf{s}))$, then $\sigma(\mathbf{s})$ is a pure client equilibrium with $\ell_2(\mathbf{s}, \sigma(\mathbf{s})) = \rho(s_2)$. Since every leveling client equilibrium maximizes the load on f_2, we find that $\ell_2(\mathbf{s}, \sigma'(\mathbf{s})) = \rho(s_2)$ for all leveling client equilibria $\sigma'(\mathbf{s}) \in \Omega_{\mathbf{s}}$. We conclude that statement (1.) holds.

Now assume $\ell_2(\mathbf{s}, \sigma(\mathbf{s})) > \ell_1(\mathbf{s}, \sigma(\mathbf{s}))$. By Lemma 29, there exists a pure client equilibrium $\sigma^*(\mathbf{s})$ such that:

$$\ell_2(\mathbf{s}, \sigma(\mathbf{s})) - \ell_1(\mathbf{s}, \sigma(\mathbf{s})) > \ell_1(\mathbf{s}, \sigma^*(\mathbf{s})) - \ell_2(\mathbf{s}, \sigma^*(\mathbf{s})).$$

Since the sum of the facility loads is the same for every client subgame profile, we find:

$$\ell_1(\mathbf{s}, \sigma^*(\mathbf{s})) < \ell_2(\mathbf{s}, \sigma(\mathbf{s})) = \rho(s_2).$$

Since every leveling client equilibrium maximizes the load on f_2, and thus minimizes the load on f_1, we find that $\ell_1(\mathbf{s}, \sigma'(\mathbf{s})) < \rho(s_2)$ for all leveling client equilibria $\sigma'(\mathbf{s}) \in \Omega_{\mathbf{s}}$. We conclude that statement (2.) holds. □

Next, we try to prove that for certain values of α, unrestricted instances with two facility players admit an α-approximate SPE $(\mathbf{s}, \sigma)$ where σ is a leveling client equilibrium profile. To this end, we will assume the existence of an instance without such an equilibrium, and show that this assumption leads to a contradiction. That is, we analyze facility games induced by leveling client equilibrium profiles, and show that for every $\mathbf{s} \in S$, one of the facilities can improve by a factor α.

Best-Response Cycles

A facility game that does not admit an α-approximate equilibrium, always contains a cycle of best-responses where the facility player performing the improving move improves by a factor of at least α. If no such cycle were to exist, then any sequence of best-improving moves would eventually end up in an α-approximate facility equilibrium. An elegant way to disprove the existence of such instances is to show that this cycle cannot exist. Unfortunately, this method cannot work, as there are instances with such a cycle for values of α arbitrarily close to 2. For example, consider the graph shown in Figure 22. For the facility game induced by any (load-symmetric) client equilibrium profile, $C = \{(v_1, v_2), (v_3, v_2), \ldots, (v_{m-1}, v_m), (v_1, v_m)\}$ is such a cycle of best-responses. Furthermore, when $m \to \infty$, the factor by which the moving player improves their load converges to 2.

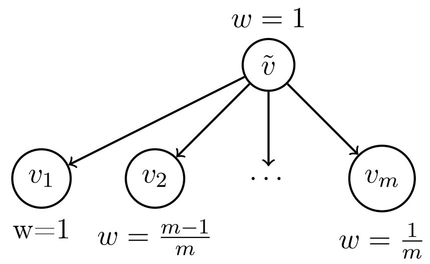


Figure 22: Host graph with cycle.

It is therefore impossible to disprove the existence of such best-response cycles. Furthermore, proofs using a potential function are unlikely to work, as these generally rely on contradicting the existence of a cycle of best-responses.

Note, however, that α-approximate facility equilibria *do* exist for the example instance, for values of α as low as $\sqrt{2}$. For example, $(v_1, v_{\lfloor m(2-\sqrt{2}) \rfloor - 1})$ is an $\sqrt{2}$-approximate facility equilibrium for the facility game induced by any leveling client equilibrium profile.

If we can show that the existence of a best-response cycle, implies the existence of some other α-approximate facility equilibrium, we will have found our contradiction. This requires some understanding of best-response dynamics in two-player games. Firstly, in a cycle of best-response moves with two players, the players move alternately. In each step, a player performs the best-response to their opponent's strategy. As such, they cannot improve from the facility placement profile resulting from the move. Secondly, the new location of the moving player depends only on the location of the opponent, not on the original location of the moving player. Therefore, if the best-response of f_1 in FPP (v_i, v_j) is moving to v_m, then moving to v_m is also the best-response for f_1 in $(v_{i'}, v_j)$ for any $v_{i'} \in V$.

