Optimal Cost Sharing Protocols for Scheduling Games

Philipp von Falkenhausen
Berlin Institute of Technology
Institute of Mathematics
p.falkenhausen@gmail.com

Tobias Harks
Berlin Institute of Technology
Institute of Mathematics
harks@math.tu-berlin.de

ABSTRACT

We consider the problem of designing cost sharing protocols to minimize the price of anarchy and stability for a class of scheduling games. Here, we are given a set of players, each associated with a job of certain non-negative weight. Any job fits on any machine, and the cost of a machine is a non-decreasing function of the total load on the machine. We assume that the private cost of a player is determined by a cost sharing protocol. We consider four natural design restrictions for feasible protocols: stability, budget balance, separability, and uniformity. While budget balance is self-explanatory, the stability requirement asks for the existence of pure-strategy Nash equilibria. Separability requires that the resulting cost shares only depend on the set of players on a machine. Uniformity additionally requires that the cost shares on a machine are instance-independent, that is, they remain the same even if new machines are added to or removed from the instance. We call a cost sharing protocol basic, if it satisfies only stability and budget balance. Separable and uniform cost sharing protocols additionally satisfy separability and uniformity, respectively. For n-player games we show that among all basic and separable cost sharing protocols, there is an optimal protocol with price of anarchy and stability of precisely $\mathcal{H}_n = \sum_{i=1}^{n} 1/i$. For uniform protocols we present a strong lower bound showing that the price of anarchy is unbounded. Moreover, we obtain several results for special cases in which either the cost functions are restricted, or the job sizes are restricted. As a byproduct of our analysis, we obtain a complete characterization of outcomes that can be enforced as a pure-strategy Nash equilibrium by basic and separable cost sharing protocols.

Categories and Subject Descriptors

J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Scheduling*

General Terms

Theory

Keywords

Cost sharing, scheduling, congestion games

1. INTRODUCTION

Congestion games play a fundamental role for many applications, including traffic networks, telecommunication networks and economics. In a congestion game, there is a set of resources and a pure strategy of a player consists of a subset of resources. The cost of a resource depends only on the number of players choosing the resource, and the private cost of a player is the sum of the costs of the chosen resources. Under these assumptions, Rosenthal proved the existence of a pure Nash equilibrium (PNE for short) [20]. An important question in congestion games is the degree of suboptimality caused by selfish resource allocation. Koutsoupias and Papadimitriou [17] introduced a measure to quantify the inefficiency of Nash equilibria which they termed the *price of anarchy.* The price of anarchy is defined as the worst-case ratio of the cost of a Nash equilibrium over the cost of a system optimum. In the past decade, considerable progress has been made in exactly quantifying the price of anarchy for many interesting classes of games. In the context of nonatomic network routing games, the price of anarchy for specific classes of cost functions is well understood, see Roughgarden and Tardos [23], Roughgarden [21] and Correa, Schulz, and Stier-Moses [11]. (For an overview of these results, we refer to the book by Roughgarden [22].) Awerbuch et al. [4], Christodoulou and Koutsoupias [9], Aland et al. [2] and Bhawalkar et al. [5] derived several tight bounds on the price of anarchy for weighted and unweighted congestion games with specific classes of latency functions. Despite these bounds, it is known that the price of anarchy for general latency functions is unbounded even on simple *parallel-arc* networks [5, 23].

Motivated by the fact that pure Nash equilibria may be very inefficient even in parallel-arc networks, we focus in this paper on the design of *cost sharing methods* as a means to leverage the resulting price of anarchy. The concrete scenario that we consider is the problem of scheduling jobs on parallel machines. We are given a set of players, each associated with a job of certain non-negative weight. Any job fits on any machine, and the cost of a machine is a non-decreasing function of the total load on the machine. We assume that the private cost of a player is determined by a cost sharing method. For instance, a simple cost sharing method that has been analyzed in the aforementioned literature is *average cost sharing*, see [5, 11, 23]. In almost all settings in the theory and practice of mechanism or protocol design, a designer may only choose protocols out of a set of *feasible* protocols. Therefore, we have to precisely define the design space of feasible protocols. To this end, we define the following four properties listed below which are defined more formally in Section 2. These properties have been introduced first by Chen et al. [8] in the context of the design of cost sharing protocols for network design games (a formal definition of these games will be given in Section 1.2).

1. *Stability*: There is at least one pure strategy Nash equilibrium in each scheduling game induced by the cost sharing protocol.

2. *Budget-balance*: For every outcome of a scheduling game induced by the cost sharing protocol, the cost of each resource is exactly covered by the collected cost shares of the players using the resource.

3. *Separability*: In each scheduling game induced by the cost sharing protocol, the cost shares of each resource are completely determined by the set of players that use it.

4. *Uniformity*: Across all scheduling games induced by the cost sharing protocol, the cost shares of a resource (for each potential set of users) depend only on the resource cost, and not on the set of available resources.

A cost sharing method is called *basic* if it satisfies (1)-(2), *separable* if it satisfies (1)-(3), and *uniform* if it satisfies (1)-(4). We briefly discuss the above four properties and refer to [8] for a more detailed treatment. The condition (2) is the least controversial in the context of cost sharing protocols. The stability condition (1) requires the existence of at least one Nash equilibrium in pure strategies. While this requirement restricts the search space for cost sharing protocols, it is certainly the solution concept of choice when mixed or correlated strategies have no meaningful physical interpretation in the game played; see also the discussion in Osborne and Rubinstein [19, § 3.2] about critics of mixed Nash equilibria. While condition (3) seems restrictive, it is crucial for practical applications in which cost sharing methods have only local information about their own resource usage (see for instance the TCP/IP protocol design, where routers drop packets based on some function of the number of packets in the queue, see [24]). Uniformity (4) is the strongest and perhaps the most problematic design restriction. A uniform protocol is not only separable but also *strongly local* in the sense that the cost shares of a resource are independent of the set of resources available to the game designer. This property may be crucial for systems in which the resources can be added or removed over time and a reconfiguration of the system (changing the cost sharing protocol) is too costly.

The goal of this paper is twofold. On the one hand side, we want to systematically analyze the achievable worst-case efficiency of Nash equilibria by basic, separable and uniform cost sharing protocols in the context of scheduling games. Besides this worst-case perspective, we also ask a larger question: Which outcomes can actually be enforced as pure Nash equilibria? More precisely, we call an outcome of a scheduling game *weakly-enforceable* if there is a basic protocol that induces the outcome as a pure Nash equilibrium. We call an outcome of a scheduling game *strongly-enforceable* if there is a basic protocol that induces the outcome as the most expensive pure Nash equilibrium.

1.1 Our Results

We study protocol design problems in the context of scheduling games, where the goal is to minimize the induced price of anarchy and the price of stability. Our results for these problems can be summarized as follows.

Among all basic and separable protocols, we provide an *optimal* protocol minimizing the resulting price of anarchy and price of stability simultaneously. For n-player scheduling games, the optimal value of the price of anarchy and stability is precisely the n-th harmonic number $\mathcal{H}_n = \sum_{i=1}^{n} 1/i$. Moreover, we obtain a complete characterization of weakly-enforceable outcomes. This characterization can be used for designing cost sharing protocols minimizing the price of stability with respect to an arbitrary objective function. We also derive sufficient conditions for an outcome to be strongly-enforceable. Our proof of this result is constructive by providing a cost sharing protocol that strongly enforces an outcome satisfying the sufficient conditions. We then show that this protocol gives rise to an optimal cost sharing protocol minimizing the price of anarchy and stability as mentioned above. For scheduling games with cost functions that have non-decreasing per-unit costs, we derive an optimal cost sharing protocol with price of anarchy equal to 1. We remark that this assumption is quite weak insofar as non-decreasing and convex cost functions satisfy non-decreasing per-unit costs.

We also study the achievable price of anarchy of uniform cost sharing protocols. We show that there is no uniform cost sharing protocol with a bounded price of anarchy. This bound even holds for a family of instances with only 3 players, at most 3 machines and cost functions with non-decreasing costs per unit. Only for instances in which the demands are integer multiples of each other, we present a cost sharing protocol with a bounded price of anarchy of n.

