Optimal Cost Sharing for Capacitated Facility Location Games

Tobias Harks[1] and Philipp von Falkenhausen*[2]

[1]*Maastricht University, Operations Research Group, School of Business and Economics,*
PO Box 616, 6200 MD, Maastricht, The Netherlands
`t.harks@maastrichtuniversity.nl`
[2]*Technische Universität Berlin, Institut für Mathematik MA 5-1,*
Straße des 17. Juni 136, 10623 Berlin, Germany.
`falkenhausen@math.tu-berlin.de`.

Abstract

We consider cost sharing for a class of facility location games, where the strategy space of each player consists of the bases of a player-specific matroid defined on the set of resources. We assume that resources have nondecreasing load-dependent costs and player-specific delays. Our model includes the important special case of capacitated facility location problems, where players have to jointly pay for opened facilities. The goal is to design cost sharing protocols so as to minimize the resulting price of anarchy and price of stability. We investigate two classes of protocols: basic protocols guarantee the existence of at least one pure Nash equilibrium and separable protocols additionally require that the resulting cost shares only depend on the set of players on a resource. We find optimal basic and separable protocols that guarantee the price of stability/price of anarchy to grow logarithmically/linearly in the number of players. These results extend our previous results (cf. [44]), where optimal basic and separable protocols were given for the case of *symmetric* matroid games *without* delays.

We finally study the complexity of computing optimal cost shares. We derive several hardness results showing that optimal cost shares cannot be approximated in polynomial time within a logarithmic factor in the number of players, unless $P = NP$. For a restricted class of problems that include the above hard instances, we devise an approximation algorithm matching the logarithmic bound.

Keywords: Game theory; Complexity theory

1 Introduction

Consider a setting where players jointly use resources that have load-dependent monetary costs and player-specific delays. The monetary cost of each resource must be shared by its users while

*Corresponding author, Phone +493031429400. This research was supported by the Deutsche Forschungsgemeinschaft within the research training group 'Methods for Discrete Structures' (GRK 1408).

the player-specific delays are unavoidable physical quantities. This setting arises in facility location models, where users share the monetary cost of opened facilities and additionally experience delays measured by the distance to the closest open facility. As players might value the delay and monetary cost differently, these delays depend on both, the player and the facility the player is assigned to. Another example appears in distributed network design, where players jointly install capacities on a subgraph satisfying user-specific connectivity requirements. Besides the monetary cost for installing enough capacity, each player experiences player-specific delays on the resources used.

Given the resource cost functions, delays and user's requirements, in an ideal world resources are allocated optimally, that is, an allocation minimizes the social costs. In distributed systems, however, players will selfishly select resources for their demands based on the cost shares they have to pay and delays they experience. While physical delays (such as switching times or travel times) are unavoidable, the ways the monetary cost of a resource is shared among its users determine the equilibrium states of the strategic game induced.

In this article, we study the design of cost sharing protocols as a means to induce efficient equilibria of the strategic game played. We consider cost sharing protocols axiomatized by the following three properties:

1. *Budget-balance*: The cost of each resource is exactly covered by the collected cost shares of the players using the resource.
2. *Stability*: There is at least one pure strategy Nash equilibrium in each game induced by the cost sharing protocol.
3. *Separability*: When assigning the cost shares on a given resource, the protocol has no information about the load on other resources.

We call a cost sharing protocol *basic* if it satisfies 1-2 and *separable* if it satisfies 1-3. These properties have been used first by Chen et al. [12] in the context of network design games. While condition 1 is straightforward, the stability condition 2 requires the existence of at least one Nash equilibrium in pure strategies (abbreviated PNE; see Osborne and Rubinstein [39, § 3.2] for several drawbacks of using mixed Nash equilibria instead). Condition 3 is for instance crucial for practical applications in which cost sharing protocols must be distributed because each resource has only local information about its own usage.

1.1 Our Results and Paper Organization

We study the design of cost sharing protocols for games in which every player wants to use resources that together form a basis of a player-specific matroid defined on the set of resources. We demonstrate that the aforementioned class of facility location games can be represented by player-specific matroids. As we assume general nondecreasing cost functions on the resources, our model also includes the case of facility location with hard capacities. Our setting has further applications in network design games, where each player wants to allocate a spanning tree in a graph.

Table 1: The results without reference are derived in this paper.

	Player-spec. Matroids		Symmetric Matroids without Delays	
	PoS	PoA	PoS	PoA
general cost	$\mathcal{H}_n$	n	$\mathcal{H}_n$ [44]	$\mathcal{H}_n$ [44]
concave cost	1	n	1	$\leq \mathcal{H}_n$ [44]
convex cost	1	1	1	1

Price of Anarchy and Stability. The main objective of this paper is to design basic or separable cost sharing protocols so as to minimize the efficiency loss of the equilibria induced. We consider the *price of anarchy* and the *price of stability* as the two prevailing performance metrics used in the literature. The price of anarchy (PoA) is defined as the worst-case ratio of the cost of a Nash equilibrium over the cost of a system optimum (see Koutsoupias and Papadimitriou [32]), while the price of stability (PoS) captures the ratio of the best possible Nash equilibrium over a system optimum (see Anshelevich et al. [3] and Correa et al. [16]).

The first main result presented in Section 3 is a characterization of Nash equilibria in games with general cost functions and strategy spaces defined by player-specific matroids, showing that only a subclass of strategy profiles (called *decharged*) are candidates for being a pure Nash equilibrium. This allows to give lower bounds by constructing instances where all decharged profiles are expensive. Our characterization of Nash equilibria using the notion of decharged profiles strictly generalizes a characterization introduced in our previous paper [44]. While in [44] we assumed symmetric strategy spaces, i.e., the same strategies are available to each player, we allow in this work *player-specific strategy spaces* and, additionally, individual *delay* for each player and resource.

As a second contribution, we give in Section 4 an algorithm that constructs decharged profiles establishing an optimal protocol with PoS matching our lower bounds. In Section 5, we further prove that this protocol is also optimal with respect to the PoA. Note that our protocol used for the positive results also fulfill the stricter separability requirement from [12], i.e., when assigning the cost shares on a given facility, the protocol has no information about the load on other facilities. When the class of cost functions is restricted to be either concave or convex, we show in Section 6 a drastic improvement of the PoS and PoA. The results in comparison to those obtained in [44] are summarized in Table 1.

Computational Complexity. In Section 7, we study the computational complexity of computing optimal cost shares that minimize the cost of the best/worst induced Nash equilibrium. We prove that both problems are strongly NP-hard and there are no polynomial time $c \log(n)$ approximations for any $0 < c < 1$, unless $P = NP$. These hardness results even hold for instances with unweighted players, zero delays, singleton strategies and unit fixed costs. In light of the hardness even for this restricted class of problems, we study approximation algorithms for the case of unweighted players, zero delays and singleton strategies, still assuming general nondecreasing costs. This setting includes several interesting classes of problems such as scheduling applications, where each player is associated with a job of unit weight. The job can be processed on a job-specific set of machines, and the monetary cost on a resource (for instance energy costs as in Yao et al. [45]) is a non-decreasing function of its total load. Another application arises in capacitated facility location with delays in $\{0, \infty\}$. We devise a polynomial time algorithm computing cost shares with an approximation guarantee of $\mathcal{H}_n$ and n for the problem of minimizing the cost of the best/worst Nash equilibrium, respectively.

1.2 Related Work

Cost sharing approaches to facility location problems and network design problems were analyzed in [31, 40]. In these works it is only required that total cost shares cover (approximately) total cost as the players play for the service of being connected. In contrast in our work, we require the stricter notion that the cost of every individual resource is paid for by the players using it. Our model also includes the case of the congested facility location problem considered by Desrochers et al. [17], where players have to jointly pay for the congestion related costs and the opening costs of the facilities. Instead of a game-theoretic problem formulation, they consider a centralized approach using mixed-integer programming techniques for minimizing total cost.

The existence of pure Nash equilibria and their price of anarchy and price of stability in congestion games has been studied by several researchers, [1, 20, 21, 22, 23, 26, 33]. The fundamental difference from our work is the ways cost shares are assigned. In congestion games, every player pays the average total cost, while in our setting we follow a design perspective of cost sharing protocols. Roughgarden and Kollias [30] considered weighted congestion games and proposed a cost sharing protocol based on the Shapley value for which they are able to prove existence of PNE and corresponding bounds on the price of anarchy/stability. They focus on polynomial cost per unit functions with nonnegative coefficients but it is not known that this protocol is optimal. For further work on congestion games with average cost sharing assuming nonincreasing marginal cost functions modeling economies of scale or buy-at-bulk we refer to [2, 3, 19, 25, 42]. There is also a large body of papers studying cost sharing protocols for continuous and convex strategy spaces assuming convex cost sharing functions, cf. [13, 24, 28, 29, 35, 36].

Christodoulou et al. [14] and follow-up papers such as [9, 15, 27] study scheduling policies and their price of anarchy in scheduling games in which each player assigns a job to machine with the goal to minimize its completion time. Since the completion time of a job depends on the order in which jobs are processed the design of scheduling policies is fundamentally different to the design of cost sharing protocols.

Cost sharing is a central topic in the area of cooperative game theory, cf. [4, 8, 37] or the survey of Moulin [34] with a pointer to further references.

2 Model and Problem Statement

In a *facility location model*, there is a set $N = \{1, \dots, n\}$ of players that choose facilities (also called resources) from a set $R = \{r_1, \dots, r_m\}$. For each player $i \in N$, there is a set Σ_i of feasible resource choices, where a strategy $X_i \in \Sigma_i$ is a set of resources, i.e. $X_i \subseteq R$, and there is a matroid

$\mathcal{M}_i = (R, \mathcal{I})$ such that Σ_i equals the set of bases of $\mathcal{M}_i$. As a set of matroid bases, Σ_i has the *basis exchange property*: given a strategy from Σ_i, one can alter it resource by resource towards another strategy from Σ_i such that all intermediate strategies are also contained in Σ_i. In other words, for any strategies $X_i, Y_i \in \Sigma_i$ and any $r \in X_i$ there is some $r' \in Y_i \setminus X_i$, $r \neq r'$, such that $X_i \setminus \{r\} \cup \{r'\} \in \Sigma_i$ (cf. Schrijver [43] for a comprehensive introduction into matroid theory). Given strategies X_i for all players $i \in N$, we denote by $X := (X_1, \ldots, X_n)$ the joint strategy profile and correspondingly $\Sigma := \prod_{i \in N} \Sigma_i$.

Furthermore, a vector $d = (d_i)_{i \in N}$ specifies the player's weights, which in a strategy profile X sum up on each resource $r \in R$ to the load $\ell_r(X) := \sum_{i \in N_r(X)} d_i$, where $N_r(X) := \{i : r \in X_i\}$ is the set of players using r. The resources' cost functions are given by a vector $c = (c_r)_{r \in R}$, they are non-decreasing in the load. Each player i incurs a player-specific delay $t_{i,r} \geq 0$ when using resource r. Altogether, a *facility location model* is represented by a tuple $I = (N, R, \Sigma, d, c, t)$.