To constrain the graph structure of facility games, we consider bipartite instances of 2-FLG. In particular, we consider non-degenerate bipartite instances (see Definition 3.6). We've argued that for any instance of unrestricted 2-FLG, there is a non-degenerate instance of bipartite 2-FLG that is equivalent for all practical purposes. Therefore, it suffices to show that these instances admit approximate SPE satisfying the criteria.

Lemma 32 shows that the smallest instance (H, U, w) of bipartite 2-FLG that does not admit any $\alpha-$approximate equilibrium must be non-degenerate, and admits a type of cycle of best-response moves.

Lemma 32. *Let (H, U, w) be an instance of bipartite 2-FLG that does not admit an α-approximate SPE $(\boldsymbol{s}, \sigma)$ where σ is a leveling client profile. Further assume that (H, U, w) has the smallest host graph among all such instances (firstly in terms of $|U|$ and secondly in terms of the number of clients). Then the following conditions hold:*

1. *(H, U, w) is non-degenerate*
2. *Let σ denote a leveling client equilibrium profile. Then the facility game induced by σ admits a cycle of best-responses C where the moving player improves by a factor larger than α. Furthermore, every location vertex $u_i \in U$ appears in some element of C.*

Proof. Assume by contradiction that the instance is degenerate. Then there exists an equivalent non-degenerate instance that is smaller and does not admit an α-approximate SPE; a contradiction.

Next, we prove the existence of C. Since the number of distinct facility placement profiles is finite, there must be a cycle of best-responses. Let C denote the shortest such cycle, and let U_C denote the set of location vertices that appear in some element of C. Then, for all $(s_i, s_j) \in S$ with $s_i, s_j \in U_C$, one of the players can sufficiently improve by moving to some location vertex $u \in U_C$. Now consider the original host graph H and the facility game induced by some leveling client profile σ'. By assumption, there is a facility player that can improve sufficiently by moving away from FPP (s_i, s_j). Furthermore, we know that the best-response for this player is moving to some $u \in U_C$.

Next, consider instance $(H', U_C, 2)$, where H' is the subgraph of H induced by $U_C \cup (V \setminus U)$. Let σ' denote some leveling client equilibrium profile, and consider the facility game induced by σ'. Since all leveling client equilibrium profiles induce the same facility loads, the facility loads induced by σ and σ' are the same for all facility placement profiles in $S' := U_C^2$. Thus, for any $(s_i, s_j) \in S'$, one of the players can improve sufficiently by moving to some $u \in U_C$. This implies that instance $(H', U_C, 2)$ does not admit any α-approximate SPE $(\mathbf{s}, \sigma')$, where σ' is leveling. Since H was assumed to be the smallest host graph with this property, we conclude that $H = H'$ and thus $U = U_C$, which proves that every vertex in U appears in some element of C. □

Although we do not prove this formally, we can assume further structure on a shortest cycle C. Specifically, we may assume that there is only one best-response for each strategy of the other facility player. Otherwise, it would be possible to construct a cycle that includes only a proper subset of the location vertices, which contradicts C being a shortest cycle. Thus, without loss of generality:

$$C = \begin{cases} \Big((u_1, u_2), (u_3, u_2), \ldots, (u_{m-1}, u_m), (u_1, u_m)\Big) & \text{if } m := |U| \text{ is even} \\ \Big((u_1, u_2), (u_3, u_2), \ldots, (u_m, u_1), (u_2, u_1), \ldots, (u_1, u_m)\Big) & \text{if } m := |U| \text{ is odd.} \end{cases} \tag{14}$$

For the remainder of this chapter, we assume the existence of a cycle C with these properties. Since labeling of vertices is arbitrary, and we can scale all weights by any positive nonzero scalar, we assume without loss of generality that u_1 is one of the vertices with maximal reach in U, and that this reach is 2. We formally define the instance satisfying all these assumptions and summarize the results so far in Lemma 33.

