To be or not to be (served)*

Yvonne Bleischwitz[1], Burkhard Monien[1], and Florian Schoppmann[1,2]

[1] Faculty of Computer Science, Electrical Engineering and Mathematics, University of Paderborn, Fürstenallee 11, 33102 Paderborn, Germany.
{yvonneb,bm,fschopp}@uni-paderborn.de
[2] International Graduate School of Dynamic Intelligent Systems

Abstract. A common modeling assumption in the realm of cost sharing is that players persuade each other to jointly submit false bids if none of the members of such a coalition loses utility and at least one gains. In order to counteract this kind of manipulation, the service provider could employ group-strategyproof (GSP) mechanisms that elicit truthful bids. The basically only general technique for the design of GSP mechanisms is due to Moulin. Unfortunately, it has limitations with regard to budget-balance (BB) and economic efficiency (EFF).
In this work, we introduce a slight modification of GSP that we call CGSP, allowing us to achieve vastly better results concerning BB and EFF. In particular, we give new CGSP mechanisms that we call "egalitarian" due to being inspired by Dutta and Ray's (1989) "egalitarian solution". We achieve 1-BB for arbitrary costs and additionally $2H_n$-EFF for the very natural and large class of subadditive costs. Egalitarian mechanisms are also acyclic mechanisms, as introduced by Mehta et al. (2007). Thus far, acyclic was known only to imply weak GSP, yet we show that it is already sufficient for the strictly stronger CGSP.
Finally, we present a framework and applications on how to cope with computational complexity.

1 Introduction and Model

1.1 Motivation

We study *cost-sharing problems* where there is a set of players having (binary) demand for some common service, and the task is to determine which subset S of players to serve and how to distribute the incurred cost $C(S)$. We follow the line of studying this problem from the incentive-compatibility angle, where decisions can solely be based on *valuations* that the players report for the service. This problem is fundamental in economics and has a broad area of applications, e.g., sharing the cost of public infrastructure projects, distributing volume discounts, or allocating development costs of low-volume built-to-order products.

In the standard model, a *service provider* takes the role of offering the common service to the n players and hence has to solve the cost-sharing problem.

* This work was partially supported by the IST Program of the European Union under contract number IST-15964 (AEOLUS).

While the valuations reported by the players are binding, they cannot be assumed to be truthful. We refer to them as *bids* in the following. The decision-making of the service provider is governed by a (commonly known) *cost-sharing mechanism* that specifies the set of served players and their respective payments for any combination of bids. The main difficulty lies in achieving *group-strategyproofness*, i.e., designing the mechanism such that players communicate their true valuations out of self-interest, even if they could form coalitions.

Apart from being group-strategyproof, there are many more desirable properties for a cost-sharing mechanism. Most naturally, it has to ensure recovery of the provider's cost as well as competitive prices in that the generated surplus is always relatively small. This constraint is referred to as *budget-balance*. Moreover, there should be a reasonable trade-off between the provider's cost and the valuations of the excluded players, meaning that the mechanism is *economically efficient*. Finally, practical applications demand for polynomial-time computability (in the size of the problem).

The essentially only known general technique for the design of group-strategyproof mechanisms is due to Moulin [15]. Unfortunately, it has severe limitations with respect to the former objectives. The pivotal point of this paper is slightly altering the group-strategyproof requirement to greatly improve performance.

1.2 The Model

Notation. For $n \in \mathbb{N}_0$, let $[n] := \{1, \dots, n\}$ and $[n]_0 := [n] \cup \{0\}$. Given $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{Q}^n$ and $S \subseteq [n]$, let $\boldsymbol{x}_S := (x_i)_{i \in S} \in \mathbb{Q}^{|S|}$ and $\boldsymbol{x}_{-S} := \boldsymbol{x}_{[n] \setminus S}$. Let $(\boldsymbol{x}_{-S}, \boldsymbol{y}_S) \in \mathbb{Q}^n$ denote the vector where the components in $\boldsymbol{x}$ for S are replaced by the respective ones from $\boldsymbol{y}$. For $k \in [|S|]$, we define $MIN_k\, S$ as the set of k smallest elements in S. Let $H_n = \sum_{i=1}^{n} \frac{1}{i}$ be the n-th harmonic number. By convention, the vector of the players' true valuations is always $\boldsymbol{v} \in \mathbb{Q}^n$, whereas an actual bid vector is denoted $\boldsymbol{b} \in \mathbb{Q}^n$.