To measure the social cost of a profile X, we use utilitarian welfare, i.e. the sum of the individual players' costs and delays, and we denote $C(X) := \sum_{r \in R} c_r(\ell_r(X)) + \sum_{i \in N : r \in X_i} t_{i,r}$. Abusing notation, we often refer to the cost on resource r by $c_r(X)$ instead of $c_r(\ell_r(X))$. We now give two examples of such models.

Example 2.1 (Capacitated Facility Location Games)**.** *In a capacitated facility location game, motivated by the classic* Facility Location *problem, every player i chooses exactly one resource, that is* $|X_i| = 1$ *for all* $X_i \in \Sigma_i$ *and* $i \in N$ *and hence* $\mathcal{M}_i$ *corresponds to a uniform matroid of rank one. Hard capacities on the facilities can be modeled by cost functions that sharply increase at the corresponding capacity.*

Example 2.2 (MST Games)**.** *We are given an undirected graph* $G = (V, E)$ *with non-negative and non-decreasing edge cost functions* $c_e(\ell), e \in E$. *In a minimum spanning tree (MST) game, every player i is associated with demand of size* $d_i > 0$ *and a subgraph* G_i *of* G. *A strategy for player is then to route its demand along a spanning tree for* G_i. *Formally, we set* $R = E$ *and the sets* X_i, $i \in N$, *are the spanning trees of* G_i, *hence* $\mathcal{M}_i$ *corresponds to the graphical matroid.*

We study how different ways of sharing the costs of a resource affect the resulting pure Nash equilibria of the induced game. To model this, we introduce cost sharing protocols Ξ that assign cost share functions $\xi_{i,r} : \Sigma \to \mathbb{R}$ for all $i \in N$ and $r \in R$ to the facility location model I and thus induce the strategic game (N, Σ, ξ). For a player i, her total private cost is $\xi_i(X) := \sum_{r \in X_i} \xi_{i,r}(X) + t_{i,r}$ and we assume that every player strives to minimize her private cost. An important solution concept in non-cooperative game theory are pure Nash equilibria. Using standard notation in game theory, for a strategy profile $X \in \Sigma$ we denote by

$$(Z_i, X_{-i}) := (X_1, \ldots, X_{i-1}, Z_i, X_{i+1}, \ldots, X_n) \in \Sigma$$

the profile that arises if only player i deviates to strategy $Z_i \in \Sigma_i$.

Definition 2.3 (Pure Nash Equilibrium (PNE))**.** *Let* (N, Σ, ξ) *be a strategic game. The profile X is a pure Nash equilibrium if no player i can strictly reduce her private cost by unilaterally moving to a different strategy, that is, for all* $i \in N$

$$\xi_i(X) \leq \xi_i(Z_i, X_{-i}) \text{ for all } Z_i \in \Sigma_i.$$

For facility location games with cost sharing protocol ξ, this translates to: a strategy profile X is a Nash equilibrium if

$$\sum_{r\in X_i}(\xi_{i,r}(X)+t_{i,r}) \leq \sum_{r\in Z_i}(\xi_{i,r}(Z_i,X_{-i})+t_{i,r}) \quad \textit{for all strategies } Z_i \in \Sigma_i \textit{ and players } i \in N.$$

Two well established concepts that quantify the efficiency of Nash equilibria are the *price of anarchy* and the *price of stability*. The price of anarchy measures the largest possible ratio of the cost of a Nash equilibrium and the cost of an optimal profile. The price of stability measures the smallest ratio of the cost of a Nash equilibrium and the cost of an optimal profile. For a cost sharing protocol Ξ, we define by $PoA(\Xi)$ and $PoS(\Xi)$ the corresponding worst case price of anarchy and price of stability across games induced by protocol Ξ. The main goal of this paper is to design cost sharing protocols that minimize the price of anarchy and price of stability, respectively. Of course, the attainable objective values crucially depend on the design space that we permit. The following properties have been first proposed by Chen et al. [12] in the context of designing cost sharing protocols for network design games.

Definition 2.4 (Properties of Cost Sharing Protocols)**.** *A cost sharing protocol Ξ is*

1. stable *if it induces only games that admit at least one pure Nash equilibrium.*
2. basic *if it is stable and additionally* budget balanced*, i.e. if it assigns all facility location models (N,R,Σ,d,c,t) with cost share functions $\xi_{i,r}$ such that for all $r\in R$ and $X\in\Sigma$*

   $$c_r(X) = \sum_{i\in N_r(X)} \xi_{i,r}(X) \textit{ and } \xi_{i,r}(X) = 0 \textit{ for all } i \notin N_r(X).$$

   This property requires $c_r(0) = 0$ for unused resources, which we will assume in the paper.
3. separable *if it is basic and if it induces only games for which in any two profiles $X, X' \in \Sigma$ for every resource $r \in R$,*

   $$N_r(X) = N_r(X') \Rightarrow \xi_{i,r}(X) = \xi_{i,r}(X') \textit{ for all } i \in N_r(X).$$

Informally, separability means that in a profile X the values $\xi_{i,r}(X),\ i\in N$ depend only on the set $N_r(X)$ of players sharing resource r and disregard all other information contained in X. Still, separable protocols can assign cost share functions that are specifically tailored to the given facility location model, for example based on an optimal profile. We denote by $\mathcal{B}_n$ and $\mathcal{S}_n$ the set of basic and separable protocols for facility location games with n players, respectively. We obtain the following optimization problems that we address in this paper.

$$\min_{\Xi\in\mathcal{B}_n} PoA(\Xi),\ \min_{\Xi\in\mathcal{B}_n} PoS(\Xi),\ \min_{\Xi\in\mathcal{S}_n} PoA(\Xi),\ \min_{\Xi\in\mathcal{S}_n} PoS(\Xi).$$

3 Characterizing Pure Nash Equilibria

Before we analyze the cost of Nash equilibria, we reveal two structural properties of PNE in the context of facility location games. For this, denote the cost of the cheapest alternative of player i to resource r in a profile $Z \in \Sigma$ by

$$\Delta_i^r(Z) := \min_{\substack{s \in R \\ Z_i + s - r \in \Sigma_i}} (c_s(Z_i + s - r, Z_{-i}) + t_{i,s}).$$

Lemma 3.1. *A PNE X in a facility location game with a separable protocol fulfills the following two properties:*

1. *Each player i is willing to pay the delay $t_{i,r}$ for each chosen resource $r \in X_i$:*
$$t_{i,r} \leq \Delta_i^r(X). \tag{D1}$$
2. *The users $N_r(X)$ of each resource r, after having paid their delays, are willing to share the cost of the resource:*
$$c_r(X) \leq \sum_{i \in N_r(X)} \left(\Delta_i^r(X) - t_{i,r}\right). \tag{D2}$$

Note that (D1) *implies that each summand $\Delta_i^r(X) - t_{i,r}$ in* (D2) *is nonnegative.*

We call a strategy profile X that fulfills (D1) and (D2) *decharged*, regardless of the cost shares. Profiles that are not decharged are called *charged*, similarly resources are called decharged or charged depending on whether both (D1) and (D2) are fulfilled for all players using the resource.

Proof. We first show (D1). Let X be a Nash equilibrium under a separable protocol ξ, $i \in N$ and $r \in X_i$. Let $s := \operatorname{argmin}_{s \in R : X_i + s - r \in \Sigma_i} c_s(X_i + s - r, X_{-i}) + t_{i,s}$ be the cheapest alternative for i to r and let $Z_i := X_i + s - r$. Then the Nash inequality for player i can be restated as

$$\begin{aligned} t_{i,r} &\leq t_{i,r} + \sum_{r* \in Z_i} (\xi_{i,r*}(Z_i, X_{-i}) + t_{i,r*}) - \sum_{r* \in X_i} (\xi_{i,r*}(X) + t_{i,r*}) \\ &= \xi_{i,s}(Z_i, X_{-i}) + t_{i,s} - (\xi_{i,r}(X) + t_{i,r}) + t_{i,r} \leq c_s(Z_i, X_{-i}) + t_{i,s} = \Delta_i^r(X). \end{aligned} \tag{1}$$

For (1) observe that going from X_i to Z_i only r and s are exchanged and since ξ is separable the cost shares for all other resources remain the same, i.e. $\xi_{i,r^*}(X) = \xi_{i,r^*}(Z_i, X_{-i})$ for $r^* \in X_i \cap Z_i$.

We now show (D2), again using the Nash inequality for player i,

$$\begin{aligned} c_r(X) &= \sum_{i \in N_r(X)} \xi_{i,r}(X) \leq \sum_{i \in N_r(X)} \left(\min_{\substack{s \in R \\ X_i + s - r \in \Sigma_i}} (\xi_{i,s}(X_i + s - r, X_{-i}) + t_{i,s}) - t_{i,r} \right) \\ &\leq \sum_{i \in N_r(X)} \left(\min_{\substack{s \in R \\ X_i + s - r \in \Sigma_i}} (c_s(X_i + s - r, X_{-i}) + t_{i,s}) - t_{i,r} \right) = \sum_{i \in N_r(X)} \left(\Delta_i^r(X) - t_{i,r}\right) \end{aligned} \tag{2}$$

Here, (2) comes from budget-balance. □

Not only are all PNE decharged, but we can also find cost shares for any decharged strategy profile such that it is a PNE.

Definition 3.2 (X-enforcing cost shares)**.** *Given a facility location model and and a decharged strategy profile X, cost shares ξ are called X-enforcing if*

$$\xi_{i,r}(X) = \frac{\Delta_i^r(X) - t_{i,r}}{\sum_{j \in N_r(X)} \Delta_j^r(X) - t_{j,r}} \cdot c_r(X),$$

$$\xi_{i,r}(Z_i, X_{-i}) = c_r(Z_i, X_{-i}) \quad \textit{if } r \in Z_i \setminus X_i \textit{ for any } Z_i \in \Sigma_i.$$

That is, in X the cost of each resource is shared among the users proportional to their cheapest alternatives minus the delay and a unilaterally deviating player pays the entire cost.

There always exists a separable protocol with such cost shares as we show in Section 4.

Lemma 3.3. *Given a decharged profile X and X-enforcing cost shares ξ, X is a PNE.*

Proof. To establish that X is a PNE, we first show that no player can improve by unilaterally exchanging a single resource in his strategy.

$$\begin{aligned} \xi_{i,r}(X) + t_{i,r} &= \frac{\Delta_i^r(X) - t_{i,r}}{\sum_{j \in N_r(X)} \Delta_j^r(X) - t_{j,r}} \cdot c_r(X) + t_{i,r} \\ &\leq \Delta_i^r(X) - t_{i,r} + t_{i,r} = \Delta_i^r(X) \end{aligned} \tag{3}$$

where (3) holds because for decharged profiles we have (D2) and hence the denominator of the cost share is not smaller than $c_r(X)$. Using the basis exchange property of matroids as introduced at the beginning of Section 2, we now conclude that no player can improve by changing his strategy to an arbitrary $Z_i \in \Sigma_i$ and thus that X is a PNE. To this end, fix such a $Z_i \in \Sigma_i$ and denote by $G(X_i \triangle Z_i)$ the bipartite graph (V, E) with $V := (X_i \backslash Z_i) \cup (Z_i \backslash X_i)$ and $E := \{(r, s) : r \in X_i \backslash Z_i, s \in Z_i \backslash X_i, (X_i + s - r) \in \Sigma_i\}$.