Definition 8.7. Consider some $\alpha \in [\phi, 2)$, and a non-degenerate instance $(H, U, 2)$ of bipartite 2-FLG. Let σ denote a leveling client equilibrium profile. Then, we call $(H, U, 2)$ an *α-instance* if it satisfies the following conditions:

1. (H, U, k) admits no α-approximate SPE where the client equilibrium profile is leveling.
2. There is a shortest cycle C of best-responses in the facility game induced by σ, where every location vertex in U appears in some element of C.
3. C is of the form shown in (14), with $\rho(u_1) = 2$ and $\rho(u_j) \leq 2$ for all $u_j \in U$.

We call this cycle C the α-cycle for this α-instance.

Lemma 33. *If no α-instance exists for some $\alpha \in [\phi, 2)$, then $\alpha_{min} < \alpha$ for all instances of unrestricted 2-FLG with two facility players.*

Contradicting the Existence of α-Instances

Lemma 33 points towards a clear goal: to prove that α-instances cannot exist for certain values of α. For an α-instance, the α-cycle C characterizes the best-responses to all strategies. Since no α-approximate facility equilibrium exists, one of the two facility playerss can improve by a factor α by a best-response move for every facility placement profile in S. We aim to use these relations between facility loads corresponding to different facility placement profiles to derive a contradiction on the structure of the host graph. We denote $u_{m+i} := u_i$ for $i \in \{1, 2, ..., m\}$ and consider facility placement profiles (u_i, u_{i+1}) on the α-cycle C. We find:

$$\ell_1(u_{i+2}, u_{i+1}) > \alpha \cdot \ell_1(u_i, u_{i+1}) \quad \forall i \in \{1, \dots, m\}, \tag{15}$$

since the moving player always improves by a factor at least α, and f_1 is the moving player for any (u_i, u_{i+1}) on S. For facility placement profiles that are not on C, at least one of the facility players can improve their load by a factor α by performing a best-response move. That is, for every $(u_i, u_j) \in S$, at least one of the following holds:

- $\ell_2(u_i, u_{i+1}) > \alpha \cdot \ell_2(u_i, u_j)$
- $\ell_1(u_{j+1}, u_j) > \alpha \cdot \ell_1(u_i, u_j)$.

We use these conditions to derive restrictions on the reaches of vertices in U.

Lemma 34. *Consider some $\alpha \in [\phi, 2)$, and let $(H, U, 2)$ denote an α-instance with α-cycle C and leveling client equilibrium profile σ. Then for all $i \in \{1, 2, \dots, m\}$, we find:*

$$\rho(u_{i+1}) > \frac{\alpha}{2} \rho(u_i). \tag{16}$$

Proof. Since σ is load-symmetric, we find:

$$\ell_1(u_i, u_i) = \ell_2(u_i, u_i) = \frac{\rho(u_i)}{2}.$$

Furthermore, one of the players can improve sufficiently by performing the best-response move. By symmetry, we assume w.l.o.g. that this is f_2. This gives:

$$\rho(u_{i+1}) \geq \ell_2(u_i, u_{i+1}) > \alpha \cdot \ell_2(u_i, u_i) = \frac{\alpha}{2}\rho(u_i)$$

□

Lemma 35. *Consider some $\alpha \in [\phi, 2)$, and let $(H, U, 2)$ denote an α-instance with α-cycle C and leveling client equilibrium profile σ. Then, if two consecutive location vertices $u_i, u_{i+1} \in C$ have equal reach, this reach is at most $\frac{4}{\alpha^2}$.*

Proof. Recall that, $\rho(u) \leq 2$ for all $u \in U$ and let $u_i, u_{i+1} \in S$ be such that $\rho(u_i) = \rho(u_{i+1}) = r$. We find:

$$\ell_2(u_{i+1}, u_i) = \ell_1(u_{i+1}, u_i) > \alpha \cdot \ell_1(u_i, u_i) = \alpha \cdot \frac{\rho(u_i)}{2} = \alpha \cdot \frac{r}{2}. \tag{17}$$

For f_2 to sufficiently improve by performing a best-response move for $\mathbf{s} = (u_{i+1}, u_i)$, we require that $\ell_2((u_{i+1}, u_i), \sigma) < \frac{2}{\alpha}$. After all, the maximum achievable load is at most 2, and f_2 can improve by a factor α. We find:

$$\frac{2}{\alpha} > \ell_2(u_{i+1}, u_i) > r \cdot \frac{\alpha}{2},$$

and conclude that $r < \frac{4}{\alpha^2}$. □

Corollary 18.1. *Consider some $\alpha \in [\phi, 2)$, and let $(H, U, 2)$ denote an α-instance. Then, $\rho(u_2) < 2$ and $\rho(u_m) < 2$.*

