Group-Strategyproof Cost Sharing Mechanisms for Makespan and Other Scheduling Problems ⋆

Janina Brenner, Guido Schäfer ∗

Technische Universität Berlin
Institut für Mathematik, Sekr. MA 5–1
Straße des 17. Juni 136
10623 Berlin, Germany

Abstract

Classical results in economics show that no truthful mechanism can achieve budget balance and efficiency simultaneously. Roughgarden and Sundararajan recently proposed an alternative efficiency measure, which was subsequently used to exhibit that many previously known cost sharing mechanisms approximate both budget balance and efficiency. In this work, we investigate cost sharing mechanisms for combinatorial optimization problems using this novel efficiency measure, with a particular focus on scheduling problems. Our contribution is threefold: First, for a large class of optimization problems that satisfy a certain cost-stability property, we prove that no budget balanced Moulin mechanism can approximate efficiency better than $\Omega(\log n)$, where n denotes the number of players in the universe. Second, we present a group-strategyproof cost sharing mechanism for the minimum makespan scheduling problem that is tight with respect to budget balance and efficiency. Finally, we show a general lower bound on the budget balance factor for cost sharing methods, which can be used to prove a lower bound of $\Omega(n)$ on the budget balance factor for completion and flow time scheduling objectives.

Key words: cooperative game theory, mechanism design, cost sharing mechanisms, truthful mechanisms, scheduling problems.

⋆ This work was supported by the DFG Research Center Matheon "Mathematics for key technologies". An extended abstract of this work appeared in *Proceedings of the 24th Symposium on Theoretical Aspects of Computer Science*, 2007.
∗ Corresponding author.
Email addresses: brenner@math.tu-berlin.de (Janina Brenner), schaefer@math.tu-berlin.de (Guido Schäfer).

1 Introduction

Many combinatorial optimization problems are concerned with establishing a good or service at a minimum cost. Often, these problems can be viewed as consisting of a set of users (or agents, players) that act strategically in order to receive this service. For instance, in network design problems, users wish to be connected to a network or a specific point of interest. In routing problems, agents want their goods or information to be transferred from one point to another. In the scheduling context, we can imagine both machines or jobs to be owned by agents who follow their selfish interests. In all these settings, the problem is, besides finding a way of providing the service, to distribute the resulting cost among the users in a fair manner. Meanwhile, the service provider may not be able to offer the service to the entire user set and must therefore decide upon a subset of users that are served.

In this paper, we study *cost sharing mechanisms* for combinatorial optimization problems, with a particular focus on scheduling problems. The general setting is as follows. We are given a set U of n players that are interested in a certain service. Every player $i \in U$ has a private *utility* $u_i \geq 0$ for receiving this service and announces a *bid* $b_i \geq 0$ which designates the maximum price she is willing to pay. Associated with the underlying optimization problem, we are given a cost function $C : 2^U \to \mathbb{R}^+$ describing the minimum cost of serving a set of players $S \subseteq U$.

A *cost sharing mechanism* M first solicits all bids $\{b_i\}_{i \in U}$ from players in U, and based on these bids (i) determines a set $S \subseteq U$ of players that receive the service, and (ii) for every player $i \in S$, fixes a non-negative payment $x_i(S)$ that she has to pay for the service. This payment is usually referred to as the *cost share* of a player $i \in S$. We assume that the mechanism complies with the following three natural assumptions: (i) a player is not charged more than her bid, (ii) a player is charged only if she receives service, and (iii) a player is guaranteed to receive service if she reports a sufficiently high bid.

Define the *benefit* of a player i as $u_i - x_i$ if i receives service and as zero otherwise. We assume that each player's strategy is to maximize her benefit. Since the outcome computed by the cost sharing mechanism depends on the bids $\{b_i\}_{i \in U}$, a player may have an incentive to misreport her actual utility, i.e., to declare a bid $b_i \neq u_i$, if advantageous.

There are several desirable properties of a cost sharing mechanism: A cost sharing mechanism M is called *strategyproof* if bidding truthfully, i.e., announcing $b_i = u_i$, is a dominant strategy for every player. If this is true even if players collude, then we call a mechanism *group-strategyproof*. A mechanism is *β-budget balanced* if the sum of the cost shares charged to the players in S

deviates by at most a factor $\beta \geq 1$ from the actual cost $C(S)$, i.e.,

$$\frac{1}{\beta} \cdot C(S) \leq \sum_{i \in S} x_i(S) \leq C(S). \tag{1}$$

We say that the cost shares satisfy *β-cost recovery* if the first inequality holds; they are *competitive* if the latter inequality is fulfilled. For a set $S \subseteq U$, define $u(S) := \sum_{i \in S} u_i$. A cost sharing mechanism M is called *efficient* if it selects a set of players that maximizes the *social welfare* $u(S) - C(S)$.

Classical results in economics [1,2] state that no truthful mechanism can approximate budget balance and efficiency simultaneously; even for simple cost functions and if only strategyproofness is required. As a consequence, most of the previous work has concentrated on either achieving budget balance or efficiency. Very recently, Roughgarden and Sundararajan [3] introduced an alternative efficiency measure that attempts to circumvent these intractability results. They define the *social cost* of a set $S \subseteq U$ as $\Pi(S) := u(U \setminus S) + C(S)$. A mechanism is said to be *α-approximate* if the set of players it determines has social cost at most α times the minimum social cost (over all subsets of U). It is not hard to see that a set S minimizes the social cost iff it maximizes the social welfare.

A large class of group-strategyproof cost sharing mechanisms are so-called *Moulin mechanisms*, based on a framework due to Moulin and Shenker [4]. This framework provides a means to obtain group-strategyproof cost sharing mechanisms from *cross-monotonic cost sharing methods* (definitions are given below). Moreover, Immorlica et al. [5] prove that every group-strategyproof cost sharing mechanism (satisfying some natural conditions) corresponds to a cross-monotonic cost sharing method. Roughgarden and Sundararajan [3] revealed a relation between the approximability of a Moulin mechanism and a property of the underlying cost sharing method, which they termed *α-summability* (definition is given below).

One focus of this paper is on cost sharing mechanisms for *parallel machine scheduling* problems. In the classical setting, we are given a set N of jobs that have to be executed on m parallel machines. The goal is to assign all jobs to the machines such that a certain objective function, such as the makespan or the sum of all completion times, is minimized. In the cost sharing context, we assume that every job is owned by a player who acts strategically in order to get his job processed at a low cost. Here, the cost that is to be distributed among the players depends on the respective objective function. It is very natural to suppose that the cost incurred by the service provider is the amount of time that he needs until all jobs are completed, leading to the minimum makespan cost function. However, one can also imagine that the service provider aims at minimizing the total time that jobs spend in the system or other completion time related objective functions.