Proposition 3.4 (Schrijver '03 [43])**.** *There exists a perfect matching in the graph $G(X_i \triangle Z_i)$.*

Consider such a matching and observe that, as shown above, no player can improve unilaterally by exchanging a single resource across a matching edge (r, s) with $r \in X_i$, $s \in Z_i$,

$$\xi_{i,r}(X) + t_{i,r} \leq \xi_{i,s}(X_i + s - r, X_{-i}) + t_{i,s} = \xi_{i,s}(Z_i, X_{-i}) + t_{i,s},$$

where in the last step we use that player i's cost share on s is independent of the other resources. Summing this up across all matching edges yields the desired

$$\xi_i(X) + \sum_{r \in X_i} t_{i,r} = \sum_{r \in X_i} (\xi_{i,r}(X) + t_{i,r}) \leq \sum_{s \in Z_i} (\xi_{i,s}(Z_i, X_{-i}) + t_{i,s}) = \xi_i(Z_i, X_{-i}) + \sum_{s \in Z_i} t_{i,r}.$$

□

Theorem 3.5 (Characterization of PNE for separable protocols)**.** *A profile $X \in \Sigma$ is a pure Nash equilibrium in the game induced by some separable protocol if and only if it is decharged.*

Proof. Follows from Lemma 3.1 and Lemma 3.3, the existence of the separable protocol is the content of Section 4. □

4 An Optimal Protocol for the Price of Stability

In this section, we deal with the existence and cost of decharged strategy profiles, which by Theorem 3.5 correspond to the (possible) PNE of a facility location game. We present an algorithm that 'decharges' any strategy profile of a facility location model, increasing the cost in the process by at most a factor $\mathcal{H}_n$. We then give a separable protocol that is X-enforcing for the output of the algorithm. Using this protocol and starting the algorithm at the optimal strategy profile of a facility location model gives an upper bound on the Price of Stability. The proof techniques used here resemble at a high level those used in [44]. However, player-specific strategy spaces and the existence of delays in addition to the facility costs require extra care. While in the setting of [44] decharged profiles are characterized by (D2), in this setting the algorithm needs ensure that also (D1) is satisfied. We start the section with a known lower bound.

Lemma 4.1 (von Falkenhausen and Harks [44])**.** *There price of stability for symmetric singleton games without transport induced by basic or separable protocols is at least $\mathcal{H}_n$.*

These games are facility location games with the restrictions that $\Sigma_i = \Sigma_j = R$, i.e. the strategies of the players are single resources, and $t_{i,r} = 0$ for all $i, j \in N$ and $r \in R$. Consequently, the Price of Stability for facility location games induced by basic or separable protocols is at least $\mathcal{H}_n$.

We now introduce Algorithm 1. The algorithm returns for every instance (N, R, Σ, d, c, t) and input profile Y a decharged profile X that costs no more than $\mathcal{H}_n \cdot C(Y)$, as shown in the upcoming lemmas.

The algorithm iteratively moves players away from charged resources. If there is a player that is not even willing to pay her delay (case called *Transp*, lines 5-6), this player is immediately moved away. Otherwise, if there are players that have been moved before (case called *Shuffle*, lines 7-8) these players are moved in a last-in-first-out order. Finally, if all players are willing to pay a non-negative cost share and non of them have been moved before (case called *Kickoff*, lines 9-11), the one who is willing to pay the least is moved. Each time, the selected player is moved to the best available (1,1)-exchange (line 14). To show that the algorithm works as desired, we prove two lemmas: First, we show that the algorithm terminates, then we deal with the cost of profile X.

Lemma 4.2. *Algorithm 1 terminates.*

Proof. For the proof, we adhere to the interpretation of each player i scheduling $|Y_i|$ jobs on the resources. We fix a player and follow one of his jobs over the course of the algorithm. The job can be moved at most once by a Kickoff, afterwards multiple times by Transp and Shuffle moves. We show that these moves strictly decrease the cost of the job independent of what happens in between the moves. More precisely, if a job of player i is moved in iteration k of the algorithm to resource s^k and stays there until iteration l (i.e., $s^k = r^l$) when it is moved to resource s^l, we show that $c_{s^k}(X^{k+1}) + t_{i,s^k} > c_{s^l}(X^{l+1}) + t_{i,s^l}$. Since there are only finitely many values for the cost of the job, this proves that the algorithm terminates.

If the move in iteration l is a Transp move, then the cost of the delay t_{i,r^l} is greater than the

Algorithm 1 Find decharged profile X

```
Input: Facility location model $(N, R, \Sigma, d, c, t)$, profile $Y$
Output: Decharged profile $X$
$k \leftarrow 1$    {step number}
$X^1 \leftarrow Y$    {starts with profile $Y$}
while there are charged resources do
    select the most expensive charged resource $r^k \leftarrow \operatorname{argmax}\{c_r(X^k) : r \in R \text{ is charged}\}$
    if (D1) not fulfilled by all players on $r^k$ then    {case called Transp}
        select such a player $i^k \in N_{r^k}(X^k)$ with $t_{i^k,r^k} > \Delta_{i^k}^{r^k}$
    else if some player on $r^k$ was moved before then    {case called Shuffle}
        select player $i^k \in N_{r^k}(X^k)$ that was moved last
    else    {case called Kickoff}
        select player who is willing to pay the least
        $i^k \leftarrow \operatorname*{argmin}_{i \in N_{r^k}(X^k)} \Delta_{i^k}^{r^k} - t_{i,r^k}$
    end if
    select cheapest (1,1)-exchange $s^k \leftarrow \operatorname*{argmin}_{\substack{s \in R \\ X_{i^k}^k + s - r^k \in \Sigma_{i^k}}} c_s(X_{i^k}^k + s - r^k, X_{-i^k}^k) + t_{i,s}$
    execute (1,1)-exchange $X^{k+1} \leftarrow (X_{i^k}^k + s^k - r^k, X_{-i}^k)$
    iterate $k \leftarrow k + 1$
end while
Return $X \leftarrow X^k$
```

cheapest (1,1)-exchange and we have

$$c_{s^k}(X^{k+1}) + t_{i,s^k} \geq t_{i,s^k} = t_{i,r^l} > \Delta_i^{r^l}(X^l) = c_{s^l}(X^{l+1}) + t_{i,s^l}. \tag{4}$$

If on the other hand the move in iteration l is a Shuffle move, the last-in-first-out scheme of Shuffles ensures that $N_{r^l}(X^{k+1}) \supseteq N_{r^l}(X^l)$ and hence

$$\begin{aligned} c_{s^k}(X^{k+1}) &\geq c_{r^l}(X^l) \\ &> \sum_{j \in N_{r^l}(X^l)} \left(\Delta_j^{r^l}(X^l) - t_{j,r^l}\right) \qquad (5) \\ &\geq \Delta_i^{r^l}(X^l) - t_{i,r^l} = c_{s^l}(X^{l+1}) + t_{i,s^l} - t_{i,r^l}. \qquad (6) \end{aligned}$$

When the algorithm does a Shuffle on resource r^l, all users of the resource fulfill (D1) (otherwise a Transp would be done), but the resource does not fulfill (D2) (otherwise it would be decharged and hence be disregarded by the algorithm). Not fulfilling (D2) leads to (5) and since every user fulfills (D1) all summands in (5) are nonnegative, which leads to (6). □

Lemma 4.3. *Profile X returned by Algorithm 1 has at most $\mathcal{H}_n$ times the cost of the input profile Y.*

Proof. Throughout the proof of this claim, we regard the set $Q := \{q_1, \dots\}$ of all jobs instead of the players they belong to. We denote the player to which a job q belongs by $i(q)$ and the resource on which job q is scheduled in profile Y by $y(q)$. For ease of exposition, we often use jobs q interchangeably with $i(q)$, e.g. $t_{q,r}$ as a shorthand for $t_{i(q),r}$. If k is the iteration of the algorithm in which q is first moved by the algorithm, we define $p(q) := |N_{y(q)}(X^k)|$ to be the number of players on $y(q)$ in X^k. This first move can either be a Transp or a Kickoff. For a Transp, we have already seen in (4) that $c_{s^k}(X^{k+1}) + t_{q,s^k} < t_{q,y(q)}$. If the first move is a Kickoff, then there are no foreign players on $y(q)$, hence $N_{y(q)}(X^k) \subseteq N_{y(q)}(Y)$ and we have

$$\begin{aligned} c_{y(q)}(Y) &\geq c_{y(q)}(X^k) \\ &> \sum_{j \in N_{y(q)}(X^k)} \left(\Delta_j^{y(q)}(X^k) - t_{j,y(q)} \right) && (7) \\ &\geq p(q) \cdot \left(\Delta_q^{y(q)}(X^k) - t_{q,y(q)} \right) = p(q) \cdot \left(c_{s^k}(X^{k+1}) + t_{q,s^k} - t_{q,y(q)} \right). && (8) \end{aligned}$$

(7) stems like (5) from the fact that in a Kickoff, the resource $y(q)$ does not fulfill (D2). Since all summands are positive (like in (5)) and the summand corresponding to job q is the smallest of the $p(q)$ summands (see line 11 of the algorithm), we have inequality (8).

Altogether, the first time a job is moved by the algorithm, we have

$$c_{s^k}(X^{k+1}) + t_{q,s^k} \leq \frac{1}{p(q)} \cdot c_{y(q)}(Y) + t_{q,y(q)}$$

regardless of whether this move is a Kickoff or a Transp. Further Transp and Shuffle moves only reduce this quantity, hence for every further iteration $\bar{k}$ of the algorithm where job q is on resource $\bar{r}$ and no job has been moved there after him, we have

$$c_{\bar{r}}(X^{\bar{k}}) + t_{q,\bar{r}} \leq \frac{1}{p(q)} \cdot c_{y(q)}(Y) + t_{q,y(q)}. \tag{9}$$

Particularly, for resources where jobs have been moved by the algorithm, the cost in the final profile X returned by the algorithm is determined by the last job that was moved to the resource in the sense of (9). In the following, we estimate the cost of such resoures by summing up (9) over all jobs on the resource X. For resources where no job was moved, the cost is no greater than in the original profile Y. Thus,

$$\begin{aligned} C(X) = &\sum_{\substack{r \in R \\ \text{jobs moved to } r}} \left(c_r(X) + \sum_{q \in N_r(X)} t_{q,r} \right) + \sum_{\substack{r \in R \\ \text{no jobs moved to } r}} \left(c_r(X) + \sum_{q \in N_r(X)} t_{q,r} \right) \\ \leq &\sum_{\substack{r \in R \\ \text{jobs moved to } r}} \sum_{q \in N_r(X)} \left(\frac{1}{p(q)} \cdot c_{y(q)}(Y) + t_{q,y(q)} \right) + \sum_{\substack{r \in R \\ \text{no jobs moved to } r}} \left(c_r(Y) + \sum_{q \in N_r(X)} t_{q,y(q)} \right) \\ \leq &\sum_{r \in R} \sum_{\substack{q \in N_r(Y) \\ q \text{ moved}}} \frac{1}{p(q)} \cdot c_r(Y) + \sum_{\substack{r \in R \\ \text{jobs remained on } r}} c_r(Y) + \sum_{\text{all jobs } q} t_{q,y(q)} && (10) \\ \leq &\sum_{r \in R} \mathcal{H}_n \cdot c_r(Y) + \sum_{\text{all jobs } q} t_{q,y(q)} \leq \mathcal{H}_n \cdot C(Y). && (11) \end{aligned}$$

In (10) we change the order of summation: before we grouped the jobs by the resource they are on in X, in (10) they are grouped by the resource they are on in Y. For each resource r the $\frac{1}{p(q)}$ fractions sum up to at most $\mathcal{H}_n$ or, if jobs remained on r, to at most $\mathcal{H}_n - 1$ by definition of $p(q)$, hence (11). □

Corollary 4.4. *Every facility location model has a decharged strategy profile X at cost $C(X) \leq \mathcal{H}_n \cdot C(Y)$ where Y is the cost-optimal strategy profile.*

Proof. Follows from Lemma 4.2 and Lemma 4.3 □

We now use the algorithm together with our insights about X-enforcing cost shares to construct a protocol that matches our lower bound for the Price of Anarchy. For this, we assume without loss of generality that players are indexed by non-increasing weights $d_1 \geq d_2 \geq \cdots \geq d_n$.