1.2 Related Work

There is a large body of work on scheduling games (or singleton congestion games) with unweighted and weighted players [1, 12, 13, 14, 15, 18]. Most of these papers study the existence and price of anarchy of pure Nash equilibria for the uniform cost sharing protocol in which the private cost of every player is equal to the cost on the resource. These works, however, do not consider the design perspective of cost sharing protocols. Christodoulou et al. [10] and follow-up papers such as [6, 16] study *coordination mechanisms* and their price of anarchy in scheduling games in which n players assign a task to one of m machines. Rather than paying a share of the resulting cost of a machine as in our scenario, the players in these games consider the completion time of their respective job as private cost. This completion time depends on the sequence in which the jobs on a machine are processed which in turn is given by the *coordination mechanism.* The notion of private cost in these papers establishes an entirely different set of Nash equilibria compared to our work and hence their results concerning the price of anarchy are unrelated to ours.

Our work is motivated by the paper by Chen et al. [8].

In this paper, the authors study the design of cost sharing protocols for network design games, see also Anshelevich et al. [3] and Chen and Roughgarden [7] for earlier work on network design games. In a network design game, each player i has a unit demand that she wishes to send along a path in a (directed or undirected) network connecting her source node s_i to her terminal node t_i. Every edge has a constant non-negative cost and the problem is to design a separable or uniform cost sharing protocol so as to minimize the price of anarchy and stability in this setting. Our approach follows their lead in terms of the feasible protocol space, but we apply cost sharing protocols to the structural different class of scheduling models. On the one hand, such scheduling models are more general in the sense that we allow arbitrary non-decreasing cost functions instead of constant costs on the resources. Moreover, in contrast to [8], we allow players to have different non-negative weights. On the other hand, scheduling models are more restricted in the sense that we consider a relatively simple strategy space for the players, that is, a pure strategy for a player is simply a single resource. Moreover, in contrast to [8], our games are symmetric, that is, every player has access to every resource. These structural differences result in different approaches and also the results of [8] are different to ours. For example, while Chen et al. [8] proved bounds on the price of anarchy for uniform protocols of order $\Theta(\log(n)), \Theta(\text{polylog}(n))$, and n for undirected single-sink instances, undirected multi-commodity instances, and directed single-sink instances, respectively, we show that for scheduling games such results are impossible. The price of anarchy for uniform protocols inducing scheduling games is unbounded. Finally, it is worth noting that, while [8] analyzed separable and uniform protocols, we additionally analyze the larger class of basic protocols.

2. MODEL AND PROBLEM STATEMENT

A *scheduling model* is represented by a tuple (N, M, d, c). Here, $N = \{1, \dots, n\}$ is a nonempty set of players and $M = \{a_1, \dots, a_m\}$ is a nonempty set of machines. Every player is associated with a task of weight d_i and $d = (d_1, \dots, d_n)$ is the combined weight vector. Every machine $a \in M$ has an associated non-negative and non-decreasing cost function $c_a : \mathbb{R}^+ \to \mathbb{R}^+$. We assume $c_a(0) = 0$ for all $a \in M$. The vector of cost functions is denoted by $c = (c_1, \dots, c_m)$. Given a scheduling model (N, M, d, c), we associate a strategic game represented by the tuple (N, X, ξ). Here, it is assumed that every task fits on every machine, thus, the set of pure strategies for player $i \in N$ is $X_i = M$ and the overall strategy space is $X = M^n$. The outcomes $x = (x_1, \dots, x_n) \in M^n$ are vectors of machines where the strategy played by player i is machine x_i. The private cost of player $i \in N$ in such an outcome x is determined by the *cost sharing method* $\xi_i : X \to \mathbb{R}^+$. A *cost sharing protocol* $\Xi : (N, M, d, c) \mapsto \xi$ provides every scheduling model with a vector $\xi = (\xi_i)_{i \in N}$ of such cost sharing methods. For $x \in M^n$, the set of players using some machine $a \in M$ is denoted by $S_a(x) := \{i \in N : x_i = a\}$ and for $a \in M$, the load on machine a is defined as $\ell_a(x) := \sum_{i \in S_a(x)} d_i$. The cost of an outcome is defined as $C(x) := \sum_{a \in M} c_a(\ell_a(x))$. Abusing notation, we will often write $c_a(x)$ instead of $c_a(\ell_a(x))$. We consider cost minimization games, thus, when choosing her strategy, each player strives to minimize her resulting private cost $\xi_i(x)$. We say that the game (N, X, ξ) on a scheduling model (N, M, d, c) is induced by the protocol Ξ.

An important solution concept in non-cooperative game theory for the analysis of strategic games are pure Nash equilibria. Using standard notation in game theory, for an outcome $x \in M^n$ we denote by

$$(a, x_{-i}) := (x_1, \dots, x_{i-1}, a, x_{i+1}, \dots, x_n) \in M^n$$

the outcome that arises if only player i deviates to strategy a.

Definition 2.1. (Pure Nash Equilibrium) Let (N, X, ξ) be a scheduling game. The outcome x is a pure Nash equilibrium if no player i can strictly reduce her private cost by unilaterally moving to a different machine, that is, for all $i \in N$

$$\xi_i(x) \le \xi_i(a, x_{-i}) \quad \text{for all } a \in M. \tag{2.1}$$

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 outcome. The price of stability measures the smallest ratio of the cost of a Nash equilibrium and the cost of an optimal outcome. 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. [8] in the context of designing cost sharing methods for network design games.

Definition 2.2. (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 scheduling models (N, M, d, c) with cost sharing methods $(\xi_i)_{i \in N}$ such that

$$c_a(x) = \sum_{i \in S_a(x)} \xi_i(x) \text{ for all } a \in M, x \in M^n. \tag{2.2}$$

   This property requires $c_a(0) = 0$ for unused machines, which we will assume in the paper.

3. *separable* if it is basic and if it induces only games (N, M^n, ξ) for which in any two outcomes $x, x' \in M^n$

$$S_a(x) = S_a(x') \Rightarrow \xi_i(x) = \xi_i(x') \; \forall i \in S_a(x), a \in M.$$

4. *uniform* if it is separable and if it assigns any two models (N, M, d, c), (N, M', d, c') with cost sharing methods $(\xi_i)_{i \in N}$ and $(\xi'_i)_{i \in N}$ such that the following condition holds. For all $a \in M \cap M'$ with $c_a = c'_a$ and all outcomes $x \in M^n$, $x' \in M'^n$

$$S_a(x) = S_a(x') \Rightarrow \xi_i(x) = \xi'_i(x') \text{ for all } i \in S_a(x).$$

Informally, separability means that in an outcome x the value $\xi_i(x)$ depends only on the set $S_{x_i}(x)$ of players sharing machine x_i and disregards all other information contained in

x. Still, separable protocols can assign cost sharing methods that are specifically tailored to the given scheduling model, for example based on an optimal outcome. Uniform protocols are not allowed to do this, they even disregard the layout of the model and assign the same cost sharing methods when machines are added to or removed from the model.

We denote by $\mathcal{B}_n, \mathcal{S}_n$ and $\mathcal{U}_n$ the set of basic, separable and uniform cost sharing protocols for scheduling games with n players, respectively. We obtain the following optimization problems that we address in this paper.

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

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

3. BASIC AND SEPARABLE PROTOCOLS

We start with studying basic and separable cost sharing protocols. While our goal is to find a cost sharing protocol minimizing the induced PoA and PoS, we first study the issue of enforceability of pure Nash equilibria by basic and separable cost sharing protocols. To be more precise, given a scheduling model (N, M, d, c), we first ask which outcomes $x \in M^n$ can be enforced as pure Nash equilibria by some basic or separable cost sharing protocol. We will differentiate between *weakly* enforceable outcomes and *strongly* enforceable outcomes, see the definition below.

