Cost Sharing Methods for Makespan and Completion Time Scheduling*

Janina Brenner and Guido Schäfer

Institute of Mathematics, Technical University Berlin, Germany
{brenner,schaefer}@math.tu-berlin.de

Abstract. Roughgarden and Sundararajan recently introduced an alternative measure of efficiency for cost sharing mechanisms. We study cost sharing methods for combinatorial optimization problems using this novel efficiency measure, with a particular focus on scheduling problems. While we prove a lower bound of $\Omega(\log n)$ for a very general class of problems, we give a best possible cost sharing method for minimum makespan scheduling. Finally, we show that no budget balanced cost sharing methods for completion or flow time objectives exist.

Keywords: game theory, mechanism design, cost sharing mechanisms, combinatorial optimization, scheduling problems.

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 that act strategically in order to receive this service. In a scheduling context, we can imagine jobs to be owned by agents wishing their jobs to be executed on a machine. Besides finding a way of providing the service, the problem is then to distribute the resulting cost among the users in a fair manner. Meanwhile, the service provider may have to 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 non-decreasing 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

* This work was supported by the DFG Research Center Matheon "Mathematics for key technologies".

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: (a) a player is not charged more than her bid, (b) a player is charged only if she receives service, and (c) 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 solely 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 *β-budget balanced* if 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.,

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

If $\beta = 1$, we simply call the cost sharing mechanism budget balanced.

A mechanism 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.* 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 [8, 24] state that budget balance and efficiency cannot be achieved 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 [25] introduced an alternative efficiency measure that attempts to circumvent the 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 based on a framework due to Moulin and Shenker [20]. 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. [13] prove that every group-strategyproof cost sharing mechanism (satisfying some natural conditions) corresponds to a cross-monotonic cost sharing method.

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 [25]. Our contribution is threefold:

1. Lower Bound on Approximability of Cost Sharing Methods. We present a general inapproximability result for cost sharing methods of combinatorial optimization problems. In particular, we prove that there is no cost sharing method that is α-summable and satisfies cost recovery for any $\alpha < \log n$, where n denotes the number of players. Our proof holds if the underlying cost function satisfies a certain "stability" property.

As a consequence, our result implies a lower bound of $\log n$ on the approximability of cost sharing 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. An 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 that have to be executed on m parallel machines. The goal is to assign all jobs to the machines such that the maximum completion time is minimized. We assume that jobs act strategically and attempt to get processed at a low cost. We develop a cross-monotonic cost sharing method for this problem that is $(2 - 1/m)$-budget balanced and $\log n$-approximate; this is tight with respect to both budget balance and approximability.

3. Budget Balance of Cost Sharing Methods for other Scheduling Problems. There are several other scheduling problems that can be considered in a cost sharing context. We show that for scheduling problems in which we aim at minimizing the total (weighted) completion (or flow) time, there is no cross-monotonic cost sharing method that is β-budget balanced for any $\beta < n/2$.

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 [20] has been applied to game-theoretic variants of classical optimization problems such as fixed multicast [1, 5, 6], submodular cost sharing [20], Steiner trees [14, 15], facility location, single-source rent-or-buy network design [22, 19, 10] and Steiner forests [16]. Lower bounds on the budget balance factor that is achievable by a cross-monotonic cost sharing mechanism are given in [13, 17]. Very recently, researchers started to investigate cost sharing mechanisms in light of the novel efficiency measure of Roughgarden and Sundararajan; see [9, 25, 26, 4].

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 [21, 2, 18], or strategyproofness (but not group-strategyproofness) is an issue [23, 11].

Related to our work is the recent work of Bleischwitz and Monien [3]. The authors present a cross-monotonic cost sharing method for the minimum makespan scheduling problem. However, as we argue below, their cost sharing mechanism does not approximate social cost.

2 Preliminaries

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

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

We say that ξ satisfies *β-cost recovery* if the first inequality holds; it is *competitive* if the latter inequality is fulfilled.