1.1 Our Results

In this paper, we study cost sharing methods for optimization problems in light of the new efficiency measure introduced by Roughgarden and Sundararajan [3]. Our contribution is threefold:

1. Lower Bound on Approximability of Cost Sharing Mechanisms.

We present a general inapproximability result for cost sharing methods for combinatorial optimization problems. In particular, we prove that there is no cost sharing method that is α-summable and satisfies β-cost recovery for any $\alpha < H_n/\beta$, where n denotes the number of players. Our proof holds if the underlying cost function satisfies a certain cost-stability property. As a consequence, our result implies a lower bound of $\Omega(\log n)$ on the approximability of Moulin mechanisms for various optimization problems, such as, for instance, facility location, minimum spanning tree (and thus also minimum Steiner tree and forest), single-source rent-or-buy, minimum makespan scheduling, etc. Despite its generality, our lower bound is tight for some specific problems such as facility location and minimum makespan scheduling.

2. Optimal Cost Sharing Method for Makespan Scheduling.

We study the *minimum makespan scheduling problem*, one of the most fundamental problems in scheduling theory, in a cost sharing context. In this problem, we are given a set of jobs N, each of which is owned by a selfish player. The objective is to assign the chosen set of jobs to m parallel machines such that the maximum completion time is minimized. We develop a cross-monotonic cost sharing method for this problem which is $(2 - 1/m)$-budget balanced and $(H_n + 1)$-approximate. With a slightly refined analysis, one can show that our method achieves a budget balance factor of $2 - 2/(m + 1)$; this is tight with respect to both budget balance and approximability.

Related to this result is the recent work of Bleischwitz and Monien [6]. The authors present a cross-monotonic $(2-2/(m+1))$-budget balanced cost sharing method for the minimum makespan scheduling problem. However, as we argue below, their cost sharing mechanism does not approximate social cost.

3. Lower Bound on Budget Balance of Cost Sharing Mechanisms.

We present a generic lower bound showing that no cross-monotonic and β-budget balanced cost sharing method exists for any $\beta < f(n)$, where f is a function that measures the maximum rate of increase of the underlying cost function C. For example, for every fixed player set, f is at least the ratio between the cost of the whole set and the sum of the costs of all its singleton subsets. We prove that this lower bound even holds for cost sharing methods in the *β-core* (definition is given below). We use this general approach to prove negative results for several fundamental scheduling problems in a cost sharing context. Namely, we show a lower bound of $\Omega(n)$ on the budget balance factor of cross-monotonic cost sharing methods for all scheduling problems in which we aim at minimizing the total (weighted) completion (or flow) time. We prove that the same bound also applies to average (weighted) completion (or flow) time objectives.

1.2 Previous and Related Work

The development of cost sharing mechanisms for combinatorial optimization problems has recently attracted a lot of attention in the theoretical computer science literature.

The framework of Moulin and Shenker [4] has been applied to game-theoretic variants of classical optimization problems such as fixed tree multicast [7–9], submodular cost sharing [4], Steiner trees [10,11], facility location, single-source rent-or-buy network design [12–14] and Steiner forests [15]. Lower bounds on the budget balance factor that is achievable by a cross-monotonic cost sharing mechanism are given in [5,16]. Very recently, researchers started to investigate cost sharing mechanisms in light of the novel efficiency measure of Roughgarden and Sundararajan; see [3,17–19].

The problem of scheduling independent jobs on parallel machines is well-studied for various objective functions. The minimum makespan version $P| \, |C_{\max}$ is shown to be NP-complete by Garey and Johnson [20]. Lenstra, Shmoys and Tardos gave a PTAS for this problem [21]. The minimum weighted completion time ($P| \, |\sum_i w_i C_i$) and average weighted completion time ($P| \, |\sum_i w_i C_i/n$) scheduling problems are also NP-complete as proved by Lenstra [22]. A PTAS for both problems has been given in [23]. However, for unit processing times or equal weights, all three problems are polynomially solvable [24,25].

Very notably, although network design problems have been studied extensively in a cost sharing context, very little attention has been given to scheduling problems; in particular if jobs are assumed to act strategically, and group-strategyproofness is a desirable objective. In most of the previous works, authors have either concentrated on scheduling problems where machines act selfishly [26–28], or strategyproofness (but not group-strategyproofness) is an issue [29,30].

After the publication of this paper, substantial progress has been made in the theory of cost sharing mechanims. Mehta, Roughgarden, and Sundararajan [31] introduced a new class of cost sharing mechanisms called *acyclic mech-*

anisms. These mechanisms generalize Moulin mechanisms and as such leave more flexibility to improve upon the approximation guarantees with respect to budget balance and social cost. However, they achieve a slightly weaker notion of truthfulness called *weak group-strategyproofness*. A mechanism is *weakly group-strategyproof* [31,32] if no coordinated bidding of a coalition $S \subseteq U$ can ever strictly increase the utility of *every* player in S. For scheduling problems with completion time related objectives, Brenner and Schäfer [33] define acyclic mechinsms with constant approximation factors for both budget balance and social cost. These results are in sharp contrast with the lower bounds for Moulin mechanisms that we prove in this paper.

1.3 Organization of Paper

The paper is structured as follows: In Section 2, we introduce some additional notation and concepts that are used in subsequent sections. The general lower bound on the approximability of cost sharing mechanisms is presented in Section 3. Our tight cost sharing mechanism for the minimum makespan scheduling is given in Section 4. The negative results with respect to approximating the budget balance factor for certain cost functions together with its applications to completion time related scheduling problems is stated in Section 5. Finally, we offer some conclusions in Section 6.

2 Preliminaries

2.1 Moulin Mechanisms

A *cost sharing method* ξ is a function $\xi : U \times 2^U \to \mathbb{R}^+$ that assigns to each user $i \in U$ and subset $S \subseteq U$ a non-negative cost share $\xi(i, S)$. We define $\xi(i, S) := 0$ for all $i \in U \setminus S$, for all $S \subseteq U$. ξ is *cross-monotonic* if the cost share of a player does not increase as the player set grows; more formally, for all $S' \subseteq S \subseteq U$ and for every $i \in S'$, it holds that $\xi(i, S') \geq \xi(i, S)$.

Similar to the definition in (1), ξ is *β-budget balanced* if

$$\forall S \subseteq U : \quad \frac{1}{\beta} \cdot C(S) \leq \sum_{i \in S} \xi(i, S) \leq C(S).$$

We say that ξ satisfies *β-cost recovery* if the first inequality holds; it is *competitive* if the latter inequality is fulfilled. If $\beta = 1$, we simply call the cost sharing mechanism budget balanced.