Definition 3.1. (Enforceable outcomes) Consider a scheduling model (N, M, d, c) and an outcome $x \in M^n$.

i) x is *weakly-enforceable* if there exists a basic cost sharing protocol Ξ such that x is a Nash equilibrium in the game (N, M^n, ξ) induced by Ξ.

ii) x is *separable weakly-enforceable* if there exists a separable cost sharing protocol Ξ such that x is a Nash equilibrium in the game (N, M^n, ξ) induced by Ξ.

iii) x is *strongly-enforceable* if there exists a separable cost sharing protocol Ξ such that x is the most expensive Nash equilibrium in the game (N, M^n, ξ) induced by Ξ, i.e. $C(x') \leq C(x)$ for all Nash equilibria $x' \in M^n$.

In the following section, we will give an exact characterization of weakly-enforceable and separable weakly-enforceable outcomes. This characterization provides a structural property that can be used to design cost sharing protocols for minimizing the price of stability for *arbitrary* objective functions.

Throughout this section, the players are assumed to be ordered by non-decreasing weights: $d_1 \leq d_2 \leq \cdots \leq d_n$.

3.1 Weakly-Enforceable Outcomes

This section provides an exact characterization of weakly-enforceable outcomes. Our characterization relies on the notion of *decharged outcomes* defined below.

Definition 3.2. (Weakly decharged outcome) Consider a scheduling model (N, M, d, c). A machine $a \in M$ is *weakly decharged* in an outcome $x \in M^n$ if

$$c_a(x) \leq \sum_{i \in S_a(x)} \min_{b \in M} c_b(b, x_{-i}). \tag{3.1}$$

The outcome x itself is called *weakly decharged* if all machines are weakly decharged.

We further introduce the weak x-enforcing protocol.

Definition 3.3. (Weak x-enforcing protocol) The *weak x-enforcing protocol* takes as input a weakly decharged outcome x. We use x to define for any outcome z and machine a the sets $S_a^0(z) := \{i \in S_a(z) \cap S_a(x)\}$ (*home players on a*) and $S_a^1(z) := \{i \in S_a(z) \backslash S_a(x)\}$ (*foreign players on a*). Then, the weak x-enforcing protocol assigns for all $i \in N, z \in M^n$ the following cost sharing methods

$$\xi_i(z) := \begin{cases} \dfrac{\min_{b \in M} c_b(b, x_{-i})}{\sum_{j \in S_{x_i}(x)} \min_{b \in M} c_b(b, x_{-j})} \cdot c_{x_i}(x), \\ \qquad \text{if } S_{z_i}(z) = S_{z_i}(x) \text{ and } c_{x_i}(x) > 0, \\ c_{z_i}(z), \text{ if } S^1_{z_i}(z) \neq \emptyset \text{ and } i = \min S^1_{z_i}(z), \\ c_{z_i}(z), \text{ if } S^1_{z_i}(z) = \emptyset,\ S_{z_i}(z) \subset S_{z_i}(x) \\ \qquad \text{and } i = \min S_{z_i}(z), \\ 0, \text{ else.} \end{cases}$$

Informally, if $S_a(z) = S_a(x)$, the players on machine a share the cost proportional to their opportunity cost (cost of change) in outcome x. Otherwise, the smallest foreign player (deviating from outcome x) or, if there are none, the smallest home player (not deviating) pays the entire cost of the machine. Observe that in weakly decharged outcomes x we have $\sum_{j \in S_a(x)} \min_{b \in M} c_b(b, x_{-j}) > 0$ for all $a \in M$ with $c_a(x) > 0$ and thus the protocol is well defined. We are now ready to state our first main result.

THEOREM 3.4. *For any scheduling model (N, M, d, c) and outcome x, the following statements are equivalent.*

(i) the outcome x is weakly decharged,

(ii) the outcome x is weakly-enforceable,

(iii) the outcome x is separable weakly-enforceable.

Observe that (iii) $\Rightarrow$ (ii) holds because by definition separable protocols are a subclass of basic protocols. We prove (i) $\Rightarrow$ (iii) and (ii) $\Rightarrow$ (i) by two lemmas.

LEMMA 3.5. *For every weakly decharged outcome x, the weak x-enforcing protocol is a separable cost sharing protocol and weakly enforces x.*

PROOF. Budget balance and separability of the cost sharing methods are clear from the definition of the protocol, thus, we prove only that x is a Nash equilibrium. For all machines $a \in M$ with $c_a(x) > 0$ we are in the first case of the definition of the protocol, thus, we obtain

$$\begin{aligned} \xi_i(x) &= \frac{\min_{b \in M} c_b(b, x_{-i})}{\sum_{j \in S_a(x)} \min_{b \in M} c_b(b, x_{-j})} \cdot c_a(x) \\ &\leq \min_{b \in M} c_b(b, x_{-i}) \leq \min_{b \in M \backslash \{a\}} \xi_i(b, x_{-i}) \text{ for all } i \in S_a(x), \end{aligned}$$

where the first inequality holds because outcome x is weakly decharged. For all other machines $a \in M$, we have $\xi_i(x) = c_a(x) = 0$ for all $i \in S_a(x)$ and thus x is a pure Nash equilibrium. $\square$

LEMMA 3.6. *Consider the scheduling model (N, M, d, c). Then, any weakly-enforceable outcome x is weakly decharged.*

PROOF. Say x is a Nash equilibrium under the basic protocol Ξ that assigns cost sharing methods ξ. Then $\xi_i(x) \leq \min_{b \in M} \xi_i(b, x_{-i})$ for all $i \in N$ and hence due to budget balance of Ξ,

$$c_a(x) = \sum_{i \in S_a(x)} \xi_i(x) \leq \sum_{i \in S_a(x)} \min_{b \in M} \xi_i(b, x_{-i}) \leq \sum_{i \in S_a(x)} \min_{b \in M} c_b(b, x_{-i})$$

for all machines $a \in M$. Thus, x is weakly decharged. □

The above characterization has a direct consequence for the design of cost sharing protocols so as to minimize the price of stability with respect to an arbitrary objective function over the strategy space. As formalized below, by Theorem 3.4 this problem reduces to solving a well-structured finite-dimensional optimization problem.

COROLLARY 3.7. *Let (N, M, d, c) be a scheduling model and let $F : M^n :\to \mathbb{R}$ be a social welfare function. Then, $\min_{\xi \in \mathcal{B}_n} PoS(\xi; F)$ and $\min_{\xi \in \mathcal{S}_n} PoS(\xi; F)$ can be reduced to solving the optimization problem*

$$\min_{x \in M^n} F(x) \;\; s.t. \;\; c_a(x) \leq \sum_{i \in S_a(x)} \min_{b \in M} c_b(b, x_{-i}) \, \forall \, a \in M.$$

3.2 Strongly-Enforceable Outcomes

In this section, we turn to strongly-enforceable outcomes. We present a slightly extended protocol that we term the the strong x-enforcing protocol. We will show that outcomes that are strongly decharged (a definition will follow shortly) are strongly-enforceable by this protocol.

Definition 3.8. (Strongly Decharged Outcome) Consider a scheduling model (N, M, d, c). A machine $a \in M$ is *strongly decharged* if it is weakly decharged and additionally

$$c_a(x) < \sum_{i \in S_a(x)} \min_{b \in M} c_b(b, x_{-i}), \text{ if } |S_a(x)| > 1 \text{ and } c_a(x) > 0. \tag{3.2}$$

Machines that are not strongly decharged are called *charged*. The outcome x is called *strongly decharged* if all machines are strongly decharged.

We now introduce the strong x-enforcing protocol.