Moulin and Shenker [20] 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 be the final player set and define the payments as $x_i(S) := \xi(i, S)$ for all $i \in S$. Jain and Vazirani [14] later observed that the result of Moulin and Shenker also holds if one considers 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

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

Social Welfare vs. Social Cost. 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., $\mathit{\Pi}(S^M) \leq \alpha \cdot \mathit{\Pi}(S)$ for all $S \subseteq U$. Since $u(U) - \mathit{\Pi}(S) = u(S) - C(S)$, the traditional definition of efficiency, and $u(U)$ is a constant, a set S has minimum social cost iff it has maximum efficiency.

Roughgarden and Sundararajan [25] 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., $S = \{i_1, \ldots, i_{|S|}\}$, 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 ordering σ and every subset $S \subseteq U$:

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

Roughgarden and Sundararajan [25] 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, the authors argue that $\max\{\alpha, \beta\}$ is a lower bound on the approximability of $M(\xi)$.

In this paper, we use $[n]$ to denote the set $\{1, \ldots, n\}$. Moreover, 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, $H_n \approx \log n$, and we use both values interchangeably.

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. Our lower bound holds if the underlying cost function C satisfies a certain "stability" property, which is fulfilled by a variety of combinatorial optimization problems such as facility location, Steiner tree, parallel machine scheduling, etc. Together with the recent result of Roughgarden and Sundararajan [25], this shows that for several problems, the approximability of Moulin mechanisms cannot be better than $\Omega(\log n)$.

Theorem 1. *Let ξ be a cost sharing method on a universe U that satisfies the β-cost recovery condition with respect to a cost function C. Suppose that there is a set $S \subseteq U$ with $|S| \geq |U|/\gamma$ for some constant $\gamma \geq 1$ such that $C(S') \geq C(S)/\delta$ for all $S' \subseteq S$ and some constant $\delta \geq 1$. Then ξ is not α-summable for any $\alpha < H_{\lceil n/\gamma \rceil}/(\beta \cdot \delta)$, where n is the number of players in U.*

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 users 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 user $i \in S_j$ 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 1 (Fixed-tree Multicast Problem). Users 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 users 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 users are located on the same vertex which is connected to the root by an edge of length 1. There are better lower bounds for this problem.

Example 2 (Facility Location Problem). Users 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 users are located directly on v. Then, the cost of a solution is independent of the number of users and equal to the opening cost of the facility. This lower bound is tight, as has been shown in [26].

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.

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. [7], 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., [12]).

Fact 1 *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 [3] 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, we define the cost share of $i \in S$ as:

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

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 [3] in the following theorem.

Theorem 2. *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 2 proves that the Moulin mechanism $M(\xi^{\text{BM}})$, 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 $M(\xi^{\text{BM}})$ can be as large as $n/2$ times the optimal social cost, where n is the number of jobs in the universe U.

Lemma 1. *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 $C(U) = 1 + (m-1)\epsilon$.

[1] At first sight, the cost shares that we state here differ from the ones defined by Bleischwitz and Monien in [3]. However, it can easily be verified that both definitions are in fact equivalent; we feel that the definition we present here is more intuitive.

Observe that the processing time of job i_j, $j \in [m]$, is maximum among all jobs in the set $S_j = \{i_1, \ldots, i_j\}$, 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. □

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 $\log n$-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 and approximate budget balance and summability.

Theorem 3. *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.*

The proof of Theorem 3 follows from Lemmas 2, 3 and 4 that are given below.

Lemma 2. *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. $\square$

We show next that the budget balance condition is satisfied.

Lemma 3. *ξ^{BS} is $(2-1/m)$-budget balanced.*

Proof. It is easy to verify that with the cost share definition in (4) we have

$$\sum_{i \in S} \xi(i, S) = \max\{\mu(S), p_{\max}(S)\}.$$

By Fact 1, $C(S) \geq \max\{\mu(S), p_{\max}(S)\}$, 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 1. $\square$

Finally, we prove that the cost shares fulfill $O(\log n)$-summability.