Proof. Follows immediately from $\rho(u_1) = 2$, $\frac{4}{\alpha^2} < 2$ and applying Lemma 35. □

Lemma 36. *Consider some $\alpha \in [\phi, 2)$, and let $(H, U, 2)$ denote an α-instance with leveling client equilibrium profile σ. Then $I_2((u_1, u_2), \sigma)$ is nonempty. Furthermore, let $\tilde{v}$ denote a client of maximum weight in $I_2((u_1, u_2), \sigma)$. Then $w(\tilde{v}) \geq 2 - \frac{2}{\alpha}$.*

Proof. To see that $I_2((u_1, u_2), \sigma)$ is nonempty, note that $\rho(u_1) = 2$ and $\ell_1((u_1, u_2), \sigma) \leq \frac{2}{\alpha}$. Thus, $w(I_2((u_1, u_2), \sigma) \geq 2 - \frac{2}{\alpha}$, which requires at least one client in $I((u_1, u_2))$ to consider f_2.

To see that there must be some client with weight at least $2 - \frac{2}{\alpha}$, first note that by Lemma 34, $\ell_2((u_1, u_2), \sigma) \geq \phi$. Assume by contradiction that no client in $I_2((u_1, u_2), \sigma)$ has a weight of at least $2 - \frac{2}{\alpha}$. Then $I_2((u_1, u_2), \sigma)$ contains at least two clients. Let v_i denote a client in $I_2((u_1, u_2), \sigma)$ with minimal weight. We find that $w(v_i) \leq \frac{1}{2}w(I_2((u_1, u_2), \sigma))$. We show that the v_i-excluded load on f_2 is larger than that on f_1, which implies that $\sigma((u_1, u_2))$ is not a client equilibrium; a contradiction. We find:

$$\begin{aligned}
\ell_{-i,2}(u_1, u_2) &\geq \rho(u_2) - \frac{1}{2}w(I_2((u_1, u_2), \sigma)) \\
&\geq \alpha - \frac{1}{2}(2 - \ell_1((u_1, u_2), \sigma)) \\
&= \alpha - 1 + \frac{1}{2}\ell_1((u_1, u_2), \sigma) \\
&> \frac{1}{\alpha} + \frac{1}{2}\ell_1((u_1, u_2), \sigma) \\
&\geq \ell_1((u_1, u_2), \sigma) = \ell_{-i,1}((u_1, u_2), \sigma)
\end{aligned}$$

□

The client $\tilde{v}$ mentioned in Lemma 36 plays a key role in α-instances. For the constructed instances with large α_{min} (see Figure 20), the best-response is often to move to a vertex adjacent from $\tilde{v}$ with a slightly smaller reach than that of the opponent, to "steal" $\tilde{v}$ from this opponent.

The discussed lemmas restrict the possible graph structure of α-instances. Several of these lemmas only hold because $\alpha \geq \phi$, which shows the significance of the golden ratio for this problem. These restrictions on the graph structure form our toolbox for deriving a contradiction. Our efforts to find this contradiction resulted in case distinction upon case distinction, not all of which we were able to solve. We illustrate how the different restrictions may lead to contradictions by considering a solved case. We also provide some intuition for how the restrictions could be used to solve the unsolved case distinctions.

Consider $\rho(u_m)$ for some α-instance for $\alpha \in [\phi, 2)$. We know that f_2 can increase their load by a factor α by moving from $((u_1, u_m), \sigma)$ to $((u_1, u_2), \sigma)$. This implies that $\ell_2((u_1, u_m), \sigma) \leq \frac{y}{\alpha}$, where we define $y := ((u_1, u_2), \sigma) < 2$. This leads to two cases, both of which restrict the graph structure in their own ways:

1. $\rho(u_m) \leq \frac{y}{\alpha} < \frac{2}{\alpha}$.
2. $\rho(u_m) > \frac{y}{\alpha}$. This implies that $\ell_2((u_1, u_m), \sigma) < \rho(u_m)$.

Intuitively, the first condition makes it difficult for a facility to sufficiently improve when moving to u_m. Since moving to u_m is the best-response when the opponent is located on u_{m-1}, this places strict conditions on facility placement profiles which include location vertex u_{m-1}.