Definition 4.5 (Enforcing Protocol). *For any facility location model (N, R, Σ, d, t, c), the Enforcing Protocol runs Algorithm 1 with a cost-optimal profile Y as input to obtain a decharged profile X with cost $C(X) \leq \mathcal{H}_n \cdot C(Y)$. Then, define for any profile Z and resource r the set of* foreign players $N_r^1(Z) := N_r(Z) \backslash N_r(X)$ *and assign the cost share functions*

$$\xi_{i,r}(Z) := \begin{cases} \dfrac{\Delta_i^r(X) - t_{i,r}}{\sum\limits_{j \in N_r(X)} \Delta_j^r(X) - t_{j,r}} \cdot c_r(X), & \text{if } N_r(Z) = N_r(X) \text{ and } c_r(X) > 0, \\ c_r(Z), & \text{if } N_r^1(Z) \neq \emptyset \text{ and } i = \min N_r^1(Z), \\ c_r(Z), & \text{if } N_r^1(Z) = \emptyset,\ N_r(Z) \subset N_r(X) \text{ and } i = \min N_r(Z), \\ 0, & \text{else.} \end{cases}$$

One can easily verify that the protocol is budget-balanced and separable. The cost shares are X-enforcing and hence the Price of Stability for the Enforcing Protocol is $\mathcal{H}_n$.

Theorem 4.6. *The Price of Stability for facility location games induced by basic and separable protocols is $\mathcal{H}_n$.*

Proof. The lower bound is given by Lemma 4.1. One can easily verfiy that the Enforcing Protocol is budget-balanced and separable. Furthermore, it is X-enforcing for a decharged profile X returned by Algorithm 1 with $C(X) \leq \mathcal{H}_n \cdot C(Y)$ and hence the Price of Stability for the Enforcing Protocol is $\mathcal{H}_n$. □

5 An Optimal Protocol for the Price of Anarchy

We now turn to the case of finding an optimal protocol for minimizing the resulting price of anarchy. We will prove that the Enforcing Protocol induces a price of anarchy of at most n. We further show that no basic protocol can have a price of anarchy below n, thus, the Enforcing Protocol is optimal.

Lemma 5.1. *The Price of Anarchy for facility location games with basic or separable cost sharing protocols is at least n.*

Proof. Consider the facility location model (N, R, Σ, d, c, t) with n players $i \in N$, resource set $R = \{r_0, r_1, \dots, r_n\}$, strategy spaces $\Sigma_i = \{(r_0), (r_i)\}$, weights $d_i = 1$ for all $i \in N$, delays $t_{i,r} = 0$ for all $i \in N, r \in R$ and constant resource cost $c_r \equiv 1$ independent of the load for all $r \in R$. Here, the optimal profile $Y = (r_0, \dots, r_0)$ hast cost $C(Y) = 1$. However, the profile $X = (r_1, \dots, r_n)$ with cost $C(X) = n$ is a Nash equilibrium under any protocol that satisfies budget-balance. □

Note that for this lower bound a singleton model without delays is sufficient. It can equivalently be constructed with a symmetric singleton model ($\Sigma_i = \Sigma_j$ for all $i, j \in N$) with delays. However, for symmetric singleton games without delays the Price of Anarchy has been shown to be $\mathcal{H}_n$ [44].

Lemma 5.2. *The Price of Anarchy for facility location games with basic or separable protocols is at most n.*

Proof. We only give a sketch of the proof as it is largely similar to the one found in [44]. A complete version can be found in the appendix.

Let (N, R, Σ, d, c, t) be a facility location model. Let $\xi_{i,r}$ for $i \in N, r \in R$ be the cost share functions assigned by the Enforcing Protocol and let X be the decharged profile returned by Algorithm 1 for the protocol, using some optimal profile Y as input. We show $C(Z) \leq n \cdot C(Y)$ for any pure Nash equilibrium Z. In a first step, we link the cost in Z to the cost in profiles (X_i, Z_{-i}) via the Nash property. The major challenge of the proof is then to estimate the cost of these profiles in relation to the cost of Y. To this end we employ properties of X found in the analysis of the algorithm in the previous section. □

6 Concave and Convex Costs

We now restrict the set of cost functions to be either concave or convex. Concave functions are frequently used to model economies of scale, that is, situations in which marginal costs are decreasing. Examples include network design games, where cost functions are modeled by fixed or concave costs, cf. [3, 7, 10, 11]. On the other hand, convex costs are used to model the sharp increase of costs if resources are scarce or if there are hard capacities on the amount of resources available. For instance in telecommunication networks, relevant cost functions are the so-called $M/M/1$-delay functions (see [6, 38]). These are convex functions of the form $c_a(x) = 1/(u_a - x)$, where u_a represents the physical capacity of arc a.

6.1 Concave Cost Functions

For concave cost functions, we now show that for facility location games, an optimal strategy profile is decharged, thus, the price of stability is one in this case.

Theorem 6.1. *The price of stability of facility location games with concave costs is one.*

Proof. We prove that any socially optimal configuration profile $X \in \Sigma$ satisfies (D1) and (D2) and, thus, by Theorem 3.5 the enforcing protocol induces X as a pure Nash equilibrium. Assume that X does not satisfy (D1). Then we can simply reassign player i to her cheapest alternative and strictly reduce the total cost, which is a contradiction to our assumption that X is socially optimal.

Assume that X does not satisfy (D2). Hence, there is a resource $r \in R$ with

$$c_r(X) > \sum_{i \in N_r(X)} \Delta_i^r(X) - t_{i,r}. \tag{12}$$

We now construct a strategy profile Z with $C(Z) < C(X)$, showing a contradiction. To this end, for each player $i \in N_r(X)$, let

$$s_i := \operatorname*{argmin}_{\substack{s \in R \\ X_i + s - r \in \Sigma_i}} \left(c_s(X_i + s - r, X_{-i}) + t_{i,s}\right)$$

be the cheapest alternative to r in X. Let $Z_i := X_i + s_i - r$ be the strategy where i switches from r to s_i, such that $\Delta_i^r(X) = c_{s_i}(Z_i, X_{-i}) + t_{i,s_i}$. For $i \notin N_r(X)$, let $Z_i := X_i$.

Inequality (12) implies that $s_i \neq r$ for all $i \in N_r(X)$ and hence that $N_r(Z) = \emptyset$, i.e. $c_r(Z) = 0$. Resources $s \neq r$ either have the same users in Z as in X, i.e. $c_s(Z) = c_s(X)$, or they have users $N_s(Z) \setminus N_s(X)$ that have switched from r to s. In the latter case, as the cost functions are concave,

$$c_s(Z) \leq \sum_{i \in N_s(Z) \setminus N_s(X)} c_s(Z_i, X_{-i}) = \sum_{i \in N_s(Z) \setminus N_s(X)} \left(\Delta_i^r(X) - t_{i,s}\right). \tag{13}$$

Combining these observations, we conclude

$$\begin{aligned} C(Z) = & \sum_{\substack{s \in R \\ N_s(Z) = N_s(X)}} \left(c_s(Z) + \sum_{i \in N_s(Z)} t_{i,s}\right) + \sum_{\substack{s \in R \\ N_s(Z) \supset N_s(X)}} \left(c_s(Z) + \sum_{i \in N_s(Z)} t_{i,s}\right) \\ \overset{(13)}{\leq} & \sum_{\substack{s \in R \\ N_s(Z) = N_s(X)}} \left(c_s(X) + \sum_{i \in N_s(X)} t_{i,s}\right) + \sum_{i \in N_r(X)} \Delta_i^r(X) + \sum_{\substack{s \in R \\ N_s(Z) \supset N_s(X)}} \sum_{i \in N_r(X)} t_{i,s} \\ \overset{(12)}{<} & \sum_{\substack{s \in R \\ N_s(Z) \supseteq N_s(X)}} \left(c_s(X) + \sum_{i \in N_s(X)} t_{i,s}\right) + c_r(X) + \sum_{i \in N_r(X)} t_{i,r} = C(X). \end{aligned}$$

This implies that X is not an optimal strategy profile, a contradiction. □

6.2 Convex Cost Functions

When cost functions are convex, there is a universally optimal protocol with a price of anarchy equal to one. We only require that the cost functions are non-negative, non-decreasing and the per-unit costs $\frac{c(\ell(Z))}{\ell(Z)}$ are non-decreasing with respect to the load $\ell(Z)$. Such functions are quite rich and contain non-negative, non-decreasing and convex functions.

We introduce the *opt-enforcing protocol* for which we prove a price of anarchy of 1. The intuition behind this protocol is similar to the enforcing protocols presented before: make all undesired outcomes unstable by charging some player a very high price.

Definition 6.2 (opt-enforcing protocol). *For a given model (N, R, Σ, d, c, t) the* opt-enforcing protocol *takes as input an optimal outcome Y. We again denote for any outcome z and resource r the set of foreign players on r by $N_r^1(Z) = \{i \in N_r(Z) \backslash N_r(Y)\}$. Then, the opt-enforcing protocol assigns the cost sharing methods*

$$\xi_{i,r}(Z) := \begin{cases} d_i \cdot \dfrac{c_r(Z)}{\ell_r(Z)}, \text{ if } r \in Z_i, N_r^1(Z) = \emptyset, & \text{(14a)} \\ c_r(Z), \text{ if } r \in Z_i, N_r^1(Z) \neq \emptyset \text{ and } i = \min N_r^1(Z), & \text{(14b)} \\ 0, \text{ else.} & \text{(14c)} \end{cases}$$

Under the opt-enforcing protocol, the players share the cost proportional to their demands on all resources without foreign players. On resources with foreign players, the foreign player with the smallest index pays the entire cost of the resource.

Theorem 6.3. *The opt-enforcing protocol is separable and has a price of anarchy of* 1.