Lemma 4. *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

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

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

$$\sum_{j=1}^{|S|} \int_0^{p_{i_j}} \frac{1}{|\{k \in S_j : p_k \geq t\}|} \,\mathrm{d}t = \int_0^{p_{\max}(S)} \sum_{r=1}^{r(t)} \frac{1}{r} \,\mathrm{d}t = \int_0^{p_{\max}(S)} H_{r(t)} \,\mathrm{d}t \leq p_{\max}(S) \cdot H_{|S|}.$$

Thus,

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

□

Lemma 4 is tight, as the following corollary shows.

Corollary 1. *Let ξ be a cost sharing method for the minimum makespan scheduling problem 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. □

5 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 $\beta < (n + 1)/2$.

Theorem 4. *Consider the* 1*-machine minimum completion time scheduling problem $1| \,| \sum_i C_i$. There is no cost sharing method ξ that is in the β-core for any $\beta < (n + 1)/2$.*

Proof. Let U be a set of n jobs and define $p_i := 1$ for each $i \in U$. Clearly, the optimal cost for every singleton set $\{i\}$, $i \in U$, is $C(\{i\}) = 1$. The β-core property therefore implies that the cost share of i is at a most 1, i.e., $\xi(i, S) \leq 1$ for all $i \in S$ and for all $S \subseteq U$. On the other hand, $C(S) = |S|(|S| + 1)/2$ for all $S \subseteq U$.

The condition of β-cost recovery now implies that for every $S \subseteq U$

$$\beta \geq \frac{C(S)}{\sum_{i \in S} \xi(i, S)} \geq \frac{\frac{|S|(|S|+1)}{2}}{|S|} = \frac{|S| + 1}{2}.$$

□

Since every β-budget balanced cross-monotonic cost sharing method is in the β-core, this theorem implies the same lower bound for the budget balance factor of *cross-monotonic* cost sharing methods for the 1-machine minimum completion time scheduling problem. Remind that cross-monotonic and n-budget balanced cost sharing methods trivially exist for these problems.

This result also carries over to all scheduling problems that are generalizations of the 1-machine minimum completion time scheduling problem, as e.g. the *minimum weighted flow time scheduling problem*, and problems with additional constraints such as release or due dates.

6 Conclusion

We proved that in many cases, efficiency is not approximable within less than logarithmic factors even with the new approach of social cost. This reduces the hope to find truly efficient cost sharing mechanisms, while on the other hand allowing us to evaluate social cost approximation factors in terms of their highest polylogarithmic power.

We studied cost sharing methods for the two cases of minimum makespan and minimum completion time scheduling. Our results demonstrate that different scheduling problems can behave very differently. While the completion time setting raises the question of how to handle problems for which the here examined framework does not allow for any (reasonable) solutions, there are many more scheduling problems that deserve to be studied.

References

1. A. Archer, J. Feigenbaum, A. Krishnamurthy, R. Sami, and S. Shenker. Approximation and collusion in multicast cost sharing. *Games and Economic Behavior*, 47(1):36–71, 2004.
2. A. Archer and E. Tardos. Truthful mechanisms for one-parameter agents. In *Proc. of the 42nd Annual Sympos. on Foundations of Computer Science*, pages 482–491. IEEE Computer Society, 2001.
3. Y. Bleischwitz and B. Monien. Fair cost-sharing methods for scheduling jobs on parallel machines. In *Proc. of the 6th Int. Conf. on Algorithms and Complexity*, volume 3998 of *Lecture Notes in Comput. Sci.*, pages 175–186, Berlin, 2006. Springer.
4. S. Chawla, T. Roughgarden, and M. Sundararajan. Optimal cost-sharing mechanisms for steiner forest problems. submitted to WINE.
5. J. Feigenbaum, A. Krishnamurthy, R. Sami, and S. Shenker. Hardness results for multicast cost-sharing. *Theoretical Computer Science*, 304:215–236, 2003.
6. J. Feigenbaum, C. Papadimitriou, and S. Shenker. Sharing the cost of multicast transmissions. *J. Comput. System Sci.*, 63(1):21–41, 2001. Special issue on internet algorithms.
7. R. Graham, E. Lawler, J. Lenstra, and A. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*, 5:287–326, 1979.