Definition 3.9. (Strong x-enforcing protocol) The *strong x-enforcing protocol* takes as input a strongly decharged outcome x. As before, we use x to define for any outcome z and machine a the sets $S_a^0(z)$ and $S_a^1(z)$. Additionally, we define the set $S_a^2(z) := \{i \in S_a(z) \backslash S_a(x) : c_{x_i}(x) = 0\}$ that we term *strong foreign players on a*. Then, the strong x-enforcing protocol assigns for all $i \in N, z \in M^n$, the following cost sharing methods:

$$\xi_i(z) := \begin{cases} \dfrac{\min_{b \in M} c_b(b, x_{-i})}{\sum_{j \in S_{x_i}(x)} \min_{b \in M} c_b(b, x_{-j})} \cdot c_{x_i}(x), \\ \qquad \text{if } S_{z_i}(z) = S_{z_i}(x) \text{ and } c_{x_i}(x) > 0, \\ c_{z_i}(z), \text{ if } S_{z_i}^2(z) \neq \emptyset \text{ and } i = \min S_{z_i}^2(z), \\ c_{z_i}(z), \text{ if } S_{z_i}^2(z) = \emptyset, \, S_{z_i}^1(z) \neq \emptyset, \, i = \min S_{z_i}^1(z), \\ c_{z_i}(z), \text{ if } S_{z_i}^1(z) = \emptyset, \, S_{z_i}(z) \subset S_{z_i}(x), i = \min S_{z_i}(z) \\ 0, \qquad \text{else.} \end{cases}$$

The protocol works almost the same as the weak x-enforcing protocol, it only accounts differently for strong foreign players.

THEOREM 3.10. *If an outcome x is strongly decharged, then the strong x-enforcing protocol is separable and strongly enforces x.*

PROOF. Separability follows from the definition of the strong x-enforcing protocol. We only show that for any Nash equilibrium $z \neq x$ we have $C(z) \leq C(x)$. To this end, fix such a z and let

$$i := \min \{j \in N : z_j \neq x_j\} \tag{3.3}$$

be the smallest player who deviates from x. First, note that for all $j > i$,

$$\xi_j(z) \leq \begin{cases} \xi_j(z_i, z_{-j}) = 0 \text{ , if } c_{x_j}(x) > 0 \\ \xi_j(x_j, z_{-j}) = 0 \text{ , if } c_{x_j}(x) = 0, \end{cases} \tag{3.4}$$

because z is a Nash equilibrium. Hence,

$$c_a(z) = 0 \text{ for all } a \neq z_i \text{ with foreign players } S_a^1(z) \neq \emptyset. \tag{3.5}$$

Also, $c_a(z) \leq c_a(x)$ for all machines $a \neq z_i$ that only have home players $S_a^0(z) = S_a(z)$, because for these machines $\ell_a(z) \leq \ell_a(x)$. Thus, we already have

$$c_a(z) \leq c_a(x) \quad \text{for all machines } a \neq z_i. \tag{3.6}$$

If there is a strong foreign player on z_i, then even $c_{z_i}(z) = 0$ and we are done. Thus, from now on we assume that there are no strong foreign players on z_i. We can bound $c_{z_i}(z)$ from above using the Nash inequality $c_{z_i}(z) = \xi_i(z) \leq \xi_i(x_i, z_{-i})$. The remaining proof focuses on bounding the value $\xi_i(x_i, z_{-i})$ from above.

The value of $\xi_i(x_i, z_{-i})$ assigned by the x-enforcing protocol depends on $S_{x_i}(x_i, z_{-i})$ and $c_{x_i}(x)$, for which there are three possibilities, according to the definition of the strong x-enforcing protocol.These cases are

1. $S_{x_i}(x_i, z_{-i}) = S_{x_i}(x)$ and $c_{x_i}(x) > 0$, where the protocol returns $\xi_i(x_i, z_{-i}) = \xi_i(x)$.
2. $S_{x_i}(x_i, z_{-i}) \subset S_{x_i}(x)$ and $i = \min S_{x_i}(x_i, z_{-i})$, where the protocol returns $\xi_i(x_i, z_{-i}) = c_{x_i}(x_i, z_{-i})$.
3. All cases in which the protocol returns $\xi_i(x_i, z_{-i}) = 0$.

In each case we will find $c_{x_i}(z) + c_{z_i}(z) \leq c_{x_i}(x) + c_{z_i}(x)$ and thus with (3.6) we have $C(z) \leq C(x)$, which proves the Theorem. Note that (3.6) already implies $c_{x_i}(z) \leq c_{x_i}(x)$.

We begin with Case 1. The condition $c_{x_i}(x) > 0$ implies that if there is some strong foreign player $j > i$ (with $z_j \neq x_j$ and $c_{x_j}(x) = 0$), then $c_{z_i}(z) = \xi_i(z) \leq \xi_i(z_j, z_{-i}) = 0$ and we are done. Thus, we will in the following assume that there are no strong foreign players at all. If $\xi_i(x) = 0$, we obtain $0 = \xi_i(x) = \xi_i(x_i, z_{-i}) \geq \xi_i(z) = c_{z_i}(z)$, because we are in Case 1. Thus, we will also assume

$$\xi_i(x) > 0. \tag{3.7}$$

We now compare the allocation of load in the outcomes z and x, respectively. First, we consider machines $a \neq z_i$, which host foreign players $j \in S_a(z) \setminus S_a(x)$. For these foreign players we obtain

$$\min_{b \in M} c_b(b, x_{-j}) \geq \min_{b \in M} c_b(b, x_{-i}) \quad (3.8a)$$
$$\geq \xi_i(x) \quad (3.8b)$$
$$> 0. \quad (3.8c)$$

Observe that (3.3) implies $j > i$ and hence (weights are ordered non-decreasingly) $d_j \geq d_i$. As the cost functions are non-decreasing, the first inequality (3.8a) follows. Inequality (3.8b) holds since x is decharged. The last inequality (3.8c) follows from (3.7). We conclude for machine a

$$c_a(a, x_{-j}) \geq \xi_j(a, x_{-j}) \geq \xi_j(x) \quad (3.9)$$
$$= \frac{c_{x_j}(x)}{\sum_{k \in S_{x_j}(x)} \min_{b \in M} c_b(b, x_{-k})} \cdot \min_{b \in M} c_b(b, x_{-j}) \quad (3.10)$$
$$> 0 = c_a(z), \quad (3.11)$$

where (3.9) holds because x is a Nash equilibrium and (3.10) stems from the definition of the protocol because there are no strong foreign players and hence $c_{x_j}(x) > 0$. The inequality (3.11) holds because of (3.8) and the equality holds because of (3.5). Hence, there must be a non-empty set of players $S_a(x) \setminus S_a(z)$. These players cannot be strong foreign players, thus $c_a(x) > 0$. With $c_a(z) = 0$ and $c_a(x) > 0$ we have $\ell_a(x) > \ell_a(z)$ for all machines $a \neq z_i$ with foreign players. For all machines a without foreign players we know $\ell_a(x) \geq \ell_a(z)$ and for machine x_i even $\ell_{x_i}(x) = \ell_{x_i}(z) + d_i$ because we are in Case 1. Since the total load is the same in x and z, we have for machine z_i

$$\ell_{z_i}(z_i, x_{-i}) = \ell_{z_i}(x) + d_i \leq \ell_{z_i}(z). \quad (3.12)$$

Consequently,

$$\xi_i(z) = c_{z_i}(z) \geq c_{z_i}(z_i, x_{-i}) \quad (3.13)$$
$$\geq \frac{c_{x_i}(x)}{\sum_{j \in S_{x_i}(x)} \min_{b \in M} c_b(b, x_{-j})} \cdot \min_{b \in M} c_b(b, x_{-i}) \quad (3.14)$$
$$= \xi_i(x) = \xi_i(x_i, z_{-i}), \quad (3.15)$$

where the first inequality (3.13) holds because of (3.12) and the second inequality (3.14) because x is decharged and $c_{z_i}(z_i, x_{-i}) \geq \min_{b \in M} c_b(b, x_{-i})$. Equality (3.15) holds by the definition of the strong x-enforcing protocol for Case 1 and the last equation holds because we assume Case 1. If $|S_{x_i}(x)| > 1$, then inequality (3.14) is strict, because x is strongly decharged (i.e., (3.2) holds) which implies $\xi_i(z) > \xi_i(x_i, z_{-i})$. This contradicts the fact that z is a Nash equilibrium. Thus, $S_{x_i}(x) = \{i\}$ and $c_{z_i}(z) = \xi_i(z) \leq \xi_i(x_i, z_{-i}) = c_{x_i}(x_i, z_{-i}) = c_{x_i}(x)$. Moreover, using $c_{x_i}(z) = 0$, because $\ell_{x_i}(z) = 0$, we obtain $c_{x_i}(z) + c_{z_i}(z) \leq c_{x_i}(x) + c_{z_i}(x)$ as desired.