Moulin and Shenker [4] showed that, given a budget balanced and cross-monotonic cost sharing method ξ, the following cost sharing mechanism $M(\xi)$ satisfies budget balance and group-strategyproofness: Initially, let $S := U$. If for each player $i \in S$, the cost share $\xi(i, S)$ is at most her bid b_i, we stop. Otherwise, remove from S all players whose cost shares are larger than their bids, and repeat. Eventually, let S^M be the final player set and define the payments as $x_i(S^M) := \xi(i, S^M)$ for all $i \in S^M$. Jain and Vazirani [10] later observed that the result of Moulin and Shenker also carries over to approximately budget balanced and cross-monotonic cost sharing methods.

Yet another fairness concept in cooperative game theory that we use in this paper is the *β-core*. A cost sharing method ξ is in the *β-core* iff it is β-budget balanced and

$$\forall S' \subseteq S \subseteq U: \quad \sum_{i \in S'} \xi(i, S) \leq C(S').$$

2.2 Social Welfare vs. Social Cost

Recall that we define the *social welfare* of a set $S \subseteq U$ as $u(S) - C(S)$, where $u(S) := \sum_{i \in S} u_i$. A cost sharing mechanism M is said to be *efficient* if it selects a set S^M of players that maximizes the social welfare (assuming truthful bidding). An alternative measure of efficiency that we consider in this paper is *social cost*: Define the *social cost* of a set $S \subseteq U$ as

$$\Pi(S) := u(U \setminus S) + C(S).$$

A mechanism M is said to be *α-approximate* if it computes a final set S^M of social cost at most α times the minimum over all sets $S \subseteq U$, i.e., $\Pi(S^M) \leq \alpha \cdot \Pi(S)$ for all $S \subseteq U$. Observe that for every set $S \subseteq U$, $u(U) - \Pi(S) = u(S) - C(S)$. Since $u(U)$ is a constant, a set S minimizes social cost iff it maximizes social welfare.

Roughgarden and Sundararajan [3] revealed a relation between the approximability of a Moulin mechanism $M(\xi)$ and a property of the cost sharing method ξ: Assume we are given an arbitrary order σ on a subset $S \subseteq U$ of players, i.e., $S = \{i_1, \dots, i_{|S|}\}$, where $i_j \prec_\sigma i_k$ if and only if $1 \leq j < k \leq |S|$. We define $S_j \subseteq S$ as the (ordered) set of the first j players of S according to the order σ. A cost sharing method ξ is *α-summable* if for every order σ and every subset $S \subseteq U$:

$$\sum_{j=1}^{|S|} \xi(i_j, S_j) \leq \alpha \cdot C(S). \tag{2}$$

The authors proved that the Moulin mechanism $M(\xi)$ is $(\alpha + \beta)$-approximate and β-budget balanced if the underlying cost sharing method ξ is α-summable and β-budget balanced. Moreover, they showed that $\max\{\alpha, \beta\}$ is a lower bound on the approximability of $M(\xi)$.

In this paper, we use $[n]$ to denote the set $\{1, \dots, n\}$. We define H_n to be the *n-th harmonic number*, i.e., $H_n := \sum_{i=1}^{n} 1/i$. As n grows to infinity, $H_n = \log n + \gamma$, where $\gamma \approx 0.577$ denotes the Euler-Mascheroni constant. Hence, $H_n = \Theta(\log n)$ and we use both values interchangeably. Unless stated otherwise, we denote by n the cardinality of the universe U.

3 A General Lower Bound on Summability

In this section, we prove a lower bound of $\Omega(\log n)$ on the summability of cost sharing methods, where n denotes the number of players in U. Our lower bound holds for every optimization problem which contains a so-called *cost stable* instance. Intuitively, we call an instance cost stable if it contains a significantly large player set whose cost does not deviate too much from the cost of any of its subsets. This property is fulfilled by a variety of combinatorial optimization problems such as facility location, Steiner tree, minimum makespan scheduling, etc. Together with the recent result of Roughgarden and Sundararajan [3], our result shows that for all these problems, the approximability of Moulin mechanisms cannot be better than $\Omega(\log n)$.

Theorem 1 *Consider an instance of a combinatorial optimization problem on a player set U inducing a cost function C. Suppose that there is a set $S \subseteq U$ of size $|S| \geq |U|/\gamma$ for some constant $\gamma \geq 1$ such that*

$$C(S') \geq \frac{1}{\delta} \cdot C(S)$$

for all $S' \subseteq S$ and some constant $\delta \geq 1$. Let ξ be a cost sharing method for this problem that satisfies the β-cost recovery condition. Then, ξ is not α-summable for any $\alpha < H_{\lceil n/\gamma \rceil}/(\beta \cdot \delta)$.

PROOF. It is sufficient to prove that there exists an order σ on U such that

$$\sum_{j=1}^{|S|} \xi(i_j, S_j) \geq \frac{H_{\lceil n/\gamma \rceil}}{\beta \cdot \delta} \cdot C(S),$$

where S_j is the set of the first j players in S and i_j is the jth player of S (ordered according to σ).

We construct σ by determining the sets S_j and players i_j inductively as follows. Initially, set $j = |S|$ and assign $S_j = S$. Now, suppose we have determined sets $S_{|S|}, \ldots, S_j$. By an average argument, there must exist a player $i \in S_j$ such that

$$\xi(i, S_j) \geq \frac{C(S_j)}{\beta \cdot |S_j|} = \frac{C(S_j)}{\beta \cdot j} \geq \frac{C(S)}{\beta\delta \cdot j},$$

since ξ satisfies the β-cost recovery condition. The last inequality holds because $S_j \subseteq S$. Assign $i_j := i$ and $S_{j-1} := S_j \setminus \{i_j\}$.

Let $S = \{i_1, \ldots, i_{|S|}\}$ be the set of players in S ordered according to the order σ constructed above. We have

$$\sum_{j=1}^{|S|} \xi(i_j, S_j) \geq \left(1 + \frac{1}{2} + \cdots + \frac{1}{|S|}\right) \cdot \frac{C(S)}{\beta\delta} \geq \frac{H_{\lceil n/\gamma \rceil}}{\beta\delta} \cdot C(S),$$

where we exploit that $|S| \geq n/\gamma$ and $|S| \in \mathbb{N}$. □

This lower bound applies to many problems, as e.g. to the following ones:

Example 2 (Fixed-tree Multicast Problem) *Players are located at vertices of an undirected graph and wish to receive a broadcasting service which is produced in a root vertex. The cost of serving a set of players U is the cost of a minimum spanning tree containing U and the root. An instance fulfilling the conditions of the above theorem is the one in which all players are located on the same vertex which is connected to the root by an edge of length 1. The lower bound for this problem has been shown in [3].*

Example 3 (Facility Location Problem) *Players are located at vertices and wish to be connected to an open facility. Facilities can be opened at a given subset of vertices. Here, a sample instance is the one in which there is only one vertex v at which a facility may be opened, and all players are located directly on v. Then, the cost of a solution is independent of the number of players and equal to the opening cost of the facility. This lower bound is tight, as has been shown in [18].*

Another example for which Theorem 1 applies is the makespan machine scheduling problem that we define in Section 4. There, we show that the bound on summability is tight for this problem.