Now consider the case where $\ell_2((u_1, u_m), \sigma) < \rho(u_m)$. We make a further case distinction on based on whether or not u_m is adjacent to $\tilde{v}$. We provide some intuition of the imposed restrictions when u_m is adjacent to $\tilde{v}$, but do not consider the case formally. This adjacency restricts the weight of $\tilde{v}$; for a large weight of $\tilde{v}$, the load on f_2 in profiles (u_x, u_m) for $u_x \in U$ with $\rho(u_x) > \rho(u_m)$ is quite large, as f_2 is always patronized by $\tilde{v}$ for such profiles. This makes it difficult for u_m to sufficiently improve by moving to u_{x+1}. Additionally, it is difficult for u_x to improve by moving to u_1, since $\ell_1((u_1, u_m), \sigma) \leq 2 - w(\tilde{v})$, which is small for large values of $w(\tilde{v})$. At the same time, smaller values of $w(\tilde{v})$ are also restrictive. It follows that for $u_x \in U$ with $\rho(u_x) < \rho(u_2)$, either $\ell_1((u_1, u_x), \sigma)$ or $\ell_1((u_2, u_x), \sigma)$ is large since there is no single client of large weight adjacent to u_1, u_2 and u_x. If the reach of u_x is large, neither facility might be able to improve from such profiles.

We now show that assuming that $\ell_2((u_1, u_m), \sigma) < \rho(u_m)$, and that $\tilde{v}$ is not adjacent to u_m, lead to a contradiction. To this end, we consider the sum of the weights of the clients adjacent to u_m and prove that this is strictly larger than $\rho(u_m)$, which contradicts the definition of the reach of a vertex. We prove Lemma 37, which gives a lower bound on $w(I(u_1, u_m))$.

Lemma 37. *Consider some $\alpha \in [\phi, 2)$, and let $(H, U, 2)$ denote an α-instance with α-cycle C and leveling client equilibrium profile σ. Let $u_i, u_j \in U$ denote a pair of location vertices with $\rho(u_j) < \rho(u_i)$ and $x := \ell_1((u_j, u_i), \sigma) < \rho(u_j)$. Then:*

$$w(I_2((u_j, u_i), \sigma)) \geq \rho(u_j) - x$$
$$w(I_1((u_j, u_i), \sigma)) \geq \max\Big(\rho(u_j) - x, \rho(u_i) - \rho(u_j)\Big).$$

Proof. Follows from Lemmas 30 and 31 and the assumption that σ is leveling. □

Theorem 19. *Consider some $\alpha \in [\phi, 2)$, and let $(H, U, 2)$ denote an α-instance with α-cycle C and leveling client equilibrium profile σ. Further assume that $\rho(u_m) > \frac{1}{\alpha}\ell_2((u_1, u_2), \sigma)$. Then, $\tilde{v}$ is adjacent to u_m.*

Proof. Assume by contradiction that u_m is not adjacent to $\tilde{v}$. We will show that $\rho(u_m) < w(I(u_1, u_m) \cup I(u_2, u_m))$, which contradicts the definition of the reach of u_m. To this end, we first provide an upper bound on the sum of the weights of the clients adjacent to all vertices in $\{u_1, u_2, u_m\}$.

By definition, $\tilde{v} \in I_2((u_1, u_2), \sigma)$. Furthermore, there is no client $v_i \neq \tilde{v}$ in this set, as this would contradict $\sigma((u_1, u_2))$ being a client equilibrium. To see this, note that were such a client equilibrium to exist, the v_i-excluded load on f_1 would be at most $2 - w(\tilde{v}) - w(v_i) \leq \frac{2}{\alpha} - w(v_i)$, while that on f_2 would be at least $\alpha - w(v_i)$, which is larger. For ease of notation, we define $y := \ell_2((u_1, u_2), \sigma)$. Then, since $y < 2 - w(I_2((u_1, u_2), \sigma))$, we find that $w(I_2((u_1, u_2), \sigma))$ is at most $2 - y$. Since $\tilde{v}$ is not adjacent to u_m, we conclude that the sum of the weights of the clients in $I(u_1, u_2)$ that are adjacent to u_m is at most $2 - y$.