Case 2 is $S_{x_i}(x_i, z_{-i}) \subset S_{x_i}(x)$ and $i = \min S_{x_i}(x_i, z_{-i})$. Here, we obtain

$$c_{z_i}(z) = \xi_i(z) \leq \xi_i(x_i, z_{-i}) = c_{x_i}(x_i, z_{-i}) \leq c_{x_i}(x),$$

where the first inequality holds because z is a Nash equilibrium. The second inequality holds because Case 2 implies $\ell_{x_i}(x_i, z_{-j}) \leq \ell_{x_i}(x)$. We also get

$$c_{x_i}(z) = \sum_{j \in S_{x_i}(z)} \xi_j(z) \leq \sum_{j \in S_{x_i}(z)} \xi_j(z_i, z_{-j}) = 0.$$

This inequality is a result of (3.4), because in this case all players $j \in S_{x_i}(z)$ have a higher index $j > i$. Consequently, we have again $c_{x_i}(z) + c_{z_i}(z) \leq c_{x_i}(x) + c_{z_i}(x)$.

Finally, we examine Case 3 where the protocol returns $\xi_i(x_i, z_{-i}) = 0$ and thus for the Nash equilibrium z we have $c_{z_i}(z) = \xi_i(z) \leq \xi_i(x_i, z_{-i}) = 0$. Again, $c_{x_i}(z) + c_{z_i}(z) \leq c_{x_i}(x) + c_{z_i}(x)$. $\square$

3.3 An Optimal Protocol

Using the insights gained in the previous sections, we show that among all basic and separable protocols, the strong x-enforcing protocol gives rise to an optimal protocol simultaneously minimizing the price of anarchy and stability. Our main result involves the n-th harmonic number $\mathcal{H}_n = \sum_{i=1}^{n} \frac{1}{i}$.

THEOREM 3.11.

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

We will prove the theorem by two subsequent lemmas. In the first lemma, we prove that $\mathcal{H}_n$ is a lower bound on the price of stability for every basic cost sharing protocol. We then continue by presenting an algorithm that returns for any scheduling model a strongly decharged outcome of cost at most $\mathcal{H}_n$ times the cost of an optimal outcome. Together with the strong x-enforcing protocol we conclude that the price of anarchy of the thus defined protocol is precisely $\mathcal{H}_n$.

LEMMA 3.12. *For basic cost sharing protocols on scheduling models with n players and non-decreasing cost functions, the price of stability is at least $\mathcal{H}_n$. This lower bounds holds even for models with unit demands.*

PROOF. Consider the scheduling model (N, M, d, c) with n players that have unit demand $d_i = 1$ for all $i \in N$ and n machines with cost functions as in Table 1.

Table 1: Cost functions for machines used in the proof of Lemma 3.12

ℓ	$c_{a_1}(\ell)$	$c_{a_2}(\ell)$	...	$c_{a_i}(\ell)$	...	$c_{a_n}(\ell)$
0	0	0	...	0	...	0
1	$1+\epsilon$	$\frac{1}{2}$	...	$\frac{1}{i}$	...	$\frac{1}{n}$
> 1	$1+\epsilon$	n	...	n	...	n

for some small $\epsilon > 0$.

The only optimal outcome is clearly $y = (a_1, \ldots, a_1)$ with $C(y) = 1 + \epsilon$. An outcome z can only be a Nash equilibrium if it is weakly decharged (Lemma 3.6). We show that the cheapest weakly decharged outcomes are those in which each machine is used by exactly one player, which all have the same cost as $x = (a_1, \ldots, a_n)$. It is easy to see that outcome x is decharged and with $C(x) = \sum_{i=1}^{n} \frac{1}{i} = \mathcal{H}_n$ this proves the lemma.

If in an outcome z some machine other than a_1 is used by multiple players, then $C(z) \geq n$, thus such outcomes are more expensive than x. If in outcome z multiple players use machine a_1, say k players, then there are at least $k-1$ unused machines and for the cheapest of these, say machine $\hat{a}$, we have $c_{\hat{a}}(1) \leq \frac{1}{k}$. Thus, z is not weakly decharged as

$$c_{a_1}(z) = 1 + \epsilon > 1 = \sum_{i \in S_{a_1}(z)} \frac{1}{k} \geq \sum_{i \in S_{a_1}(z)} \min_{b \in M} c_b(b, z_{-i}).$$

Algorithm 1 Find strongly decharged outcome x

1: $k \leftarrow 1$ {stepnumber}
2: $x^1 \leftarrow y$ {starts with optimal outcome y}
3: $t_i \leftarrow 0$ for all $i \in N$ {stores when a player was last moved}
4: **while** there are charged machines **do**
5: $a^k \leftarrow \text{argmax}\,\{c_a(x^k) : a \in M \text{ is charged}\}$ {select the most expensive charged machine}
6: **if** $\min\{c_b(b, x^k_{-i}) : b \in M\} = 0$ for $i = \min S_{a^k}(x^k)$ **then**
7: {i can move to cost-free machine, called *Zero-move*}
8: $i^k \leftarrow \min S_{a^k}(x^k)$ {select smallest player}
9: **else if** $\max\{t_i : i \in S_{a^k}(x^k)\} > 0$ **then**
10: {a player on a^k was moved before, called *Shuffle*}
11: $i^k \leftarrow \text{argmax}\,\{t_i : i \in S_{a^k}(x^k)\}$ {select last moved player}
12: **else**
13: {no foreign players on a^k, called *Kick-off*}
14: $i^k \leftarrow \min S_{a^k}(x^k)$ {select smallest player}
15: **end if**
16: $b^k \leftarrow \text{argmin}\,\{c_b(b, x^k_{-i^k}) : b \in M\}$ {select cheapest available machine}
17: $x^{k+1} \leftarrow (b^k, x^k_{-i^k})$ {move player}
18: $t_{i^k} \leftarrow k$ {store stepnumber}
19: $k \leftarrow k + 1$ {iterate}
20: **end while**
21: **return** $x \leftarrow x^k$

Altogether, only such outcomes in which all machines are used by exactly one player are cheap weakly-enforceable outcomes. $\square$

While the previous Lemma showed that there sometimes are no weakly-enforceable outcomes cheaper than $\mathcal{H}_n$ times the cost of an optimal outcome, the following lemma shows that we always find strongly decharged outcomes of at most $\mathcal{H}_n$ times the cost of an optimal outcome.

LEMMA 3.13. *Any scheduling model (N, M, d, c) with an optimal outcome y has a strongly decharged outcome x with $C(x) \leq \mathcal{H}_n \cdot C(y) = \sum_{k=1}^{n} \frac{1}{k} \cdot C(y)$.*

PROOF. The desired outcome x is found by Algorithm 1. The algorithm takes as input an optimal outcome y. In each cycle k of the algorithm's main loop (lines 4 - 20), a player i^k on the most expensive charged machine a^k is selected (line 5) and moved to the cheapest available machine b^k (lines 16, 17). If possible, the algorithm selects a player who can be moved to a cost-free machine, this is called *Zero-move* (line 6). Otherwise, it selects a player that has been moved before in a last-in/first-out scheme which is maintained through the variables t_i that store the cycle in which each player was last moved. Such moves are called *Shuffles* (line 9). If neither a Zero-move nor a Shuffle is possible, the smallest player on the machine is selected, which is called *Kick-off* (line 12). The algorithm terminates when no charged machines are left.