We remark that there are stronger lower bounds for e.g. the Steiner tree and Steiner forest problems [18].

4 Minimum Makespan Scheduling

We consider the classical *minimum makespan scheduling problem*. We are given a set of n jobs N that have to be scheduled on m identical machines. Each job $i \in N$ has a non-negative *processing time* p_i, which is the time needed to execute i on one of the machines. We denote the completion time of job i by C_i. Every machine can execute at most one job at a time; preemption of jobs is not allowed. The objective is to schedule all jobs in N on the m machines such that the maximum completion time $\max_{i \in N} C_i$, also called *makespan*, is minimized. Following the naming scheme introduced by Graham et al. [34], this problem is referred to as $P| |C_{\max}$.

In a game-theoretic variant of the machine scheduling problem, each job is associated with a player, who wants her job to be processed on one of the m machines. We therefore identify the universe of players U with the set of jobs N. The cost $C(S)$ incurred to schedule all jobs in S is the minimum makespan. We are interested in designing a cost sharing mechanism for the minimum makespan scheduling problem that is β-budget balanced and α-approximate for every possible instance.

Let $p_{\max}(S)$ denote the maximum processing time over all jobs in S. Define $\mu(S)$ as the average machine load, i.e., $\mu(S) := \sum_{i \in S} p_i/m$. The following fact is folklore (see, e.g., [35]).

Fact 4 *For a given set $S \subseteq U$ of jobs, let $C(S)$ be the makespan of an optimal schedule for S. The following two inequalities hold:*

(1) $C(S) \leq \mu(S) + (1 - \frac{1}{m}) \cdot p_{\max}(S)$;

(2) $C(S) \geq \max\{\mu(S), p_{\max}(S)\}$.

4.1 Cross-Monotonic Cost Shares

Bleischwitz and Monien [6] describe a cross-monotonic cost sharing method ξ^{BM} for the above machine scheduling problem. We briefly review their cost sharing method. [1]

We call a job i *large* with respect to S if $p_i = p_{\max}(S)$ and *small* otherwise. Let $\ell(S)$ be the number of large jobs in S. Given a subset $S \subseteq U$ of the jobs,

[1] At first sight, the cost shares that we state here differ from the ones defined by Bleischwitz and Monien in [6]. However, it can easily be verified that both definitions are in fact equivalent; we feel that the definition we present here is more intuitive.

we define the cost share of $i \in S$ as:

$$\xi^{\mathrm{BM}}(i,S) := \begin{cases} \dfrac{p_i}{m} + \dfrac{p_i - \mu(S)}{\ell(S)} & \text{if } p_i = p_{\max}(S) \text{ and } p_i > \mu(S), \\ \dfrac{p_i}{m} & \text{otherwise.} \end{cases} \tag{3}$$

The intuition is as follows: Every job gets a cost share of p_i/m. If the average machine load $\mu(S)$ is less than the maximum processing time $p_{\max}(S)$, every large job additionally obtains an equal share of the cost $p_{\max}(S) - \mu(S)$. We summarize one of the main results of Bleischwitz and Monien [6] in the following theorem.

Theorem 5 ([6]) ξ^{BM} *is a* $(2m/(m+1))$*-budget balanced cross-monotonic cost sharing method for the minimum makespan scheduling problem. Moreover, there is no* β*-budget balanced cross-monotonic cost sharing method* ξ *for this problem, for any* $\beta < 2m/(m+1)$*.*

Albeit Theorem 5 proves that the Moulin mechanism $M(\xi^{\mathrm{BM}})$, driven by the cost sharing method ξ^{BM} by Bleischwitz and Monien, is optimal with respect to budget balance, we show below that it is far from being optimal with respect to social cost. In fact, the social cost of the final set S^M output by $M(\xi^{\mathrm{BM}})$ can be as large as $n/2$ times the optimal social cost, where n is the number of jobs in the universe U.

Lemma 6 *For every* $n \in \mathbb{N}$*, there exists an instance of the minimum makespan scheduling problem such that the cost sharing method* ξ^{BM} *is not* α*-summable for any* $\alpha < n/2$*.*

PROOF. It is sufficient to define an instance of the minimum makespan scheduling problem on n jobs and a permutation σ for which the cost share sum in (2) with respect to ξ^{BM} is at least $n/2$ times the minimum makespan.

Let $U := \{i_1, \dots, i_m\}$ be an (ordered) set of m jobs, where $m = n$ is the number of machines. Define the processing time of job i_j to be $p_{i_j} := 1 + (j-1)\epsilon$ for all $j \in [m]$ and some small $\epsilon > 0$. Since the number of jobs equals the number of machines, the makespan of an optimal assignment for U is $C(U) = 1 + (m-1)\epsilon$.

Observe that the processing time of job i_j, $j \in [m]$, is maximum among all jobs in the set $S_j = \{i_1, \dots, i_j\}$, i.e., i_j is large. Furthermore, i_j is the only large job in S_j and thus $\ell(S_j) = 1$. The average machine workload of S_j is

$$\mu(S_j) = \frac{1}{m}\sum_{l=1}^{j} p_{i_l} = \frac{1}{m}\left(j + \frac{j(j-1)\epsilon}{2}\right) \leq 1 + (j-1)\epsilon = p_{\max}(S_j).$$

Hence, the cost share that job i_j obtains with respect to S_j is

$$\xi^{\text{BM}}(i_j, S_j) = \frac{p_{i_j}}{m} + p_{i_j} - \mu(S_j) = p_{i_j} - \mu(S_{j-1}),$$

where we define $S_0 := \emptyset$. We obtain

$$\xi^{\text{BM}}(i_j, S_j) = (1 + (j-1)\epsilon) - \frac{1}{m}\left((j-1) + \frac{(j-1)(j-2)\epsilon}{2}\right) \geq 1 - \frac{j-1}{m}.$$

Therefore,

$$\sum_{j=1}^{m} \xi^{\text{BM}}(i_j, S_j) \geq m - \frac{m(m-1)}{2m} = \frac{m}{2} + \frac{1}{2} \geq \frac{m}{2}(1 + (m-1)\epsilon) = \frac{m}{2} \cdot C(U),$$

where the last inequality holds if we choose ϵ sufficiently small. $\square$

Intuitively, this high summability gives voice to the fact that processing times exceeding the average workload $\mu(S)$ are punished in an unfair manner: Instead of sharing the additional cost of $p_{\max}(S) - \mu(S)$ among all jobs for which $p_i > \mu(S)$, only those jobs attaining the maximum processing time come up for it. We tackle this problem in the next section.