Next, we establish lower bounds on $w(I(u_1, u_m))$ and $w(I(u_2, u_m))$. By Lemma 37 and $\ell_2((u_1, u_m), \sigma) \leq \frac{y}{\alpha}$:

$$\begin{aligned} w(I(u_1, u_m)) &= w(I_2((u_1, u_m), \sigma)) + w(I_1((u_1, u_m), \sigma)) \\ &\geq \rho(u_m) - \frac{y}{\alpha} + \max\left\{\rho(u_m) - \frac{y}{\alpha}, 2 - \rho(u_m)\right\} \\ &\geq \rho(u_m) + 2 - \frac{2y}{\alpha}. \end{aligned}$$

To establish the bound on $w(I(u_2, u_m))$, we note that some facility can sufficiently improve by performing the best-response move for facility placement profile (u_2, u_m). If this facility is f_1, then:

$$\ell_1((u_2, u_m), \sigma) < \frac{1}{\alpha}\ell_1((u_1, u_m), \sigma) \leq 1.$$

This implies that $w(I(u_2, u_m)) \geq y - \ell_1((u_2, u_m), \sigma) \geq y - 1$. If f_2 is the improving facility instead, then:

$$\ell_2((u_2, u_m), \sigma) \leq \frac{\ell_2((u_2, u_3), \sigma)}{\alpha} \leq \frac{2}{\alpha}$$

We apply Lemma 37 to find $w(I(u_2, u_m)) \geq y - \frac{2}{\alpha}$. Since $\frac{2}{\alpha} > 1$, we conclude that in either case, $w(I(u_2, u_m)) \geq y - \frac{2}{\alpha}$.

We now establish a lower bound on $w(I(u_1, u_m) \cup I(u_2, u_m))$ using the lower bounds on $w(I(u_1, u_m))$, $w(I(u_2, u_m))$ and upper bound on the sum of the weights of the clients in $I(u_1, u_2)$ that are adjacent to u_m. Recall that $y \geq \alpha$ to find:

$$\begin{aligned} w(I(u_1, u_m) \cup I(u_2, u_m)) &> \Big(\rho(u_m) + 2 - \frac{2y}{\alpha}\Big) + \Big(y - \frac{2}{\alpha}\Big) - \Big(2 - y\Big) \\ &= \rho(u_m) + 2y - \frac{2y + 2}{\alpha} \\ &= \rho(v_m) + 2y(1 - \frac{1}{\alpha}) - \frac{2}{\alpha} \\ &\geq \rho(v_m) + 2\alpha(1 - \frac{1}{\alpha}) - \frac{2}{\alpha} \\ &= \rho(v_m) + 2(\alpha - 1 - \frac{1}{\alpha}) \\ &\geq \rho(v_m), \end{aligned}$$

where the last inequality follows from $\alpha \geq \phi$. By definition, $\rho(v_m) \geq w(I(u_1, u_m) \cup I(u_2, u_m))$. We conclude that $\rho(v_m) > \rho(v_m)$; a contradiction. □

Theorem 19 illustrates how to use the developed theory to restrict the structure of α-instances. If the remaining open cases could similarly be eliminated for certain values of α, this would lead to an upper bound on the approximation constant for this class of 2-FLG. However, this is beyond the scope of this thesis.

9 Discussion and Conclusions

In this thesis, we considered subgame perfect equilibria in a model for the two-stage facility game with unsplittable clients. Unlike previous publications, the client behavior in our model allows for non-uniqueness of equilibria in subgames induced by facility placement profiles. This crucial distinction necessitated the use of different proof techniques. In particular, our efforts required characterizing several types of equilibria for these client games. Using these techniques, we demonstrated that instances of the two-stage facility location game do not generally admit subgame perfect equilibria and that deciding their existence is NP-hard. However, we identified two conditions under which they do exist: the existence of a balanced client equilibrium profile and the condition of equal client weights. Furthermore, we established a tight bound of 2 on the price of anarchy for the class of instances that admit subgame perfect equilibria.

Additionally, we developed an algorithm to find subgame perfect equilibria in unweighted instances of the two-stage facility location game. This algorithm utilizes improving move dynamics for the facility players. We have shown that cycling cannot occur and that each iteration can be computed in polynomial time. We conjecture that the number of iterations needed is always polynomial in the input size of the problem, although we were not able to prove this assertion. Consequently, we could not conclude whether it is possible to find a subgame perfect equilibrium in unweighted instances in polynomial time. Discovering a polynomial bound or identifying an instance requiring an exponential number of iterations would both yield interesting results.