Proof. Budget balance and separability are clear from the definition of the protocol. For stability it can easily be verified that for an instance (N, R, Σ, d, c, t) the optimal outcome Y is a Nash equilibrium. We only prove the bound on the price of anarchy, showing that all Nash equilibria X have the same cost as a socially optimal profile Y, i.e. for every player i

$$\sum_{r \in X_i} (\xi_{i,r}(X) + t_{i,r}) \leq \sum_{r \in Y_i} (\xi_{i,r}(Y) + t_{i,r}). \tag{15}$$

By definition of Nash equilibria,

$$\sum_{r \in X_i} (\xi_{i,r}(X) + t_{i,r}) \leq \sum_{r \in Y_i} (\xi_{i,r}(Y_i, X_{-i}) + t_{i,r}). \tag{16}$$

Two cases are to be considered for every resource $r \in Y_i$: either it hosts a nonempty set $N_r^1(X)$ of foreign players or $N_r^1(X) = \emptyset$. If there are foreign players on r, then one of them will pay for the entire cost there and hence (14c) gives $\xi_{i,r}(Y_i, X_{-i}) = 0$ (note that by Definition 6.2 $i \notin N_r^1(Y_i, X_{-i})$). If there are no foreign players on r, then $\ell_r(Y_i, X_{-i}) \leq \ell_r(Y)$ yields

$$\frac{c_r(Y_i, X_{-i})}{\ell_r(Y_i, X_{-i})} \leq \frac{c_r(Y)}{\ell_r(Y)},$$

because the cost per unit is non-decreasing. Plugging this into (14a) we have $\xi_{i,r}(Y_i, X_{-i}) \leq \xi_{i,r}(Y)$. With $\xi_{i,r}(Y_i, X_{-i}) \leq \xi_{i,r}(Y)$ in both cases for all resources $r \in Y_i$,

$$\sum_{r \in Y_i} (\xi_i(Y_i, X_{-i}) + t_{i,r}) \leq \sum_{r \in Y_i} (\xi_{i,r}(Y) + t_{i,r}),$$

which combined with (16) yields (15). □

7 Computational Complexity of Cost Shares

Our protocols introduced in the previous sections involve decharged profiles which in turn rely on the computation of optimal profiles which are generally NP-hard to compute. A natural question is whether there are different approaches that allow to compute optimal cost shares or cost shares with good approximation guarantees in polynomial time (see also the open question in [44][Section 5.1]). We will answer this question negatively by showing that the problem of computing optimal cost shares is strongly NP-hard and not approximable by any constant factor even for instances with unweighted players, zero delays and singleton strategies. In light of this hardness we restrict our model to unweighted players, zero delays and singleton strategies, and switch our objective: previously we looked for cost shares such that the best/worst Nash equilibrium is a good approximation of the social optimum, now we ask for *polynomial time computable* cost shares that approximate the optimal cost shares in terms of cost of the best/worst Nash equilibrium.

7.1 Hardness and Inapproximability

Let I be an instance of a matroid facility location game and $\Xi(I)$ the set of possible cost shares for I. We investigate the computational complexity of the following two optimization problems

$$\min_{\xi\in\Xi(I)} \min_{X\in P(I,\xi)} \frac{C(X)}{C(Y)} \qquad \text{(Best Nash)}$$

$$\min_{\xi\in\Xi(I)} \max_{X\in P(I,\xi)} \frac{C(X)}{C(Y)}, \qquad \text{(Worst Nash)}$$

where $P(I,\xi)$ denotes the set of Nash equilibria of the game induced by I and ξ, Y is some optimal strategy profile.

Theorem 7.1. *Problem* Best Nash *is strongly NP-complete and there are no $c \log n$ approximation algorithms for any $c < 1$, unless $P = NP$. This holds even for instances with unweighted players, zero delays, singleton strategies and unit fixed costs.*

Proof. By Corollary 3.5, problem Best Nash can be reformulated as

$$\min_{X\in\Sigma} C(X) \text{ s.t. } X \text{ is decharged.}$$

Now, given any $X \in \Sigma$, we can check in polynomial time whether X is feasible (i.e., it fulfills (D1) and (D2)) and whether $C(X) = k$ for some given k, thus, Best Nash is in NP.

We prove inapproximability by providing an approximation preserving reduction from Hitting Set. An instance of Hitting Set consists of a collection C of subsets of a finite set of elements E. A hitting set for C is a subset $S \subseteq E$ such that S contains at least one element from each subset in C. The goal is to minimize the cardinality of the hitting set.

Given an instance of Hitting Set, we create an instance of Best Nash by identifying C with Σ, E with R and assuming fixed unit costs on the resources and no delays.

Claim 7.2. *There is a hitting set of cardinality less or equal to k if and only if there is a decharged profile with cost less or equal k.*

Proof. In the constructed cost sharing instance, every strategy profile is decharged: with no delays (D1) is always fulfilled, and with fixed unit resource costs (D2), too. Given a strategy profile, the used resources form a hitting set and the cost of the profile is equal to the cardinality of the set. Given a hitting set, a strategy profile can be constructed by assigning each player i to one of the resources in Σ_i that is in the hitting set. □

The proof follows now from the above claim and the fact that Hitting Set is equivalent to the Set Cover problem [5] which is known not to be approximable within $c \log n$ for any $c < 1$ unless $P = NP$ [41]. □

We now prove an inapproximability result for computing cost shares minimizing the cost of the worst Nash equilibrium

Theorem 7.3. *Problem* Worst Nash *is strongly NP-hard and there are no $c \log n$ approximation algorithms for any $c < 1$ unless $P = NP$. This holds even for instances with unweighted players, zero delays, singleton strategies and unit fixed costs.*

Proof. We again reduce from Hitting Set. Given an instance of Hitting Set, we construct a cost sharing game like above and add two more players denoted player a and player b. We set $\Sigma_a = \Sigma_b = R$.

Claim 7.4. *There is a hitting set of cardinality less or equal to k if and only if*

$$\min_{\xi \in \Xi(I)} \max_{X \in P(I,\xi)} C(X) \leq k.$$

Proof. For the direction $\Leftarrow$, observe that if there are cost shares ξ such that the most expensive PNE X has $C(X) \leq k$, then the resources used in X form a hitting set of cardinality less than or equal to k. For the $\Rightarrow$ direction, given a hitting set of size at most k, we construct an assignment of players to resources that costs at most k and in which both a and b share a resource with other players. We call this profile X. We assign the following cost shares for all $r \in R$ and $Z \in \Sigma$:

$$\xi_{a,r}(Z) = \begin{cases} 1, & \text{if } N_r(Z) = \{a\} \text{ or } N_r(Z) = \{a,b\}, \\ \frac{1}{2}, & \text{if } a \in N_r(Z) \text{ and } N_r(Z) \subseteq N_r(X) \cup \{a,b\} \text{ and } N_r(Z) \not\subseteq \{a,b\}, \\ 0, & \text{if } a \in N_r(Z) \text{ and } N_r(Z) \not\subseteq N_r(X) \cup \{a,b\} \text{ and } b \notin N_r(Z), \\ 1, & \text{if } a \in N_r(Z) \text{ and } N_r(Z) \not\subseteq N_r(X) \cup \{a,b\} \text{ and } b \in N_r(Z), \\ 0, & \text{if } a \notin N_r(Z), \end{cases}$$

$$\xi_{b,r}(Z) = \begin{cases} 1, & \text{if } b \in N_r(Z) \text{ and } a \notin N_r(Z), \\ 0, & \text{else}, \end{cases}$$

and for all $i \neq a, b$

$$\xi_{i,r}(Z) = \begin{cases} 1 - \xi_{a,r}(Z) - \xi_{b,r}(Z), & \text{if } i = \min N_r(Z) \setminus N_r(X), \\ 1 - \xi_{a,r}(Z) - \xi_{b,r}(Z), & \text{if } N_r(Z) \setminus N_r(X) = \emptyset \text{ and } i = \min N_r(Z), \\ 0, & \text{else}. \end{cases}$$

The profile X is a PNE under these cost shares, as player a pays $\frac{1}{2}$ and would pay either $\frac{1}{2}$ or 1 on any other resource, player b pays 0 and all other players pay 1 or less and would pay 1 if they switched.

For any profile $Z \neq X$ with $C(Z) > C(X)$, there is some resource r with $N_r(X) = \emptyset$ and $N_r(Z) \neq \emptyset$. We show that such a profile cannot be an equilibrium. For an equilibrium, the definition of the cost shares for player b implies that $Z_b = Z_a$, as otherwise player b could immediately reduce her cost by switching to Z_a. We, hence, assume from now on $Z_b = Z_a$. If $N_r(Z) = \{a, b\}$, then player a has cost 1 and, if $N_{X_a}(Z) \neq \emptyset$ could reduce her cost to $\frac{1}{2}$ by switching to X_a or, in case $N_{X_a}(Z) = \emptyset$, could reduce her cost to 0 by switching to some other resource s with $N_s(Z) \not\subseteq N_s(X)$. If $N_r(Z) \neq \{a, b\}$, then there are other players than a and b using r as $N_r(Z) \neq \emptyset$ and $Z_b = Z_a$. Hence, either $\{a, b\} \cap N_r(Z) = \emptyset$ and player a could reduce her cost by switching to r, or $\{a, b\} \subset N_r(Z)$ and player a could reduce her cost by switching somewhere else as above – unless $N_s(Z) = \emptyset$ for all $s \neq r$, in which case $C(Z) = 1$, a contradiction to $C(Z) > C(X)$. Consequently, X with $C(X) \leq k$ is the most expensive PNE of the game. □

□

7.2 Approximation Algorithms

We have shown that it is computationally hard to find cost shares that approximate the cost of the optimal cost shares within a logarithmic factor in n. This hardness result even holds for instances with unweighted players, zero delays, singleton strategies and unit fixed costs. In light of this hardness even for this restricted class of problems, we study approximation algorithms for the case of unweighted players, zero delays and singleton strategies, still assuming general nondecreasing costs. Such instances are given as $I = (N, R, \Sigma, c)$, with $d_i = 1$ and $t_{i,r} = 0$ for all $i \in N$ and $r \in R$. We present in the following an approximation algorithm that computes in polynomial time a decharged profile whose cost is bounded from above by $\mathcal{H}_n$ times the cost of an optimal profile - matching the performance bound presented in Section 4.

For a given instance $I = (N, R, \Sigma, c)$, the algorithm starts with an empty strategy profile X and updates X as it iteratively assigns the n players to the resources in R. While the algorithm runs, let $\bar{n}(X)$ be the number of players not yet assigned to a resource and let $G(X)$ be a directed graph representing the current allocation X. The graph has vertices for all players, all resources and additional source and sink vertices s and t. $G(X)$ is bipartite with arcs according to the following rules:

- arc (i, r) from player i to resource r if $r \in \Sigma_i$
- arc (r, i) from resource r to player i if $r = X_i$, i.e., i is assigned to r in X
- arc (s, i) if player i is not assigned to any resource in X.

All above arcs have cost 0 and capacity 1.

Theorem 7.5. *Algorithm 2 computes in polynomial time cost shares that guarantee a price of stability of at most $\mathcal{H}_n$ and a price of anarchy of at most n.*

Proof. We prove the theorem in four lemmas.