4.2 Approximate Cost Shares

We continue by proposing new cost shares ξ^{BS} for the minimum makespan scheduling problem that are still $(2 - 1/m)$-budget balanced and cross-monotonic, but concurrently $(H_n + 1)$-summable. This is tight in terms of both budget balance and summability.

We use a different definition of *small* and *large* jobs here: A job i is *large* with respect to S iff $p_i > \mu(S)$ and *small* otherwise. The cost share of a job $i \in S$ with respect to S is defined as

$$\xi^{\text{BS}}(i, S) := \begin{cases} \dfrac{p_i}{m} + \displaystyle\int_{\mu(S)}^{p_i} \frac{1}{|\{j \in S : p_j \geq t\}|} \,\mathrm{d}t & \text{if } p_i > \mu(S), \\[2ex] \dfrac{p_i}{m} & \text{otherwise.} \end{cases} \tag{4}$$

Intuitively, every job receives a cost share of p_i/m. A large job i obtains some additional cost share: for every time instant $t \in [\mu(S), p_i]$, i shares the cost of $1\mathrm{d}t$ evenly with all other jobs in S whose processing time is at least t.

We show that ξ^{BS} is a cost sharing method that satisfies cross-monotonicity, approximate budget balance and summability.

Theorem 7 *ξ^{BS} is a cross-monotonic, $(2-1/m)$-budget balanced and (H_n+1)-summable cost sharing method for the minimum makespan scheduling problem.*

Our cost sharing method is essentially tight: Bleischwitz and Monien [6] proved that no cross-monotonic cost sharing method for this problem achieves a budget balance factor better than $(2-1/m)$. Moreover, using Theorem 1 we show (see Corollary 11 below) that no cost sharing method that satisfies the β-cost recovery condition can be α-summable for any $\alpha < H_n/\beta$.

The proof of Theorem 7 follows from Lemmas 8, 9 and 10 that are given below.

Lemma 8 *ξ^{BS} is cross-monotonic.*

PROOF. Consider some set $S \subseteq U$ and a job $i \in S$. We prove that if a new job $j \notin S$ is added to S, the cost share of i does not increase.

If i was small in S, then it remains small, and hence i's cost share stays p_i/m. If i was large in S and becomes small in $S \cup \{j\}$, then i's cost share decreases to p_i/m. It remains to show that the cost share of i does not increase if i stays large. Note that by adding job j, the number of jobs whose processing time is at least t for some $t \geq 0$ does not decrease. Moreover, we have

$$\int_{\mu(S)}^{p_i} \frac{1}{|\{j \in S : p_j \geq t\}|} \,\mathrm{d}t \geq \int_{\mu(S\cup\{j\})}^{p_i} \frac{1}{|\{j \in S \cup \{j\} : p_j \geq t\}|} \,\mathrm{d}t,$$

since $\mu(S) \leq \mu(S \cup \{j\})$. This concludes the proof. □

We show next that the budget balance condition is satisfied.

Lemma 9 *ξ^{BS} is $(2-1/m)$-budget balanced.*

PROOF. With the cost share definition in (4) we have

$$\begin{aligned}\sum_{i\in S} \xi(i,S) &= \sum_{i\in S} \frac{p_i}{m} + \sum_{i\in S:\, p_i > \mu(S)} \int_{\mu(S)}^{p_i} \frac{1}{|\{j \in S : p_j \geq t\}|} \,\mathrm{d}t \\ &= \mu(S) + \int_{\mu(S)}^{p_{\max}(S)} 1 \,\mathrm{d}t = \max\{\mu(S), p_{\max}(S)\}.\end{aligned}$$

By Fact 4, $C(S) \geq \max\{\mu(S), p_{\max}(S)\}$, which proves competitiveness. Moreover, the cost shares satisfy $(2-1/m)$-cost recovery because

$$\left(2-\frac{1}{m}\right)\cdot\max\{\mu(S),p_{\max}(S)\}\geq\mu(S)+\left(1-\frac{1}{m}\right)p_{\max}(S)\geq C(S),$$

where the last inequality follows from Fact 4. □

Finally, we prove that the cost shares fulfill (H_n+1)-summability.

Lemma 10 ξ^{BS} *is* (H_n+1)*-summable.*

PROOF. Let σ be an arbitrary order on the jobs in U, and let $S:=\{i_1,\ldots,i_{|S|}\}\subseteq U$ be a subset of U ordered according to σ. First, observe that

$$\begin{aligned}\sum_{j=1}^{|S|}\xi^{\text{BS}}(i_j,S_j)&=\sum_{j=1}^{|S|}\left(\frac{p_{i_j}}{m}+\int\limits_{\mu(S)}^{p_{i_j}}\frac{1}{|\{k\in S_j:p_k\geq t\}|}\,\mathrm{d}t\right)\\&\leq\sum_{j=1}^{|S|}\left(\frac{p_{i_j}}{m}+\int\limits_{0}^{p_{i_j}}\frac{1}{|\{k\in S_j:p_k\geq t\}|}\,\mathrm{d}t\right)\\&\leq\mu(S)+\sum_{j=1}^{|S|}\int\limits_{0}^{p_{i_j}}\frac{1}{|\{k\in S_j:p_k\geq t\}|}\,\mathrm{d}t.\end{aligned}$$

Fix a point in time $t\in[0,p_{\max}(S)]$. Define $r(t)$ as the number of jobs in S whose processing time is at least t. Using this definition, we obtain

$$\sum_{j=1}^{|S|}\int\limits_{0}^{p_{i_j}}\frac{1}{|\{k\in S_j:p_k\geq t\}|}\,\mathrm{d}t=\int\limits_{0}^{p_{\max}(S)}\sum_{r=1}^{r(t)}\frac{1}{r}\,\mathrm{d}t=\int\limits_{0}^{p_{\max}(S)}H_{r(t)}\,\mathrm{d}t\leq p_{\max}(S)\cdot H_{|S|}.$$

Thus,

$$\sum_{j=1}^{|S|}\xi^{\text{BS}}(i_j,S_j)\leq\mu(S)+p_{\max}(S)\cdot H_{|S|}\leq(H_n+1)\cdot C(S).$$ □

Using Theorem 1, we can prove that Lemma 10 is essentially tight.

Corollary 11 *Let* ξ *be a cost sharing method for the minimum makespan scheduling problem* $P|p_i=1|C_{\max}$ *that satisfies the* β*-cost recovery condition. Then the summability of* ξ *is no better than* H_n/β*.*

PROOF. Consider an instance that consists of n jobs with unit processing times and $m := n$ machines. Clearly, $C(S) = 1 = C(U)$ for all $S \subseteq U$. Theorem 1 now gives a lower bound of H_n/β. $\square$

5 A General Lower Bound on Budget Balance

While we have given cross-monotonic $(2-1/m)$-budget balanced and (H_n+1)-summable cost shares for the minimum makespan scheduling problem, we identify a class of problems to which no constantly budget balanced and cross-monotonic cost sharing method exists in this section. We show that both weighted completion time scheduling and average completion time scheduling belong to this class, as well as all of their generalizations.