Lastly, we investigated the existence of approximate equilibria, which proved to be a more challenging topic compared to "true" subgame perfect equilibria. Our focus was primarily on unrestricted instances with two facility players, as limiting the scope to this particular scenario kept the challenge somewhat manageable. We demonstrated that 2-approximate equilibria are guaranteed to exist for such instances. Additionally, we identified instances of this type that admit no α-approximate equilibria for α values strictly smaller than the golden ratio ϕ. Although we made substantial efforts to refine these bounds, we were unable to complete a proof. However, we conjecture that for instances with two facility players, ϕ-approximate equilibria always exist. Much remains unknown about the existence of approximate equilibria, including whether and how the number of facility players affects the values of α for which α-approximate equilibria are guaranteed to exist. We did prove that deciding whether a general instance of 2-FLG admits an α-approximate equilibrium, for α smaller than the golden ratio, is an NP-hard problem.

10 Suggestions for Further Research

The two main open problems presented in this paper offer ample opportunity for further research on the discussed version of the two-stage facility location game. Foremost is the running time of the algorithm for finding subgame perfect equilibria in unweighted 2-FLG. Throughout this thesis, our primary focus was on establishing the existence of equilibria, rather than designing efficient algorithms. Demonstrating that these equilibria can be found in polynomial time would thus likely indicate the existence of significantly faster algorithms compared to the one proposed in this report, even if the latter does turn out to have a polynomial running time. It might be possible for alternative algorithms to directly exploit the structure of the host graph to determine which facility placement profiles correspond to subgame perfect equilibria, bypassing the need for considering improving moves for the facility players. Such an algorithm would likely be orders of magnitude faster than ones based on improving moves. The availability of an efficient method for finding subgame perfect equilibria could make the application of this model to real-world problems more intriguing.

The other major open question pertains to identifying the values of α for which α-approximate equilibria are guaranteed to exist. Further results on this area would undeniably be of great interest, even if they are only applicable to specific classes of instances. Nevertheless, our experience has shown that obtaining such results is very challenging, even for the relatively simple case of unrestricted instances with two facility players. Therefore, time might be better spent researching other open problems, such as those of an algorithmic nature as previously mentioned. Should one attempt to tackle this challenge, we recommend solving the problem for just two facility players before extending the analysis to general instances of 2-FLG. The theory developed in this thesis can serve as a solid foundation for this process, as we believe our approach to be sound.

Additionally, exploring further alternative versions of the two-stage facility game would be interesting. The research on these games, to which this thesis is but the most recent addition, has already yielded some results. Investigating further variations could enhance our overall comprehension of such games, and help identify which conditions are sufficient for the existence of equilibria.

References

[1] Hee Kap Ahn, Siu Wing Cheng, Otfried Cheong, Mordecai Golin, and René Van Oostrum. "Competitive facility location: the Voronoi game". In: *Theoretical Computer Science* 310 (Jan. 2004), pp. 457–467. ISSN: 0304-3975. DOI: 10.1016/J.TCS.2003.09.004.

[2] Noga Alon, Michal Feldman, Ariel D. Procaccia, and Moshe Tennenholtz. "Strategyproof Approximation Mechanisms for Location on Networks". In: *ArXiv:* 0907.2049 (2009).

[3] Hau Chan, Aris Filos-Ratsikas, Bo Li, Minming Li, and Chenhao Wang. "Mechanism Design for Facility Location Problems: A Survey". In: *ArXiv:* 2106.03457 (2021).

[4] Y. Cheng, Wei Yu, and Guochuan Zhang. "Strategy-proof approximation mechanisms for an obnoxious facility game on networks". In: *Theor. Comput. Sci.* 497 (2013), pp. 154–163.

[5] Otfried Cheong, Sariel Har-Peled, Nathan Linial, and Jiří Matouek. "The One-Round Voronoi Game". In: *Discrete & Computational Geometry* 31 (2002), pp. 125–138.

[6] Stephen A. Cook. "The complexity of theorem-proving procedures". In: *Proceedings of the Annual ACM Symposium on Theory of Computing* (May 1971), pp. 151–158. ISSN: 07378017. DOI: 10.1145/800157.805047.

[7] Anthony Downs. "An Economic Theory of Political Action in a Democracy". In: *Journal of Political Economy* 65 (Apr. 1957), pp. 135–150. ISSN: 0022-3808. DOI: 10.1086/257897.