8. J. Green, E. Kohlberg, and J. J. Laffont. Partial equilibrium approach to the free rider problem. *Journal of Public Economics*, 6:375–394, 1976.
9. A. Gupta, J. Könemann, S. Leonardi, R. Ravi, and G. Schäfer. An efficient cost-sharing mechanism for the prize-collecting steiner forest problem. In *ACM-SIAM Sympos. on Discrete Algorithms*. ACM Press, 2007. to appear.
10. A. Gupta, A. Srinivasan, and É. Tardos. Cost-sharing mechanisms for network design. In *Proc. of the Seventh Int. Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, 2004.
11. B. Heydenreich, R. Müller, and M. Uetz. Decentralization and mechanism design for online machine scheduling. unpublished manuscript.
12. D. Hochbaum, editor. *Approximation Algorithms for NP-hard Problems*. PWS Publishing Company, 1997.
13. N. Immorlica, M. Mahdian, and V. S. Mirrokni. Limitations of cross-monotonic cost sharing schemes. In *Proc. of the Sixteenth Annual ACM-SIAM Sympos. on Discrete Algorithms*, pages 602–611. ACM Press, 2005.
14. K. Jain and V. Vazirani. Applications of approximation algorithms to cooperative games. In *Proc. of the 33rd Annual ACM Sympos. on the Theory of Computing (STOC)*, pages 364–372, 2001.
15. K. Kent and D. Skorin-Kapov. Population monotonic cost allocations on MSTs. In *Proc. of the 6th Int. Conf. on Operational Research (Rovinj, 1996)*, pages 43–48. Croatian Oper. Res. Soc., Zagreb, 1996.
16. J. Könemann, S. Leonardi, and G. Schäfer. A group-strategyproof mechanism for Steiner forests. In *Proc. of the Sixteenth Annual ACM-SIAM Sympos. on Discrete Algorithms*, pages 612–619. ACM Press, 2005.
17. J. Könemann, S. Leonardi, G. Schäfer, and S. van Zwam. From primal-dual to cost shares and back: a stronger LP relaxation for the Steiner forest problem. In *Automata, Languages and Programming*, volume 3580 of *Lecture Notes in Comput. Sci.*, pages 930–942. Springer, Berlin, 2005.
18. A. Kovacs. Fast monotone 3-approximation algorithm for scheduling related machines. In *Proc. of the 13th Annual European Sympos. on Algorithms*, Lecture Notes in Comput. Sci. Springer, 2005.
19. S. Leonardi and G. Schäfer. Cross-monotonic cost sharing methods for connected facility location games. *Theor. Comput. Sci.*, 326(1-3):431–442, 2004.
20. H. Moulin and S. Shenker. Strategyproof sharing of submodular costs: budget balance versus efficiency. *Econom. Theory*, 18(3):511–533, 2001.
21. N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, pages 166–196, 2001.
22. M. Pál and É. Tardos. Group strategyproof mechanisms via primal-dual algorithms. In *Proc. of the 44th Sympos. on the Foundations of Computer Science (FOCS)*, pages 584–593, 2003.
23. R. Porter. Mechanism design for online real-time scheduling. In *Proc. of the ACM Conference on Electronic Commerce*. ACM Press, 2004.
24. K. Roberts. The characterization of implementable choice rules. In J. J. Laffont, editor, *Aggregation and Revelation of Preferences*. North-Holland, 1979.
25. T. Roughgarden and M. Sundararajan. New trade-offs in cost-sharing mechanisms. In *STOC*, 2006.
26. T. Roughgarden and M. Sundararajan. Approximately efficient cost-sharing mechanisms. arXiv report, http://www.arxiv.org/pdf/cs.GT/0606127, June 2006.