Consider a cost sharing game on a universe U of n players whose cost function $C : 2^U \to \mathbb{R}$ is non-decreasing, i.e. $C(S') \leq C(S)$ for all $S' \subseteq S \subseteq U$. If there is an instance to the cost sharing game for which $C(U)$ exceeds $\sum_{i \in U} C(\{i\})$ by a factor of $f(n)$, then the β-core of this game is empty for all $\beta < f(n)$. This is due to the fact that players can never be charged more than the cost they incur when being served alone, and therefore the players in a set S cannot pay more than $\sum_{i \in S} C(\{i\})$.

In the case of general (not necessarily non-decreasing) cost functions, using sets T_i containing i instead of the singletons $\{i\}$ itself can yield even better lower bounds. Intuitively, we choose the subset $T_i \subseteq S$ for which the amount that player $i \in S$ is allowed to pay is smallest.

Theorem 12 *Consider a cost sharing game on a universe U of n players and its cost function C. Let $f : \mathbb{N} \to \mathbb{R}$ be a non-decreasing function. Suppose there is a set S of size $|S| \geq |U|/\gamma$ for some constant $\gamma \geq 1$, and arbitrary sets $T_i \subseteq S$ with $i \in T_i$ such that*

$$C(S) \geq f(|S|) \cdot \sum_{i \in S} C(T_i).$$

Then, there is no cost sharing method ξ in the β-core for any $\beta < f(\frac{n}{\gamma})$ for this game.

PROOF. Assume that ξ is a cost sharing method in the β-core for this problem. First, the core property implies that the cost share of player i in the set $S \supseteq T_i$ is at most the cost induced by the set T_i, i.e. $\xi(i, S) \leq \sum_{j \in T_i} \xi(j, S) \leq C(T_i)$ for all $i \in S$. Second, we assume $C(S) \geq f(|S|) \cdot \sum_{i \in S} C(T_i)$. The condition of β-cost recovery now implies that for every $S \subseteq U$:

$$\beta \geq \frac{C(S)}{\sum_{i \in S} \xi(i, S)} \geq \frac{f(|S|) \cdot \sum_{i \in S} C(T_i)}{\sum_{i \in S} C(T_i)} \geq f\Big(\frac{n}{\gamma}\Big). \qquad \square$$

Since every *cross-monotonic* β-budget balanced cost sharing method is in the β-core, this theorem implies the same lower bound on the budget balance factor of a cross-monotonic cost sharing method for the respective problem. In the following two sections, we apply Theorem 12 to the parallel machine scheduling problems with completion time and average completion time objectives.

5.1 Minimum Weighted Completion Time Scheduling

In the *minimum weighted completion time scheduling problem*, we are given a set of n jobs N and m identical machines. Each job $i \in N$ has a processing time p_i and a weight w_i. The objective is to assign all n jobs to the m machines such that the total weighted completion time $\sum_{i \in N} w_i C_i$ is minimized.

In the cost sharing context, we define $U := N$ as before, and let C be the total weighted completion time of an optimal schedule. We show that the β-core of this scheduling problem is empty for $\beta < (n+1)/2$, even for the unweighted single machine case with unit processing times.

Corollary 13 *Consider the single machine minimum completion time scheduling problem with unit processing times $1|p_i = 1|\sum_i C_i$. There is no cost sharing method ξ that is in the β-core for any $\beta < (n+1)/2$ for this game.*

PROOF. Clearly, the cost of every singleton set $\{i\}$, $i \in U$, is $C(\{i\}) = 1$. Set $T_i := \{i\}$. On the other hand, $C(U) = n(n+1)/2$. Thus,

$$C(U) \geq \frac{n+1}{2} \cdot \sum_{i \in U} C(T_i),$$

and using Theorem 12 with $S = U$ and $f(n) = (n+1)/2$ yields the claim. $\square$

This lower bound carries over to all generalizations of the single machine minimum completion time scheduling problem, e.g. to the *minimum weighted flow time scheduling problem* and problems with additional constraints such as release or due dates. Note that the trivial cost sharing method $\xi^{\text{WCT}}(i, S) :=$

$w_i p_i$ for all $i \in S$ and $S \subseteq U$ is cross-monotonic and n-budget balanced for $P| \,| \sum_i w_i C_i$, as shown by the following lemma.

Lemma 14 *Consider the minimum weighted completion time scheduling problem* $P| \,| \sum_i w_i C_i$. ξ^{WCT} *is a cross-monotonic n-budget balanced cost sharing method for this problem.*

PROOF. ξ^{WCT} is obviously cross-monotonic. It is competitive since $p_i \leq C_i$ for every job $i \in S$ and all $S \subseteq U$.

To show n-cost recovery, we first consider the single machine case. Take an optimal schedule and number the jobs accordingly. Smith [36] proved that if job i is scheduled before job j in an optimal schedule, i.e. $i < j$, then $p_i/w_i \leq p_j/w_j$. Thus, either $p_i \leq p_j$ or $w_i > w_j$ (or both) are true, and the following inequality holds for all $i < j$:

$$w_j p_i \leq \max\{w_i p_i, w_j p_j\} \leq w_i p_i + w_j p_j.$$

Using this, we can bound the cost of an optimal schedule for a set $S \subseteq U$ by

$$\begin{aligned}\sum_{j=1}^{|S|} w_j C_j &= \sum_{j=1}^{|S|} w_j \cdot \Big(\sum_{i=1}^{j} p_i\Big) = \sum_{j=1}^{|S|}\sum_{i=1}^{j} w_j p_i \leq \sum_{j=1}^{|S|} \Big(\sum_{i=1}^{j-1}(w_i p_i + w_j p_j) + w_j p_j\Big) \\ &= \sum_{j=1}^{|S|}\Big(j \cdot w_j p_j + (|S| - j) w_j p_j\Big) = |S| \cdot \sum_{j=1}^{|S|} w_j p_j \leq n \cdot \sum_{j=1}^{|S|} \xi^{\text{WCT}}(j, S),\end{aligned}$$

which proves n-cost recovery for the single machine case.