[8] Christoph Dürr and Kim Thang Nguyen. "Nash Equilibria in Voronoi Games on Graphs". In: *ArXiv:* cs/0702054 (2007).

[9] Jack Edmonds and Richard M. Karp. "Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems". In: *J. ACM* 19.2 (Apr. 1972), pp. 248–264. ISSN: 0004-5411. DOI: 10.1145/321694.321699. URL: https://doi.org/10.1145/321694.321699.

[10] Michal Feldman, Amos Fiat, and Iddan Golomb. "On voting and facility location". In: *EC 2016 - Proceedings of the 2016 ACM Conference on Economics and Computation* (July 2016), pp. 269–286. DOI: 10.1145/2940716.2940725.

[11] Michal Feldman and Yoav Wilf. "Strategyproof facility location and the least squares objective". In: *ACM Conference on Economics and Computation.* 2013.

[12] Gaëtan Fournier. "General distribution of consumers in pure Hotelling games". In: *International Journal of Game Theory* 48 (2016), pp. 33–59.

[13] Gilbert Laporte H. A. Eiselt and Jacques-Francois Thisse. "Competitive Location Models: A Framework and Bibliography". In: *Transportation Science* 27.1 (1993), pp. 44–54. ISSN: 00411655, 15265447.

[14] Harold Hotelling. "Stability in Competition". In: *The Economic Journal* 39 (Mar. 1929), p. 41. ISSN: 00130133. DOI: 10.2307/2224214.

[15] S. Keshav. *An Engineering Approach to Computer Networking: ATM Networks, the Internet, and the Telephone Network.* 1997.

[16] Elon Kohlberg. "Equilibrium store locations when consumers minimize travel time plus waiting time". In: *Economics Letters* 11 (Jan. 1983), pp. 211–216. ISSN: 0165-1765. DOI: 10.1016/0165-1765(83)90137-4.

[17] Simon Krogmann, Pascal Lenzner, Louise Molitor, and Alexander Skopalik. "Two-Stage Facility Location Games with Strategic Clients and Facilities". In: *IJCAI International Joint Conference on Artificial Intelligence* (2021), pp. 292–298. ISSN: 10450823. DOI: 10.24963/ijcai.2021/41.

[18] Simon Krogmann, Pascal Lenzner, and Alexander Skopalik. "Strategic Facility Location with Clients that Minimize Total Waiting Time". In: (Nov. 2022).

[19] Pinyan Lu, Xiaorui Sun, Yajun Wang, and Zeyuan Allen Zhu. "Asymptotically optimal strategy-proof mechanisms for two-facility games". In: *ACM Conference on Economics and Computation*. 2010.

[20] Mamta Mishra, Surya Prakash Singh, and M. P. Gupta. "Location of competitive facilities: a comprehensive review and future research agenda". In: *Benchmarking* (Apr. 2022). ISSN: 14635771. DOI: 10.1108/BIJ-11-2021-0638.

[21] Dov Monderer and Lloyd S. Shapley. "Potential Games". In: *Games and Economic Behavior* 14 (May 1996), pp. 124–143. ISSN: 0899-8256. DOI: 10.1006/GAME.1996.0044.

[22] John Nash. "Non-Cooperative Games". In: *Annals of Mathematics* 54 (1951), pp. 286–295. ISSN: 0003486X. DOI: 10.2307/1969529.

[23] Matías Núñez and Marco Scarsini. "Competing over a finite number of locations". In: *Economic Theory Bulletin* 4 (2015), pp. 125–136.

[24] Hans Peters. "Game Theory". In: (2015). DOI: 10.1007/978-3-662-46950-7.

[25] Hans Peters, Marc Schröder, and Dries Vermeulen. "Hotelling's location model with negative network externalities". In: *International Journal of Game Theory* 47 (2018), pp. 811–837.

[26] Ariel D. Procaccia and Moshe Tennenholtz. "Approximate mechanism design without money". In: *ACM Conference on Economics and Computation*. 2009.

[27] Robert W. Rosenthal. "A class of games possessing pure-strategy Nash equilibria". In: *International Journal of Game Theory* 2 (Dec. 1973), pp. 65–67. ISSN: 00207276. DOI: 10.1007/BF01737559.

[28] Amartya Sen. *Collective Choice and Social Welfare*. Harvard University Press, Dec. 2017. DOI: 10.4159/9780674974616/HTML.