First, we show that the algorithm terminates. To this end, observe that Shuffles are only performed when Zero-moves are not possible. Hence, if in cycle k a Shuffle is performed, the following inequalities hold.

$$\min_{b \in M} c_b(b, x^k_{-j}) > 0 \quad \text{for all } j \in S_{a^k}(x^k). \tag{3.16}$$

We now consider two cases. For $|S_{a^k}(x^k)| = 1$, we obtain

$$\begin{aligned} c_{a^k}(x^k) &> \min_{b \in M} c_b(b, x^k_{-i^k}) && (3.17)\\ &= c_{b^k}(b^k, x^k_{-i^k}) = c_{b^k}(x^{k+1}), && (3.18) \end{aligned}$$

where (3.17) follows because a^k is charged in x^k. Equality (3.18) follows since Algorithm 1 moves i^k to the cheapest available machine.

If $|S_{a^k}(x^k)| > 1$, then we obtain

$$\begin{aligned} c_{a^k}(x^k) &\geq \sum_{j \in S_{a_k}(x^k)} \min_{b \in M} c_b(b, x^k_{-j}) && (3.19)\\ &> \min_{b \in M} c_b(b, x^k_{-i^k}) && (3.20)\\ &= c_{b^k}(b^k, x^k_{-i^k}) = c_{b^k}(x^{k+1}), \end{aligned}$$

where (3.19) is valid because a^k is charged in x^k. The second inequality (3.20) holds because of (3.16) and the equalities follow as above. In both cases, a Shuffle moves the player to a strictly cheaper machine. To see that the algorithm terminates, we will now follow some player i over the course of the algorithm. Each Zero-move and each Shuffle take her to a strictly cheaper machine. If the player is moved in cycle k and is next moved by a Shuffle in cycle l, the cost of her machine $x_i^{k+1} = x_i^l$ may increase in the meantime as other players arrive on that machine. The algorithm assures by its last-in/first-out mechanism that these other players have been moved again before the Shuffle in cycle l and consequently the cost has decreased to the original level $c_{x_i^{k+1}}(x^{k+1}) \geq c_{x_i^l}(x^l)$. Since only machines with positive costs can be charged, this implies that after a Zero-move, the player will never again be considered for Shuffles. Hence, a player can be moved by at most one Kick-off, afterwards a sequence of Shuffles and thereafter only Zero-moves. The sequence of Shuffles is finite because each Shuffle takes the player to a strictly cheaper machine. Once the player has has been moved by a Zero-move, further Zero-moves are only possible if in between some other player arrives on the player's machine via a Kick-off or a Shuffle, but again this is only finitely often possible. Altogether, each player can only be moved finitely often and thus the algorithm terminates after a finite number of cycles.

To complete the proof, we show that the final outcome x has cost $C(x) \leq \mathcal{H}_n \cdot C(y)$. The concept of this final part of the proof is that in outcome x the cost of every used machine is determined by the player who has last moved there or, if there are no such players, the home players. For this, some new notation is needed. Let $p_i, i \in N$, correspond to the position (by index) of player i on her optimal machine y_i, i.e., on any machine a we have $p_j = 1$ for player $j = \max S_a(y)$, $p_{j'} = 2$ for $j' = \max(S_a(y) \backslash \{j\})$ and so on. Consequently, when some player i performs her Kick-off in cycle k, there are p_i players sharing her machine $a^k = y_i$ at that moment and she is the smallest of them. We obtain for machine b^k

that she is moved to

$$\begin{aligned} c_{b^k}(x^{k+1}) &= c_{b^k}(b^k, x^k_{-i}) = \min_{b\in M} c_b(b, x^k_{-i}) \\ &\le \frac{1}{p_i}\cdot \sum_{j\in S_{a^k}(x^k)} \min_{b\in M} c_b(b, x^k_{-j}) && (3.21) \\ &\le \frac{1}{p_i}\cdot c_{a^k}(x^k) && (3.22) \\ &\le \frac{1}{p_i}\cdot c_{y_i}(y), && (3.23) \end{aligned}$$

where the first inequality (3.21) is valid because i is the smallest of the p_i players on machine a^k in step k, the second inequality (3.22) holds because a^k is charged in x^k and the last inequality (3.23) holds because there are no foreign players on $a^k = y_i$ and hence $\ell_{a^k}(x^k) \le \ell_{a^k}(y) = \ell_{y_i}(y)$.

Since Shuffles and Zero-moves assign player i to cheaper machines, after her last move in cycle k', she is on machine $b^{k'}$ at cost $c_{b^{k'}}(x^{k'}) \le \frac{1}{p_i}\cdot c_{y_i}(y)$. Altogether, in the final outcome x, the cost of a machine $a \in M$ to which players have been moved is determined by the last player who was moved there, that is, $i_a := \operatorname{argmax}\{t_i : i \in S_a(x)\}$. We thus obtain $c_a(x) \le \frac{1}{p_{i_a}}\cdot c_{y_{i_a}}(y)$. For machines $a \in M$ that are used in x but where no player has been moved, the player $i_a := \max S_a(y)$ with $p_{i_a} = 1$ is still on machine a. In this case, the cost is bounded from above by $c_a(x) \le c_a(y) = \frac{1}{p_{i_a}}\cdot c_{y_{i_a}}(y)$. Unused machines $a \in M$ have cost $c_a(x) = 0$. Altogether, we obtain $c_a(x) = \frac{1}{p_{i_a}}\cdot c_{y_{i_a}}(y)$ for all $a \in M$ with $\ell_a(x) > 0$, and $c_a(x) = 0$ for all $a \in M$ with $\ell_a(x) = 0$. This yields the desired bound for the cost of outcome x, because now every used machine $a \in M$ has a unique player i_a that determines the machine's cost. We obtain

$$\begin{aligned} C(x) = \sum_{a\in M} c_a(x) &\le \sum_{\substack{a\in M\\ \ell_a(x)>0}} \frac{1}{p_{i_a}}\cdot c_{y_{i_a}}(y) \le \sum_{i\in N}\frac{1}{p_i}\cdot c_{y_i}(y) \\ &\le \sum_{a\in M} \mathcal{H}_{p_{\max}}\cdot c_a(y) = \mathcal{H}_{p_{\max}}\cdot C(y) \le \mathcal{H}_n\cdot C(y), \end{aligned}$$

where $p_{\max} := \max\{|S_a(y)| : a \in M\}$.

Observe that the bound for the price of anarchy obtained here can be much lower than $\mathcal{H}_n$ for scheduling models that have optimal outcomes, where the players are scattered over the machines and where therefore $p_{\max}$ is smaller than n. □

Remark 3.14. While Lemma 3.13 shows that an optimal outcome can be turned into a strongly decharged outcome of cost at most $\mathcal{H}_n$ times the cost of an optimal outcome, this holds true more generally: Algorithm 1 turns every outcome into a strongly decharged outcome with a cost increase of a factor at most $\mathcal{H}_n$. This may be useful if the computation of an optimal outcome is not possible in polynomial time. Still, Algorithm 1 does not run in polynomial time and this issue deserves further attention.

3.4 Non-Decreasing Cost per Unit

In this section we require that the cost functions are non-negative, non-decreasing and the per-unit costs $\frac{c(x)}{\ell(x)}$ are non-decreasing with respect to the load $\ell(x)$. Such functions are still quite rich as they 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 x-enforcing protocols: make all undesired outcomes unstable by charging some player a very high price.

Definition 3.15. (opt-enforcing protocol) Given a scheduling model (N, M, d, c) the *opt-enforcing protocol* takes as input an optimal outcome y. We again denote for any outcome z and machine a the set of foreign players on a by $S^1_a(z) = \{i \in S_a(z)\backslash S_a(y)\}$. Then, the opt-enforcing protocol assigns the cost sharing methods

$$\xi_i(z) := \begin{cases} d_i\cdot\dfrac{c_{z_i}(z)}{\ell_{z_i}(z)},\ \text{if } S^1_{z_i} = \emptyset & (3.24a) \\ c_{z_i}(z),\ \text{if } S^1_{z_i} \ne \emptyset \text{ and } i = \min S^1_{z_i}(z) & (3.24b) \\ 0,\ \text{else.} & (3.24c) \end{cases}$$

Under the opt-enforcing protocol, the players share the cost proportional to their job weights on all machines without foreign players. On machines with foreign players, the foreign player with the smallest index pays the entire cost of the machine.