For the general case, consider the set $S_k \subseteq S$ of jobs that are scheduled on machine $k \in [m]$ in an optimal schedule. Clearly, the schedule for machine k is optimal for the corresponding single machine problem on the set of jobs S_k, for which the above inequality holds. Summing up over all machines, we obtain

$$\sum_{j \in S} w_j C_j = \sum_{k \in [m]} \sum_{j \in S_k} w_j C_j \leq \sum_{k \in [m]} n \cdot \sum_{j \in S_k} \xi^{\text{WCT}}(j, S_k) = n \cdot \sum_{j \in S} \xi^{\text{WCT}}(j, S). \qquad \square$$

5.2 Minimum Average Completion Time Scheduling

In the *minimum average completion time scheduling problem*, the setting is as above, but with the objective of minimizing the total *average* weighted completion time, i.e. $C(S) = \sum_{i \in S} w_i C_i / |S|$ for all $S \subseteq U$. In classical machine scheduling, where an optimal (or approximate) solution for the whole set U of players is sought, the problems with average weighted completion time and weighted completion time objectives coincide, since the objectives only differ by a constant factor of $|U|$. However, during the run of a Moulin mechanism, the size of the current player set varies, and thus $|S|$ can no more be seen as a constant. As a matter of fact, due to the division by $|S|$, the cost function is not monotone for this game, as the following example shows:

Example 15 *Consider an instance on a single machine and three jobs* $\{1, 2, 3\}$ *with processing times* $p_i = i$ *and unit weights. The average completion time is* 3 *if only job* 3 *is scheduled,* $(1 + 4)/2 = 2.5$ *if jobs* 1 *and* 3 *are scheduled, and* $(1+3+6)/3 > 3$ *if all three jobs are scheduled. Hence, the cost of an optimal schedule can increase as well as decrease when a job is added to the scheduled set.*

For this reason, we need a slightly more elaborated instance with non-uniform processing times to show that the β-core of this game is empty for $\beta < (n + 4)/8$. Nevertheless, the lower bound holds even for the unweighted single machine case.

Corollary 16 *Consider the single machine minimum average completion time scheduling problem* $1| \mid \sum_i C_i/n$. *There is no cost sharing method* ξ *that is in the* β*-core for any* $\beta < (n + 4)/8$ *for this game.*

PROOF. Let $U = S \dot\cup L$ be a set of n jobs, where $|S| = n/2 - 1$ and $|L| = n/2 + 1$; we call the jobs in S *small* and those in L *large*. Define $p_i := \epsilon$ for all $i \in S$, and $p_i := 1$ for all $i \in L$. The optimal cost for every singleton set $\{i\}$, $i \in S$, is $C(\{i\}) = \epsilon$. Set $T_i := \{i\}$ for all small jobs $i \in S$. For the large jobs $i \in L$, set $T_i := S \cup \{i\}$. In an optimal schedule for T_i, first all small jobs in S are processed and finally the large job i. The cost of an optimal schedule is thus

$$C(T_i) = \frac{1}{|T_i|}\left(\sum_{j=1}^{|S|} j\cdot\epsilon + (|S|\cdot\epsilon + 1)\right) \le \frac{2}{n}\left(1 + \epsilon n\left(\frac{n}{8} + \frac{1}{2}\right)\right).$$

We obtain

$$\begin{aligned}\sum_{i\in U} C(T_i) &\le \left(\frac{n}{2} - 1\right)\cdot\epsilon + \left(\frac{n}{2} + 1\right)\left(\frac{2}{n}\left(1 + \epsilon n\left(\frac{n}{8} + \frac{1}{2}\right)\right)\right)\\ &\le \frac{n+2}{n}\left(1 + \epsilon n\left(\frac{n}{8} + 1\right)\right). \qquad (5)\end{aligned}$$

Define $\epsilon' := \epsilon n(n/8 + 1)$. On the other hand,

$$C(U) \ge \frac{(\frac{n}{2} + 1)(\frac{n}{2} + 2)}{2n} = \frac{(n+2)(n+4)}{8n}. \qquad (6)$$

Combining inequalities (5) and (6), we obtain

$$C(U) \geq \frac{n+4}{8(1+\epsilon')} \cdot \sum_{i \in U} C(T_i).$$

By Theorem 12, we obtain a lower bound of $\beta \geq (n+4)/(8(1+\epsilon'))$ for any cost sharing method in the β-core. The claim now follows by choosing ϵ sufficiently small. □

Note that the trivial cost sharing method $\xi^{\text{ACT}}(i, S) := w_i p_i/n$ for all $i \in S$ and $S \subseteq U$ is cross-monotonic and n-budget balanced for $P| \,|\sum_i w_i C_i/n$.

Lemma 17 *Consider the minimum average weighted completion time scheduling problem* $P| \,|\sum_i w_i C_i/n$. ξ^{ACT} *is a cross-monotonic* n*-budget balanced cost sharing method for this problem.*

PROOF. ξ^{ACT} is obviously cross-monotonic. It is competitive since $w_i p_i/n \leq w_i C_i/|S|$ for every job $i \in S$ and all $S \subseteq U$. The proof of n-cost recovery is analogous to the non-average case.

On a single machine, Smith's rule still holds for every optimal schedule for S since the average cost is only a constant factor times the non-average cost for fixed S. Thus, the cost of an optimal schedule for a set $S \subseteq U$ is bounded by

$$\sum_{j=1}^{|S|} w_j C_j/|S| \leq \sum_{j=1}^{|S|} w_j p_j = n \cdot \sum_{j=1}^{|S|} \xi^{\text{ACT}}(j, S).$$

For the general case, again, considering the sets $S_k \subseteq S$ for all machines $k \in [m]$ and summing up yields

$$\sum_{j \in S} w_j C_j/|S| \leq \sum_{k \in [m]} \sum_{j \in S_k} w_j C_j/|S_k| \leq \sum_{k \in [m]} n \cdot \sum_{j \in S_k} \xi^{\text{ACT}}(j, S_k) = n \cdot \sum_{j \in S} \xi^{\text{ACT}}(j, S),$$

proving n-cost recovery. □

6 Conclusion

We proved that the efficiency of Moulin mechanisms is not approximable within less than logarithmic factors even with the new social cost efficiency measure. Our lower bound holds if the underlying optimization problem satisfies a certain cost-stability property. This reduces the hope to find truly efficient cost sharing mechanisms for these problems. On the other hand, the new efficiency measure allows us to characterize cost sharing mechanisms in terms of their best polylogarithmic approximation factor.

Although most of the previously known cross-monotonic and approximately budget balanced cost sharing methods for combinatorial optimization problems turned out to simultaneously achieve the best possible social cost efficiency [3,17–19], our work reveals that different cost sharing methods achieving the same budget balance factor may indeed behave very differently with respect to approximate social cost.

We studied cost sharing methods for makespan and completion time related scheduling problems. Our results demonstrate that the tractability of these problems in a cost sharing context heavily depends on the respective objective function. Our negative result on the budget balance factor for cross-monotonic cost sharing methods motivates the investigation of alternative cost sharing models; perhaps with a weaker notion of truthfulness for cooperative games.

References

[1] J. Green, E. Kohlberg, J. J. Laffont, Partial equilibrium approach to the free rider problem, Journal of Public Economics 6 (1976) 375–394.