Algorithm 2 Compute Semi-Proportional Cost Shares

```
Input: instance $I = (N, R, \Sigma, c)$
Output: cost shares $\xi$
start with empty profile $X$, no players assigned to resources
while not all players $N$ assigned to resources $R$ do
    compute $G(X)$
    for all combinations $r \in R$ and $1 \leq v \leq \bar{n}(X)$ do
        compute, if existent, an integer $s, r$ flow $f$ with flow value $v(f) = v$.
    end for
    from all combinations of $r$ and $v(f)$ for which a flow was found, choose the one with the
    lowest resulting cost-per-unit $\frac{1}{v(f)}(c_r(|N_r(X)| + v(f)) - c_r(X))$. For each $(i, r)$ arc used in
    this flow, update $X$ such that player $i$ is assigned to resource $r$.
end while
assign cost shares for all $i \in N$ and $Z \in \Sigma$
    $\xi_i(Z) := \begin{cases} \frac{c_{Z_i}(Z)}{|N_{Z_i}(Z)|} & \text{, if } N_{Z_i}(Z) \subseteq N_{Z_i}(X) \\ \frac{c_{Z_i}(Z)}{|N_{Z_i}(Z) \setminus N_{Z_i}(X)|} & \text{, if } N_{Z_i}(Z) \not\subseteq N_{Z_i}(X) \text{ and } i \in N_{Z_i}(Z) \setminus N_{Z_i}(X) \\ 0 & \text{, otherwise.} \end{cases}$
Return $\xi$
```

Lemma 7.6. *The profile X used in the definition of the cost shares is a PNE.*

Proof. We use a counter k to enumerate the iterations of the algorithm's main while loop. Denote by X^k the profile at the beginning of iteration k and denote by f^k the selected flow. We observe that on each resource r, the sequence of the per-unit costs of the load increments is nondecreasing, that is, if in iteration k the load on r is increased and the next load increase on r is in iteration l, then

$$\frac{c_r(X^{k+1}) - c_r(X^k)}{|N_r(X^{k+1})| - |N_r(X^k)|} \leq \frac{c_r(X^{l+1}) - c_r(X^l)}{|N_r(X^{l+1})| - |N_r(X^l)|}$$

because otherwise Algorithm 2 had increased the load on r in iteration k directly to $N_r(X^{l+1})$ instead of $N_r(X^{k+1})$. Thus, if on resource r the last load increment was in iteration k^*, then the per unit cost of r in X, which is the average over all such increments, is no greater then the per unit cost of the increment in k^*. This in turn is no greater than the cost of adding one player to any other resource $\bar{r}$ in X^{k^*} because otherwise the algorithm had increased the load on $\bar{r}$ instead of increasing it on r. Hence, for all $i \in N_r(X)$ and $\bar{r} \in \Sigma_i$,

$$\begin{aligned} \xi_i(X) = \frac{c_r(X)}{|N_r(X)|} &\leq \frac{c_r(X) - c_r(X^{k^*})}{|N_r(X)| - |N_r(X^{k^*})|} \leq \frac{c_{\bar{r}}(|N_{\bar{r}}(X^{k^*})| + 1) - c_{\bar{r}}(|N_{\bar{r}}(X^{k^*})|)}{1} \\ &\leq c_{\bar{r}}(|N_{\bar{r}}(X^{k^*})| + 1) \leq c_{\bar{r}}(|N_{\bar{r}}(X)| + 1) = \xi_i(\bar{r}, X_{-i}). \end{aligned}$$

□

Lemma 7.7. *Let Y be some optimal strategy profile. Then, the profile X used to define the cost shares has cost $C(X) \leq \mathcal{H}_n \cdot C(Y)$.*

Proof. We first check that updating X^k corresponding to the flow f^k as done in line 7 works as desired. Clearly, updating X^k this way is feasible with regard to the strategy space Σ. Moreover, when we add $v(f^k)$ players going from X^k to X^{k+1}, the load only changes on one resource r and the cost difference of X^k and X^{k+1} is $C(X^{k+1}) - C(X^k) = c_r(X^{k+1}) - c_r(X^k)$.

To bound the per-unit cost increase in a given iteration k, note that for each resource $\bar{r}$ with less users than in the optimal profile Y, $|N_{\bar{r}}(X^k)| < |N_{\bar{r}}(Y)|$, there is an integer $s, \bar{r}$ flow with flow value $|N_{\bar{r}}(Y)| - |N_{\bar{r}}(X^k)|$. As the algorithm does not choose this flow, the corresponding per-unit cost is greater than that on the resource r,

$$\frac{c_r(X^{k+1}) - c_r(X^k)}{v(f^k)} \leq \frac{c_{\bar{r}}(Y) - c_{\bar{r}}(X^k)}{N_{\bar{r}}(X^k)| - |N_{\bar{r}}(Y)|}. \tag{17}$$

Noting that the number of players missing on such resources is in total at least $\bar{n}(k)$, i.e.,

$$\bar{n}(k) \leq \sum_{\substack{\bar{r} \in R \\ |N_{\bar{r}}(X^k)| < |N_{\bar{r}}(Y)|}} (|N_{\bar{r}}(X^k)| - |N_{\bar{r}}(Y)|),$$

we sum up (17) over all such resources,

$$\frac{c_r(X^{k+1}) - c_r(X^k)}{v(f^k)} \leq \sum_{\substack{\bar{r} \in R \\ |N_{\bar{r}}(X^k)| < |N_{\bar{r}}(Y)|}} \frac{c_{\bar{r}}(Y) - c_{\bar{r}}(X^k)}{\bar{n}(k)} \leq \frac{C(Y)}{\bar{n}(k)}. \tag{18}$$

Then,

$$\begin{aligned}
C(X) &= \sum_k \left(C(X^{k+1}) - C(X^k) \right) \\
&\leq \sum_k \frac{v(f^k)}{\bar{n}(X^k)} \cdot C(Y) = \sum_k \frac{v(f^k)}{\sum_{j \geq k} v(f^j)} \cdot C(Y) = \sum_k \sum_{i=1}^{v(f^k)} \frac{1}{\sum_{j \geq k} v(f^j)} \cdot C(Y) \\
&\leq \sum_k \sum_{i=1}^{v(f^k)} \frac{1}{\sum_{j \geq k} v(f^j) - i + 1} \cdot C(Y) = \mathcal{H}_n \cdot C(Y).
\end{aligned}$$

□

Lemma 7.8. *Let Y be some optimal strategy profile and $Z \in P(I, \xi)$ a PNE. Then, $C(Z) \leq n \cdot C(Y)$.*

Proof. First observe that from (18) we can derive $c_r(X^{l+1}) - c_r(X^l) \leq C(Y)$ for every iteration l where the load on a resource r is increased. Particularly, $c_r(X) \leq \sum_{i \in N_r(X)} C(Y)$. We estimate the cost of Z by estimating the cost shares of groups of players in Z.

For players that are in X on resources r with $N_r(X) = N_r(Z)$, by the above observation $\sum_{i \in N_r(X)} \xi_i(Z) = c_r(X) \leq \sum_{i \in N_r(X)} C(Y)$.

For players that are in X on resources r with $N_r(Z) \subset N_r(X)$, we choose the smallest k such that $|N_r(Z)| + 1 \leq |N_r(X^k)|$ and let q be the number of times the algorithm has increased the load on r up to iteration k. Then, $q \leq |N_r(Z)| + 1$ and by our above observation the q load increases will increase the cost by up to $q \cdot C(Y)$. Then, for all $i \in N_r(X)$,

$$\xi_i(Z) \leq \xi_i(r, Z_{-i}) = \frac{c_r(|N_r(Z)| + 1)}{|N_r(Z)| + 1} \leq \frac{c_r(X^k)}{q} \leq \frac{q \cdot C(Y)}{q} = C(Y),$$

The first inequality holds because Z is a Nash equilibrium, the second inequality follows from our previous observations $|N_r(Z)| + 1 \leq |N_r(X^k)|$ and $q \leq |N_r(Z)| + 1$. For the last inequality observe that $c_r(X^k)$ is the sum of the costs of the q load increases up to iteration k.

For players that are in X on resources r with $N_r(Z) \not\subseteq N_r(X)$, we have $\xi_i(Z) \leq \xi_i(X_i, Z_{-i}) = 0$ for all players $i \in N_r(X)$ because Z is a PNE. Summing up across all player gives the desired

$$C(Z) = \sum_{i \in N} \xi_i(Z) \leq \sum_{i \in N} C(Y) = n \cdot C(Y).$$

□

Lemma 7.9. *The runtime of Algorithm 2 is polynomial in the size of the input instance.*

Proof. For an instance I with n players and m resources, the algorithm's main while loop (lines 2 to 8) can run at most n times. Computing $G(X)$ can be done in $O(mn)$ time, the $O(mn)$ flows of an iteration can each be computed in $O(m^3 n^3)$ with the Edmonds-Karp algorithm [18], updating X can be done in $O(n)$ time. Hence, the algorithm's runtime is bounded by $O(m^4 n^5)$. □

□

8 Conclusions

We considered in this paper facility location games where facilities have nondecreasing load-dependent costs and players experience player-specific delays when connecting to an open facility. We designed several cost sharing protocols for this setting and proved that they induce the smallest possible price of anarchy and price of stability, respectively. The following problems remain open and deserve further research. We assumed that the player's strategy space is described by the set of bases of a player-specific matroid. While matroids contain many interesting classes (such as the facility location games), other classes such as general multi-commodity networks are not covered. The problem of designing optimal cost sharing protocols for general strategy spaces still eludes us. Our results regarding the computational complexity of cost sharing protocols show that optimal cost shares cannot be approximated by logarithmic factor in the number of players. While we devised the best possible approximation algorithm for singleton strategies and delays in $\{0, \infty\}$, the case of general matroids with arbitrary delays remains unresolved.

Acknowledgments. We thank two anonymous referees for their extraordinary effort in providing helpful and detailed comments for this paper.

References

[1] H. Ackermann, H. Röglin, and B. Vöcking. Pure Nash equilibria in player-specific and weighted congestion games. *Theoret. Comput. Sci.*, 410(17):1552–1563, 2009.

[2] S. Albers. On the value of coordination in network design. *SIAM J. Comput.*, 38(6):2273–2302, 2009.

[3] E. Anshelevich, A. Dasgupta, J. Kleinberg, É. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. *SIAM J. Comput.*, 38(4):1602–1623, 2008.

[4] A. Archer, J. Feigenbaum, A. Krishnamurthy, R. Sami, and S. Shenker. Approximation and collusion in multicast cost sharing. *Games Econom. Behav.*, 47(1):36–71, 2004.

[5] G. Ausiello, A. D'Atri, and M. Protasi. Structure preserving reductions among convex optimization problems. *Journal of Computer and System Sciences*, 21(1):136 – 153, 1980.

[6] D. Bertsekas and R. Gallagher. *Data Networks – 2nd Edition*. Prentice-Hall, 1992.

[7] V. Bilò, A. Fanelli, M. Flammini, G. Melideo, and L. Moscardelli. Designing fast converging cost sharing methods for multicast transmissions. *Theory Comput. Syst.*, 47(2):507–530, 2010.

[8] A. Bogomolnaia, R. Holzman, and H. Moulin. Sharing the cost of a capacity network. *Math. Oper. Res.*, 35(1):173–192, 2010.