THEOREM 3.16. *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 private cost functions. For stability it can easily be verified that for an instance (N, M, d, c) the optimal outcome y is a Nash equilibrium. We only proof the bound on the price of anarchy, showing that all Nash equilibria x are optimal outcomes using the Nash inequalities $\xi_i(x) \le \xi_i(y_i, x_{-i})$ for any $i \in N$. Two cases are to be considered for such a machine y_i: either it hosts foreign players $S^1_{y_i}(x)$ or $S^1_{y_i}(x) = \emptyset$. If there are foreign players on y_i, then one of them will pay for the entire cost there and hence (3.24c) gives $\xi_i(x) \le \xi_i(y_i, x_{-i}) = 0$. If there are no foreign players on y_i, then $d_i \le \ell_{y_i}(y_i, x_{-i}) \le \ell_{y_i}(y)$ yields

$$\frac{c_{y_i}(y_i, x_{-i})}{\ell_{y_i}(y_i, x_{-i})} \le \frac{c_{y_i}(y)}{\ell_{y_i}(y)},$$

because the cost per unit is non-decreasing. Plugging this into (3.24a) we have $\xi_i(x) \le \xi_i(y_i, x_{-i}) \le \xi_i(y)$. In both cases $\xi_i(x) \le \xi_i(y)$ for all $i \in N$ and thus

$$C(x) = \sum_{i\in N}\xi_i(x) \le \sum_{i\in N}\xi_i(y) = C(y)$$

which implies that every Nash equilibrium x is also an optimal outcome. □

4. UNIFORM PROTOCOLS

The separable protocols that we introduced so far were always tailored to some desirable outcome, either an enforceable outcome or even an optimal outcome. Since uniform protocols need to assign cost sharing methods independent of the set M, they cannot be based on specific outcomes. We show in this section that uniformity leads in general to an unbounded price of anarchy. Only for games in which the demands are integer multiples of each other we introduce the semi-ordered protocol that gives a price of anarchy of n. The question of $\min_{\xi\in\mathcal{U}_n} PoS(\xi)$ remains open.

4.1 Lower Bound

THEOREM 4.1. *There is no uniform protocol for which the price of anarchy has an upper bound. This holds even for models with at most* 3 *players,* 3 *machines and non-decreasing costs per unit.*

PROOF. The essence of uniform protocols is that adding machines to or removing them from the model does not change the cost shares of players using a certain machine, as long as the player set and the weight vector remain the same. This motivates the definition of cost share functions $\hat{\xi}_i$ that return the private cost ξ_i of player i as a function of the machine a that she uses and the set of players $S \subseteq N$ sharing the machine.

$$\hat{\xi}_i(a, S) := \xi_i(x) \,\forall\, a \in M, S \subseteq N, i \in S, x \in M^n : S_a(x) = S. \tag{4.1}$$

Nash equilibria can be expressed via cost share functions as follows. For all $i \in N, a \in M$ it holds that

$$\hat{\xi}_i(x_i, S_{x_i}(x)) \leq \hat{\xi}_i(a, S_a(x) \cup \{i\}). \tag{4.2}$$

For the proof of the theorem, we propose a number of scheduling models and show that for any uniform cost sharing protocol at least one of these models has a Nash equilibrium of more than q times the cost of an optimal outcome for arbitrary $q \geq 2$. Throughout the entire proof, the player set will always be $N = \{1, 2, 3\}$ with weights $d = (4, 3, 2)$. The machines will be a subset of $M = \{a_1, \ldots, a_7\}$ with cost functions as outlined in Table 2.

Table 2: Cost functions used in proof of Theorem 4.1

ℓ	$c_{a_1}(\ell)$	$c_{a_2}(\ell)$	$c_{a_3}(\ell)$	$c_{a_4}(\ell)$	$c_{a_5}(\ell)$	$c_{a_6}(\ell)$	$c_{a_7}(\ell)$
0	0	0	0	0	0	0	0
2	0	0	0	0	0	q^2	0
3	0	0	0	q^3	0	q^4	0
4	1	q	1	$2q^3$	q^3	$2q^4$	q^4
5	2	q^3	q^5		$2q^3$		
6	q^3	q^5					
7	q^4						

First, consider the model with machines $M_1 = \{a_1, a_2\}$ and their respective cost functions. The optimal outcome $y_1 = (a_2, a_1, a_1)$ has cost $C(y_1) = q + 2$, while the outcome $x_1 = (a_1, a_2, a_2)$ has cost $C(x_1) = q^3 + 1$. Either x_1 is a Nash equilibrium and hence the protocol has a price of anarchy greater than q or or one of the three players can reduce her private cost by choosing a different machine, which results by (4.2) in the following three cases.

a) $\hat{\xi}_1(a_2, \{1, 2, 3\}) < \hat{\xi}_1(a_1, \{1\}) = 1$. In this case, consider the model with machines $M_2 = \{a_2, a_3\}$. The optimal outcome $y_2 = (a_3, a_2, a_2)$ with $C(y_2) = q^3 + 1$ is not a Nash equilibrium and due to stability some other outcome x with cost $C(x) \geq q^5$ has to be a Nash equilibrium.

b) $\hat{\xi}_2(a_1, \{1, 2\}) < \hat{\xi}_2(a_2, \{2, 3\}) \leq q^3$. In this case, consider $M_3 = \{a_1, a_2, a_4\}$. The optimal outcome $y_3 = (a_1, a_2, a_4)$ has cost $C(y_3) = 1$, while the outcome $x_3 = (a_2, a_1, a_4)$ has cost $C(x_3) = q$. Either x_3 is a Nash equilibrium or, again, one of the players can reduce her private cost by choosing a different machine, which leads to the following cases.

  b.1) $\hat{\xi}_1(a_1, \{1, 2\}) < \hat{\xi}_1(a_2, \{1\}) = q$. This contradicts b), that is $\hat{\xi}_1(a_1, \{1, 2\}) = c_1(d_1 + d_2) - \hat{\xi}_2(a_1, \{1, 2\}) > q^4 - q^3 > q$.

  b.2) $\hat{\xi}_1(a_4, \{1, 3\}) < \hat{\xi}_1(a_1, \{1\}) = q$. In this case, consider $M_4 = \{a_2, a_4, a_5\}$. The optimal outcome $y_4 = (a_2, a_5, a_4)$ with $C(y_4) = q$ is not a Nash equilibrium and due to stability some other outcome x with cost $C(x) \geq q^3$ has to be a Nash equilibrium.

  b.3) Players 2 and 3 cannot reduce their private cost as $\xi_2(x_3) = c_{a_1}(x_3) = 0$ and $\xi_3(x_3) = c_{a_4}(x_3) = 0$.

c) $\hat{\xi}_3(a_1, \{1, 3\}) < \hat{\xi}_3(a_2, \{2, 3\}) \leq q^3$. In this case, consider $M_5 = \{a_1, a_2, a_6\}$. The optimal outcome $y_5 = (a_2, a_1, a_1)$ has cost $C(y_5) = q + 1$, while the outcome $x_5 = (a_1, a_2, a_6)$ has cost $C(x_5) = q^2 + 1$. Either x_5 is a Nash equilibrium or, again, one of the players can reduce her private cost by choosing a different machine.

  c.1) $\hat{\xi}_1(a_2, \{1, 2\}) < \hat{\xi}_1(a_1, \{1\}) = 1$. In this case, consider again $M_3 = \{a_1, a_2, a_4\}$. The optimal outcome $y_3 = (a_1, a_2, a_4)$ with $C(y_3) = 1$ is not a Nash equilibrium and due to stability some other outcome x with cost $C(x) \geq q$ has to be a Nash equilibrium.