[2] K. Roberts, The characterization of implementable choice rules, in: J. J. Laffont (Ed.), Aggregation and Revelation of Preferences, North-Holland, 1979.

[3] T. Roughgarden, M. Sundararajan, New trade-offs in cost-sharing mechanisms, in: Proceedings of the 38th ACM Symposium on Theory of Computing, 2006, pp. 79–88.

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

[5] N. Immorlica, M. Mahdian, V. S. Mirrokni, Limitations of cross-monotonic cost sharing schemes, in: Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms, 2005, pp. 602–611.

[6] Y. Bleischwitz, B. Monien, Fair cost-sharing methods for scheduling jobs on parallel machines, in: Proceedings of the 6th International Conference on Algorithms and Complexity, Vol. 3998 of Lecture Notes in Computer Science, Springer, 2006, pp. 175–186.

[7] A. Archer, J. Feigenbaum, A. Krishnamurthy, R. Sami, S. Shenker, Approximation and collusion in multicast cost sharing, Games and Economic Behavior 47 (1) (2004) 36–71.

[8] J. Feigenbaum, A. Krishnamurthy, R. Sami, S. Shenker, Hardness results for multicast cost-sharing, Theoretical Computer Science 304 (2003) 215–236.

[9] J. Feigenbaum, C. Papadimitriou, S. Shenker, Sharing the cost of multicast transmissions, Journal of Computer and System Sciences 63 (1) (2001) 21–41, special issue on internet algorithms.

[10] K. Jain, V. Vazirani, Applications of approximation algorithms to cooperative games, in: Proceedings of the 33rd ACM Symposium on Theory of Computing, 2001, pp. 364–372.

[11] K. Kent, D. Skorin-Kapov, Population monotonic cost allocations on MSTs, in: Proceedings of the 6th International Conference on Operational Research, Croatian Operational Research Society, Zagreb, 1996, pp. 43–48.

[12] M. Pál, E. Tardos, Group strategyproof mechanisms via primal-dual algorithms, in: Proceedings of the 44th Symposium on the Foundations of Computer Science, 2003, pp. 584–593.

[13] S. Leonardi, G. Schäfer, Cross-monotonic cost sharing methods for connected facility location games, Theoretical Computer Science 326 (1–3) (2004) 431–442.

[14] A. Gupta, A. Srinivasan, É. Tardos, Cost-sharing mechanisms for network design, in: Proceedings of the 7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, 2004.

[15] J. Könemann, S. Leonardi, G. Schäfer, A group-strategyproof mechanism for Steiner forests, in: Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms, 2005, pp. 612–619.

[16] J. Könemann, S. Leonardi, G. Schäfer, S. van Zwam, From primal-dual to cost shares and back: a stronger LP relaxation for the Steiner forest problem, in: Proceedings of the 32nd International Colloquium on Automata, Languages and Programming, Vol. 3580 of Lecture Notes in Computer Science, Springer, 2005, pp. 930–942.

[17] A. Gupta, J. Könemann, S. Leonardi, R. Ravi, G. Schäfer, An efficient cost-sharing mechanism for the prize-collecting Steiner forest problem, in: Proceedings of the 18th ACM-SIAM Symposium on Discrete Algorithms, 2007, pp. 1153–1162.

[18] T. Roughgarden, M. Sundararajan, Optimal efficiency guarantees for network design mechanisms, in: Proceedings of the 12th Conference on Integer Programming and Combinatorial Optimization, 2007, pp. 469–483.

[19] S. Chawla, T. Roughgarden, M. Sundararajan, Optimal cost-sharing mechanisms for Steiner forest problems, in: Proceedings of the 2nd International Workshop on Internet and Network Economics, Springer, 2006, pp. 112–123.

[20] M. R. Garey, D. S. Johnson, Strong NP-completeness results: Motivation, examples and implications, Journal of the ACM 25 (3) (1978) 499–508.

[21] J. K. Lenstra, D. B. Shmoys, É. Tardos, Approximation algorithms for scheduling unrelated parallel machines, in: Proceedings of the 28th Symposium on the Foundations of Computer Science, 1987, pp. 217–224.

[22] P. Brucker, Scheduling Algorithms, Springer, New York, USA, 1998.

[23] F. Afrati, E. Bampis, C. Chekuri, D. Karger, C. Kenyon, S. Khanna, I. Milis, M. Queyranne, M. Skutella, C. Stein, M. Sviridenko, Approximation schemes for minimizing average weighted completion time with release dates, in: Proceedings of the 40th Symposium on the Foundations of Computer Science, 1999, pp. 32–43.

[24] B. Simons, Multiprocessor scheduling of unit-time jobs with arbitrary release times and deadlines, SIAM Journal on Computing 12 (2) (1983) 294–299.

[25] R. McNaughton, Scheduling with deadlines and loss functions, Management Sciences 6 (1959) 1–12.

[26] N. Nisan, A. Ronen, Algorithmic mechanism design, Games and Economic Behavior (2001) 166–196.

[27] A. Archer, É. Tardos, Truthful mechanisms for one-parameter agents, in: Proceedings of the 42nd Symposium on the Foundations of Computer Science, 2001, pp. 482–491.

[28] A. Kovacs, Fast monotone 3-approximation algorithm for scheduling related machines, in: Proceedings of the 13th Annual European Symposium on Algorithms, Lecture Notes in Computer Science, Springer, 2005.

[29] R. Porter, Mechanism design for online real-time scheduling, in: Proceedings of the 5th ACM Conference on Electronic Commerce, 2004.

[30] B. Heydenreich, R. Müller, M. Uetz, Decentralization and mechanism design for online machine scheduling, in: Proceedings of the 1st International Workshop on Computational Social Choice, Springer, 2006, pp. 136–147.

[31] A. Mehta, T. Roughgarden, M. Sundararajan, Beyond Moulin mechanisms, in: Proceedings of the 8th ACM Conference on Electronic Commerce, 2007.

[32] N. Devanur, M. Mihail, V. Vazirani, Strategyproof cost-sharing mechanisms for set cover and facility location games, in: Proceedings of the 4th ACM Conference on Electronic Commerce, 2003.

[33] J. Brenner, G. Schäfer, Singleton acyclic mechanisms and their applications to scheduling problems, Tech. Rep. 42-2007, Institute of Mathematics, Technical University Berlin.

[34] R. Graham, E. Lawler, J. Lenstra, A. R. Kan, Optimization and approximation in deterministic sequencing and scheduling: a survey, Annals of Discrete Mathematics 5 (1979) 287–326.

[35] D. Hochbaum (Ed.), Approximation Algorithms for NP-hard Problems, PWS Publishing Company, 1997.

[36] W. Smith, Various optimizers for single-stage production, in: Naval Research Logistics Quarterly, Vol. 3, 1956, pp. 59–66.