[9] I. Caragiannis. Efficient coordination mechanisms for unrelated machine scheduling. In C. Mathieu, editor, *Proc. 20th Annual ACM-SIAM Sympos. on Discrete Algorithms*, pages 815–824. SIAM, 2009.

[10] C. Chekuri, J. Chuzhoy, L. Lewin-Eytan, J. Naor, and A. Orda. Non-cooperative multicast and facility location games. In J. Feigenbaum, J. C.-I. Chuang, and D. M. Pennock, editors, *Proc. 7th ACM Conf. Electronic Commerce*, pages 72–81. ACM, 2006.

[11] H.-L. Chen and T. Roughgarden. Network design with weighted players. *Theory Comput. Syst.*, 45(2):302–324, 2009.

[12] H.-L. Chen, T. Roughgarden, and G. Valiant. Designing network protocols for good equilibria. *SIAM J. Comput.*, 39(5):1799–1832, 2010.

[13] Y.-J. Chen and J. Zhang. Design of price mechanisms for network resource allocation via price of anarchy. *Math. Program.*, 131(1-2):333–364, 2012.

[14] G. Christodoulou, E. Koutsoupias, and A. Nanavati. Coordination mechanisms. In J. Díaz, J. Karhumäki, A. Lepistö, and D. Sannella, editors, *Proc. 31st Internat. Colloquium on Automata, Languages and Programming*, volume 3142 of *LNCS*, pages 45–56. Springer, 2004.

[15] R. Cole, J. R. Correa, V. Gkatzelis, V. S. Mirrokni, and N. Olver. Inner product spaces for minsum coordination mechanisms. In L. Fortnow and S. P. Vadhan, editors, *Proc. 43rd Annual ACM Sympos. Theory Comput.*, pages 539–548. ACM Press, 2011.

[16] J. Correa, A. Schulz, and N. Stier-Moses. Selfish routing in capacitated networks. *Math. Oper. Res.*, 29(4):961–976, 2004.

[17] M. Desrochers, P. Marcotte, and M. Stan. The congested facility location problem. *Location Science*, 3(1):9–23, 1995.

[18] J. Edmonds and R. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19(2):248–264, 1972.

[19] A. Epstein, M. Feldman, and Y. Mansour. Strong equilibrium in cost sharing connection games. *Games Econom. Behav.*, 67(1):51–68, September 2009.

[20] E. Even-Dar, A. Kesselman, and Y. Mansour. Convergence time to Nash equilibrium in load balancing. *ACM Trans. Algorithms*, 3(3):1–21, 2007.

[21] D. Fotakis, S. Kontogiannis, E. Koutsoupias, M. Mavronicolas, and P. Spirakis. The structure and complexity of Nash equilibria for a selfish routing game. In P. Widmayer, F. Triguero, R. Morales, M. Hennessy, S. Eidenbenz, and R. Conejo, editors, *Proc. 29th Internat. Colloquium on Automata, Languages and Programming*, volume 2380 of *LNCS*, pages 123 – 134, 2002.

[22] M. Gairing, B. Monien, and K. Tiemann. Routing (un-) splittable flow in games with player-specific linear latency functions. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *Proc. 33rd Internat. Colloquium on Automata, Languages and Programming*, volume 4051 of *LNCS*, pages 501–512, 2006.

[23] T. Harks and M. Klimm. On the existence of pure Nash equilibria in weighted congestion games. *Math. Oper. Res.*, 37(3):419–436, 2012.

[24] T. Harks and K. Miller. The worst-case efficiency of cost sharing methods in resource allocation games. *Oper. Res.*, 59(6):1491–1503, 2011.

[25] M. Hoefer. Competitive cost sharing with economies of scale. *Algorithmica*, 60:743–765, 2011.

[26] S. Ieong, R. McGrew, E. Nudelman, Y. Shoham, and Q. Sun. Fast and compact: A simple class of congestion games. In M. M. Veloso and S. Kambhampati, editors, *Proc. 20th National Conf. Artificial Intelligence and the 17th Innovative Appl. Artificial Intelligence Conf.*, pages 489–494. AAAI Press / The MIT Press, 2005.

[27] N. Immorlica, L. Li, V. Mirrokni, and A. Schulz. Coordination mechanisms for selfish scheduling. *Theor. Comput. Sci.*, 410(17):1589–1598, 2009.

[28] R. Johari and J. Tsitsiklis. A scalable network resource allocation mechanism with bounded efficiency loss. *IEEE J. Sel. Area Comm.*, 24(5):992–999, 2006.

[29] R. Johari and J. N. Tsitsiklis. Efficiency of scalar-parameterized mechanisms. *Oper. Res.*, 57(4):823–839, 2009.

[30] K. Kollias and T. Roughgarden. Restoring pure equilibria to weighted congestion games. In L. Aceto, M. Henzinger, and J. Sgall, editors, *Proc. 38th Internat. Colloquium on Automata, Languages and Programming*, number 6756 in LNCS, pages 539–551, 2011.

[31] J. Könemann, S. Leonardi, G. Schäfer, and S. van Zwam. A group-strategyproof cost sharing mechanism for the steiner forest game. *SIAM J. Comput.*, 37(5):1319–1341, 2008.

[32] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In C. Meinel and S. Tison, editors, *Proc. 16th Internat. Sympos. Theoretical Aspects of Comput. Sci.*, number 1563 in LNCS, pages 404–413, 1999.

[33] I. Milchtaich. Congestion games with player-specific payoff functions. *Games Econom. Behav.*, 13(1):111 – 124, 1996.

[34] H. Moulin. Axiomatic cost and surplus sharing. In K. J. Arrow, A. K. Sen, and K. Suzumura, editors, *Handbook of Social Choice and Welfare*, volume 1, chapter 06, pages 289–357. Elsevier, 1 edition, 2002.

[35] H. Moulin. The price of anarchy of serial, average and incremental cost sharing. *Econom. Theory*, 36(3):379–405, 2008.

[36] H. Moulin. An efficient and almost budget balanced cost sharing method. *Games and Economic Behavior*, 70(1):107–131, September 2010.

[37] H. Moulin and S. Shenker. Strategyproof sharing of submodular costs: budget balance versus efficiency. *Economic Theory*, 18(3):511–533, 2001.

[38] A. Orda, R. Rom, and N. Shimkin. Competitive routing in multi-user communication networks. *IEEE Transactions on Networking*, 1:510–521, 1993.

[39] M. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

[40] M. Pál and É. Tardos. Group strategyproof mechanisms via primal-dual algorithms. In *FOCS*, pages 584–593, 2003.

[41] R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. pages 475–484, New York, NY, 1997. ACM.

[42] O. Rozenfeld and M. Tennenholtz. Strong and correlated strong equilibria in monotone congestion games. In M. Mavronicolas and S. Kontogiannis, editors, *Proc. 2nd Internat. Workshop on Internet and Network Econom.*, volume 4286 of *LNCS*, pages 74 – 86, 2006.

[43] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, 2003.

[44] P. von Falkenhausen and T. Harks. Optimal cost sharing for resource selection games. *Math. Oper. Res.*, 38(1):184–208, February 2013.

[45] F. Yao, A. Demers, and S. Shenker. A scheduling model for reduced cpu energy. In *Proc. 35th Annual IEEE Sympos. Foundations Comput. Sci.*, pages 374–382, Washington, DC, USA, 1995. IEEE Computer Society.

A Appendix: Proof of Lemma 5.2

Lemma A.1. *The Price of Anarchy for facility location games with basic or separable protocols is at most n.*

Proof. Let (N, R, Σ, d, c, t) be a facility location model. Let $\xi_{i,r}$ for $i \in N, r \in R$ be the cost share functions assigned by the Enforcing Protocol and let X be the decharged profile returned by Algorithm 1 for the protocol with intermediate profiles $X^1, X^2, \dots, X$. We use notation for players and jobs interchangeably, denoting jobs by the letter q, for example $q \in N_r(X)$ for the jobs on a resource. For each job q, define $y(q)$ and $p(q)$ as in the analysis of the algorithm and denote additionally by $x(q)$ the resource job q is on in profile X and by X^q the algorithm's intermediate profile in which q is first on $x(q)$. From the analysis of the algorithm, we know

$$c_{x(q)}(X^q) + t_{q,x(q)} \leq \frac{1}{p(q)} c_{y(q)}(Y) + t_{q,y(q)} \tag{19}$$

for all jobs q that were moved by the algorithm. For jobs q that were not moved by the algorithm, we set $X^q := Y$.

To prove the lemma, we show $C(Z) \leq n \cdot C(Y)$ for any pure Nash equilibrium Z. To this end, we fix such a profile Z and link it to the profiles (X_i, Z_{-i}) via the Nash property,

$$\begin{aligned} C(Z) &= \sum_{i \in N} \left(\xi_i(Z) + \sum_{r \in Z_i} t_{r,i} \right) \leq \sum_{i \in N} \left(\xi_i(X_i, Z_{-i}) + \sum_{r \in X_i} t_{r,i} \right) = \sum_{i \in N} \left(\sum_{\substack{r \in X_i \\ N_r^1(Z) = \emptyset}} \xi_{i,r}(X_i, Z_{-i}) + \sum_{r \in X_i} t_{i,r} \right) \\ &= \sum_{\substack{r \in R \\ N_r^1(Z) = \emptyset}} \sum_{i \in N_r(X)} (\xi_{i,r}(X_i, Z_{-i}) + t_{i,r}) + \sum_{\substack{r \in R \\ N_r^1(Z) \neq \emptyset}} \sum_{i \in N_r(X)} t_{i,r} \\ &\leq \sum_{\substack{r \in R \\ N_r^1(Z) = \emptyset}} \left(\sum_{q \in N_r(X) \cap N_r(Y)} \left(c_r(Y) + t_{q,r} \right) + 2 \cdot \sum_{q \in N_r(X) \setminus N_r(Y)} \left(\frac{1}{p(q)} c_{y(q)}(Y) + t_{q,y(q)} \right) \right) + \sum_{\substack{r \in R \\ N_r^1(Z) \neq \emptyset}} \sum_{q \in N_r(X)} t_{q,r} \end{aligned} \tag{20}$$

Proving (20) is a major challenge of this proof and beforehand we give a brief intuition for this inequality: for jobs that are moved by the algorithm we have an at most logarithmic cost-increase going from profile Y to profile Z, represented by the second term, while for jobs not moved by the algorithm, the cost-increase can even be linear as represented by the first term. In our worst-case example in Lemma 5.1, this linear cost-increase dominates the logarithmic cost-increase: no jobs are moved by the algorithm.

To prove (20), we partition the resources without foreign players into two sets,

- $R_1 := \{r \in R : N_r^1(Z) = \emptyset \text{ and } |N_r(X) \backslash N_r(Z)| \leq 1\}$ - resources where at most one job is missing,
- $R_2 := \{r \in R : N_r^1(Z) = \emptyset \text{ and } |N_r(X) \backslash N_r(Z)| > 1\}$ - resources, where multiple jobs are missing.

For the resources in both sets, we find bounds corresponding to (20) in two separate claims. Afterwards we will combine the two claims to prove the lemma.