  c.2) $\hat{\xi}_1(a_6, \{1, 3\}) < \hat{\xi}_1(a_1, \{1\}) = 1$. In this case, consider $M_6 = \{a_3, a_5, a_6\}$. The optimal outcome $y_6 = (a_3, a_5, a_6)$ with $C(y_6) = q^2 + 1$ is not a Nash equilibrium and due to stability some other outcome x with cost $C(x) \geq q^3$ has to be a Nash equilibrium.

  c.3) Player 2 cannot reduce her private cost because $\xi_2(x_5) = c_{a_2}(x_5) = 0$.

  c.4) $\hat{\xi}_3(a_1, \{1, 3\}) < \hat{\xi}_3(a_6, \{3\}) = q^2$. In this case, consider $M_7 = \{a_1, a_6, a_7\}$. The optimal outcome $y_7 = (a_1, a_7, a_6)$ with $C(y_7) = q^2 + 1$ is not a Nash equilibrium and due to stability some other outcome x with cost $C(x) \geq q^3$ has to be a Nash equilibrium.

  c.5) $\hat{\xi}_3(a_2, \{2, 3\}) < \hat{\xi}_3(a_6, \{3\}) = q^2$. This extends the original assumption from c) $\hat{\xi}_3(a_1, \{1, 3\}) < \hat{\xi}_3(a_2, \{2, 3\}) < q^2$ and therefore implies c.4).

Altogether, every uniform cost sharing protocol allows in at least one of the analyzed cases a Nash equilibrium of at least q times the cost of an optimal outcome for an arbitrary $q \geq 2$. Consequently, the price of anarchy is not bounded. ☐

4.2 Models with Restricted Weights

Although uniform protocols in general do not allow a bound on the price of anarchy, the following class of games permits uniform cost sharing protocols with a bounded price of anarchy. We assume that the player's weights are either uniform, i.e. $d_1 = \ldots = d_n$, or they are multiples of each other. In the following, we propose a semi-ordered protocol that a has a price of anarchy of at most n for such games. In this section, we assume that the players are indexed with their weights in non-increasing order: $d_1 \geq d_2 \geq \ldots \geq d_n$. The semi-ordered protocol lets the players one after the other choose a machine and lets them pay only for the additional cost they cause on that machine, thus making the choice of a player i independent of the choices of all players $j > i$.

Definition 4.2. (Semi-ordered Protocol) The semi-ordered protocol assigns for all $i \in N$

$$\xi_i(x) := c_{x_i}\Bigg(\sum_{j \in S_{x_i}(x) : j \leq i} d_j\Bigg) - c_{x_i}\Bigg(\sum_{j \in S_{x_i}(x) : j < i} d_j\Bigg). \tag{4.3}$$

THEOREM 4.3. *The semi-ordered protocol is uniform and its price of anarchy is at most n for instances, where the players' weights are multiples of each other, i.e. $d_i = q_i \cdot d_{i+i}$ for all $i < n$ and some $q_i \in \mathbb{N}$.*

PROOF. Budget balance, separability and uniformity of the cost sharing methods are clear. A Nash equilibrium can be found by asking the players in the order of their index to choose a machine that minimizes their private cost considering the choice of all previous players. For proving the bound on the price of anarchy, consider a model (N, M, d, c) which fulfills the restriction on the players' weights. Suppose y is an optimal outcome and x a Nash equilibrium. First, we show that $\xi_i(x) \leq \max_{j \leq i} \xi_j(y)$ holds for all $i \in N$, which is motivated by the idea that a player can always choose a machine that she or one of the larger players had chosen in the optimal outcome. To this end, fix player $i \in N$. On some machine $a \in \{y_1, \ldots, y_{i-1}\}$ there is in outcome x less load from the first $i-1$ players than in the optimal outcome y from the first i players. Due to the restriction on the players' weights this difference in load on machine a has to be at least d_i yielding

$$\sum_{j \in S_a(x): j<i} d_j + d_i \leq \sum_{j \in S_a(y): j \leq i} d_j.$$

Also, there is a player $k \leq i$ (hence $d_k \geq d_i$), $k \in S_a(y)$ who in outcome y uses machine a and for whom due to the weight restrictions

$$\sum_{j \in S_a(y): j<k} d_j \leq \sum_{j \in S_a(x): j<i} d_j < \sum_{j \in S_a(x): j<i} d_j + d_i \leq \sum_{j \in S_a(y): j \leq k} d_j.$$

Combining the above inequalities with (4.3) yields

$$\begin{aligned}
\xi_i(x) &\leq \xi_i(a, x_{-i}) \\
&= c_a\Big(\sum_{j \in S_a(x): j<i} d_j + d_i\Big) - c_a\Big(\sum_{j \in S_a(x): j<i} d_j\Big) \\
&\leq c_a\Big(\sum_{j \in S_a(y): j \leq k} d_j\Big) - c_a\Big(\sum_{j \in S_a(y): j<k} d_j\Big) \\
&= \xi_k(y) \leq \max_{j \leq i} \xi_j(y).
\end{aligned}$$

This implies:

$$C(x) = \sum_{i \in N} \xi_i(x) \leq \sum_{i \in N} \max_{j \leq i} \xi_j(y) \leq n \cdot \max_{j \in N} \xi_j(y) \leq n \cdot C(y)$$

proving the claim. □

5. REFERENCES

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

[2] S. Aland, D. Dumrauf, M. Gairing, B. Monien, and F. Schoppmann. Exact price of anarchy for polynomial congestion games. In *Proc. of STACS*, pages 218–229, 2006.

[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] B. Awerbuch, Y. Azar, and A. Epstein. The price of routing unsplittable flow. In *Proc. of STOC*, pages 57–66. ACM, 2005.

[5] K. Bhawalkar, M. Gairing, and T. Roughgarden. Weighted congestion games: Price of anarchy, universal worst-case examples, and tightness. In *Proc. of ESA*, pages 17–28. Springer, 2010.

[6] I. Caragiannis. Efficient coordination mechanisms for unrelated machine scheduling. In *Proc. of SODA*, pages 815–824, 2009.

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

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

[9] G. Christodoulou and E. Koutsoupias. The price of anarchy of finite congestion games. In *Proc. of STOC*, pages 67–73, 2005.

[10] G. Christodoulou, E. Koutsoupias, and A. Nanavati. Coordination mechanisms. In *Proc. of ICALP*, pages 45–56. Springer, 2004.

[11] J. R. Correa, A. S. Schulz, and N. E. Stier-Moses. Selfish routing in capacitated networks. *Math. of Oper. Res.*, 29:961–976, 2004.

[12] E. Even-Dar, A. Kesselman, and Y. Mansour. Convergence time to Nash equilibrium in load balancing. *ACM Trans. Algorithms*, 3(3):32, 2007.

[13] D. Fotakis, S. Kontogiannis, E. Koutsoupias, M. Mavronicolas, and P. Spirakis. The structure and complexity of Nash equilibria for a selfish routing game. In *Proc. of ICALP*, pages 123 – 134, 2002.

[14] M. Gairing, B. Monien, and K. Tiemann. Routing (un-) splittable flow in games with player-specific linear latency functions. In *Proc of ICALP*, pages 501–512, 2006.

[15] S. Ieong, R. McGrew, E. Nudelman, Y. Shoham, and Q. Sun. Fast and compact: A simple class of congestion games. In *Proc. AAAI*, pages 489–494, 2005.

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

[17] E. Koutsoupias and C. H. Papadimitriou. Worst-case equilibria. In *STACS*, pages 404–413, 1999.

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

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

[20] R. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *Inter. J. Game Theory*, 2(1):65–67, 1973.

[21] T. Roughgarden. The price of anarchy is independent of the network topology. *Journal of Computer and System Sciences*, 67:341–364, 2002.

[22] T. Roughgarden. *Selfish Routing and the Price of Anarchy*. The MIT Press, 2005.

[23] T. Roughgarden and É. Tardos. How bad is selfish routing? *J. ACM*, 49(2):236–259, 2002.

[24] R. Srikant. *The Mathematics of Internet Congestion Control*. Birkhäuser Boston, 2003.