Claim A.2. *For $r \in R_1$,*

$$\sum_{i \in N_r(X)} \xi_{i,r}(X_i, Z_{-i}) + t_{i,r} \leq \sum_{q \in N_r(X) \cap N_r(Y)} \Big(c_r(Y) + t_{q,r}\Big) + 2 \cdot \sum_{q \in N_r(X) \backslash N_r(Y)} \left(\frac{1}{p(q)} c_{y(q)}(Y) + t_{q,y(q)}\right).$$

Proof. Recall that for all $r \in R$

$$c_r(X) \leq \begin{cases} c_r(Y) & \text{if } N_r(X) \setminus N_r(Y) = \emptyset, \qquad (21) \\ c_r(X^{q_r}) \leq \dfrac{1}{p(q_r)} c_{y(q_r)}(Y) + t_{q_r,y(q_r)} - t_{q_r,r} & \text{if } N_r(X) \setminus N_r(Y) \neq \emptyset, \qquad (22) \end{cases}$$

where job q_r denotes the last job moved to r by the algorithm. The second inequality (22) follows from (19). To prove the claim, we have for $r \in R_1$,

$$\begin{aligned} \sum_{i \in N_r(X)} \xi_{i,r}(X_i, Z_{-i}) + t_{i,r} &= \mathbf{1}_{N_r(Z) \neq \emptyset} \sum_{i \in N_r(Z)} \xi_{i,r}(Z) + \mathbf{1}_{N_r(X) \backslash N_r(Z) = \{i^*\}} \xi_{i^*,r}(X_{i^*}, Z_{-i^*}) + \sum_{i \in N_r(X)} t_{i,r} & (23) \\ &\leq \mathbf{1}_{N_r(Z) \neq \emptyset} c_r(X) + \mathbf{1}_{N_r(X) \backslash N_r(Z) = \{i^*\}} c_r(X) + \sum_{i \in N_r(X)} t_{i,r} & (24) \\ &\leq \sum_{q \in N_r(X) \cap N_r(Y)} c_r(Y) + 2 \cdot \sum_{q \in N_r(X) \backslash N_r(Y)} \left(\frac{1}{p(q)} c_{y(q)}(Y) + t_{q,y(q)} - t_{q,r}\right) + \sum_{q \in N_r(X)} t_{q,r} & (25) \\ &\leq \sum_{q \in N_r(X) \cap N_r(Y)} \Big(c_r(Y) + t_{q,r}\Big) + 2 \cdot \sum_{q \in N_r(X) \backslash N_r(Y)} \left(\frac{1}{p(q)} c_{y(q)}(Y) + t_{q,y(q)}\right), \end{aligned}$$

where $\mathbf{1}$ denotes the indicator function tied to the condition in subscript and, if applicable, i^* is the single player using resource r in X but not in Z. Equation (23) is due to the nature of R_1 and for (24) we use that there are no foreign players on machines $r \in R_1$. In inequality (25), we estimate using both (21) and (22). For the case where both indicator functions are true, we multiply the term from (22) by 2. For the term from (21) this is not necessary because, if both indicator functions are true and $N_r(X) \backslash N_r(Y) = \emptyset$, then there are multiple jobs $q \in N_r(X) \cap N_r(Y)$ and, hence, $\sum_{q \in N_r(X) \cap N_r(Y)} c_r(Y) \geq 2 \cdot c_r(Y)$. □

Claim A.3. *For $r \in R_2$,*

$$\sum_{i \in N_r(X)} \xi_{i,r}(X_i, Z_{-i}) + t_{i,r} \leq \sum_{q \in N_r(X) \cap N_r(Y)} c_r(Y) + t_{i,r} + \sum_{q \in N_r(X) \backslash N_r(Y)} \frac{1}{p(q)} c_{y(q)}(Y) + t_{q,y(q)}.$$

Proof. We denote the jobs on $r \in R_2$ in profile X by $q^r_1, \ldots, q^r_{|N_r(X)|}$ such that they are indexed with non-increasing weights $d_{q^r_1} \geq \cdots \geq d_{q^r_{|N_r(X)|}}$. Let $s(r) := \min\{i : q^r_i \in N_r(Z)\}$. Since the jobs are indexed in the same order as their players, the protocol assigns for $i \leq |N_r(X)|$

$$\xi_{q^r_i, r}(X_{q^r_i}, Z_{-q^r_i}) = \begin{cases} c_r(\ell_r(Z) + d_{q^r_i}) & \text{if } i < s(r), \\ c_r(\ell_r(Z)) & \text{if } i = s(r), \\ 0 & \text{if } i > s(r). \end{cases} \tag{26}$$

We now define an automorphism $\sigma_r : \{q^r_1, \ldots, q^r_{|N_r(X)|}\} \to \{q^r_1, \ldots, q^r_{|N_r(X)|}\}$ that maps the first $t(r) := |N_r(X) \backslash N_r(Y)|$ jobs (by index) to $N_r(X) \backslash N_r(Y)$, such that

- $\sigma_r(q^r_1)$ is the last job that was moved to r by the algorithm,
- $\sigma_r(q^r_2)$ is the second-last job that was moved to r by the algorithm,
- ...
- $\sigma_r(q^r_{t(r)})$ is the first job that was moved to r by the algorithm.

The remaining jobs are mapped arbitrarily to $N_r(X) \cap N_r(Y)$, keeping σ_r bijective. Then,

$$\ell_r(Z) \leq \ell_r(X) - \sum_{j=1}^{s(r)-1} d_{q^r_j} \tag{27a}$$

$$\leq \ell_r(X) - \sum_{j=1}^{s(r)-1} d_{\sigma_r(q^r_j)} \tag{27b}$$

$$\leq \ell_r(X^{\sigma_r(q^r_{s(r)})}), \tag{27c}$$

where (27a) holds because $N_r(Z) \subset N_r(X)$ and $q^r_1, \ldots, q^r_{s(r)-1} \notin N_r(Z)$ by definition of $s(r)$. Inequality (27b) holds because we indexed the jobs from big to small and hence the first $s(r) - 1$ jobs are the 'biggest' jobs on resource r. For (27c), if $s(r) \leq t(r)$, that is, if $\sigma_r(q^r_{s(r)})$ was moved to resource r, then in profile $X^{\sigma_r(q^r_{s(r)})}$ none of the jobs $\sigma_r(q^r_{s(r)-1}), \ldots, \sigma_r(q^r_1)$ moved to r after job $\sigma_r(q^r_{s(r)})$ are on resource r, and consequently (27c) follows. Otherwise, if $s(r) > t(r)$, that is, if $\sigma_r(q^r_{s(r)}) \in N_r(Y)$, then $Y = X^{\sigma_r(q^r_{s(r)})}$ and in this profile none of the jobs $\sigma_r(q^r_{t(r)}), \ldots, \sigma_r(q^r_1)$ that were moved to resource r are on resource r and hence (27c) follows. We find likewise for $i < s(r)$,

$$\ell_r(Z) + d_{q^r_i} \leq \ell_r(X) - \sum_{j=1}^{i-1} d_{q^r_j} \leq \ell_r(X) - \sum_{j=1}^{i-1} d_{\sigma_r(q^r_j)} \leq \ell_r(X^{\sigma_r(q^r_i)}), \tag{28}$$

where the above inequalities hold for similar reasons as (27). We complete the proof of the claim by

$$
\begin{align}
\sum_{i\in N_r(X)} \xi_{i,r}(X_i, Z_{-i}) + t_{i,r} &= c_r(\ell_r(Z)) + \sum_{i=1}^{s(r)-1} c_r(\ell_r(Z) + d_{q_i^r}) + \sum_{i\in N_r(X)} t_{i,r} \tag{29}\\
&\le \sum_{i=1}^{s(r)} c_r(\ell_r(X^{\sigma_r(q_i^r)})) + \sum_{i\in N_r(X)} t_{i,r} \le \sum_{q\in N_r(X)} c_r(X^{\sigma_r(q)}) + t_{i,r} \tag{30}\\
&= \sum_{q\in N_r(X)} c_r(X^q) + t_{i,r} \tag{31}\\
&\le \sum_{q\in N_r(X)\cap N_r(Y)} c_r(Y) + t_{i,r} + \sum_{q\in N_r(X)\setminus N_r(Y)} \frac{1}{p(q)} c_{y(q)}(Y) + t_{q,y(q)}, \tag{32}
\end{align}
$$

where equation (29) follows from (26) and inequality (30) follows from (27) and (28). Equation (31) holds because σ_r is an automorphism on $S_r(X)$ and finally inequality (32) follows from our definition of the intermediate profiles X^q and our results regarding these profiles as in (19). □

We now continue the proof of Lemma 5.2 where we left off with (20) and conclude across both sets,

$$
\begin{align}
C(Z) &\le \sum_{\substack{r\in R\\ N_r^1(Z)=\emptyset}} \sum_{i\in N_r(X)} (\xi_{i,r}(X_i, Z_{-i}) + t_{i,r}) + \sum_{\substack{r\in R\\ N_r^1(Z)\neq\emptyset}} \sum_{i\in N_r(X)} t_{i,r} \notag\\
&\le \sum_{\substack{r\in R\\ N_r^1(Z)=\emptyset}} \left(\sum_{q\in N_r(X)\cap N_r(Y)} \left(c_r(Y) + t_{q,r}\right) + 2\cdot \sum_{q\in N_r(X)\setminus N_r(Y)} \left(\frac{1}{p(q)} c_{y(q)}(Y) + t_{q,y(q)}\right)\right) + \sum_{\substack{r\in R\\ N_r^1(Z)\neq\emptyset}} \sum_{q\in N_r(X)} t_{q,r} \tag{33}\\
&\le \sum_{r\in R} \left(\sum_{q\in N_r(X)\cap N_r(Y)} \left(c_r(Y) + t_{q,r}\right) + \sum_{q\in N_r(Y)\setminus N_r(X)} \left(\frac{2}{p(q)} c_r(Y) + 2\cdot t_{q,r}\right)\right) \tag{34}\\
&\le \sum_{r\in R} \left(|N_r(Y)\cap N_r(X)|\cdot c_r(Y) + \sum_{p=|N_r(Y)\cap N_r(X)|+1}^{|N_r(Y)|} \frac{2}{p}\cdot c_r(Y) + 2\cdot \sum_{q\in N_r(Y)} t_{q,r}\right) \tag{35}\\
&\le \sum_{r\in R} \left(|N_r(Y)|\cdot c_r(Y) + 2\cdot \sum_{q\in N_r(Y)} t_{q,r}\right) \le n\cdot C(Y). \notag
\end{align}
$$

Here, (33) follows from Claims A.2 and A.3. In (34), we change the order of summation: instead summing up the $q \in N_r(X) \setminus N_r(Y)$ that were moved *from* other resources by the algorithm, we sum up the $q \in N_r(Y) \setminus N_r(X)$ that were moved *to* other resources by the algorithm. At the same time we extend summation across all resources $r \in R$ where before we only summed up across resources with $N_r^1(Z) = \emptyset$. For (35), recall how we introduced $p(q)$: the first job that is moved away has $p(q) = |N_r(Y)|$, the next has $p(q) = |N_r(Y) - 1|$ until the last job that is moved away has $p(q) = |N_r(Y) \cap N_r(X)| + 1$. □