A *cost-sharing problem* is specified by a *cost function* $C : 2^{[n]} \to \mathbb{Q}_{\geq 0}$ mapping each subset of the $n \in \mathbb{N}$ players to the cost of serving them. In the following, we first focus on incentive-compatibility. Afterwards, we relate cost-sharing mechanisms to the cost of serving the selected players.

Definition 1. *A* cost-sharing mechanism $M = (Q \times x) : \mathbb{Q}^n \to 2^{[n]} \times \mathbb{Q}^n$ *is a function where* $Q(\boldsymbol{b}) \in 2^{[n]}$ *is the set of players to be served and* $x(\boldsymbol{b}) \in \mathbb{Q}^n$ *is the vector of cost shares.*

All mechanisms are required to fulfill three standard properties. For all $\boldsymbol{b} \in \mathbb{Q}^n$:

- *No positive transfers* (NPT): Players never get paid, i.e., $x_i(\boldsymbol{b}) \geq 0$.
- *Voluntary participation* (VP): Players never pay more than they bid and are only charged when served, i.e., if $i \in Q(\boldsymbol{b})$ then $x_i(\boldsymbol{b}) \leq b_i$, else $x_i(\boldsymbol{b}) = 0$.
- *Consumer sovereignty* (CS): For any player $i \in [n]$ there is a threshold bid $b_i^+ \in \mathbb{Q}_{\geq 0}$ such that i is served if bidding at least b_i^+, regardless of the other players' bids; i.e., there is a $b_i^+ \in \mathbb{Q}_{\geq 0}$ such that if $b_i \geq b_i^+$ then $i \in Q(\boldsymbol{b})$.

Note that VP and NPT imply that players may opt to not participate (by submitting a negative bid). Together with CS this is referred to as *strict* CS.

We assume players' utilities $u_i : \mathbb{Q}^n \to \mathbb{Q}$ to be *quasi-linear*, i.e., $u_i(\boldsymbol{b}) := v_i - x_i(\boldsymbol{b})$ if $i \in Q(\boldsymbol{b})$ and 0 otherwise. Under this premise and with *rational* players, mechanisms should elicit truthful bids ($\boldsymbol{b} = \boldsymbol{v}$) even if collusion is feasible:

Definition 2. *A mechanism is* group-strategyproof *(GSP) if for all true valuation vectors $\boldsymbol{v} \in \mathbb{Q}^n$ and coalitions $K \subseteq [n]$ there is no bid vector $\boldsymbol{b} \in \mathbb{Q}^n$ with $\boldsymbol{b}_{-K} = \boldsymbol{v}_{-K}$ such that $u_i(\boldsymbol{b}) \geq u_i(\boldsymbol{v})$ for all $i \in K$ and $u_i(\boldsymbol{b}) > u_i(\boldsymbol{v})$ for at least one $i \in K$.*

A mechanism is *weakly GSP* (WGSP) if the inequalities in Definition 2 are required to be strict for all $i \in K$.

Cost shares selected by a GSP mechanism only depend on the set of served players and not on the bids [15]. This gives rise to the following definition:

Definition 3. *A* cost-sharing method *is a function $\xi : 2^{[n]} \to \mathbb{Q}^n_{\geq 0}$ that maps each set of players to a vector of cost shares.*

Clearly, every GSP mechanism $(Q \times x)$ *induces* a unique cost-sharing method ξ, by setting $\xi(S) := x(\boldsymbol{b})$ where $b_i < 0$ if $i \notin S$ and $b_i = b_i^+$ if $i \in S$. A cost-sharing method ξ is *cross-monotonic* if $\xi_i(S \cup T) \leq \xi_i(S)$ for all $S, T \subseteq [n]$ and all $i \in [n]$. In his seminal work [15], Moulin gave the straightforward mechanism $Moulin_\xi$ that is GSP for any cross-monotonic ξ. $Moulin_\xi$ repeatedly rejects players whose bids are below their current cost share until all remaining players can afford their payments. For any GSP mechanism M with cross-monotonic cost shares ξ, $Moulin_\xi$ produces the same utility for each player as M. Thus, we call any GSP mechanism with cross-monotonic cost shares a *Moulin mechanism*.

Avoiding coalitional cheating alone is clearly not sufficient, as it does not yet relate to the cost of serving the selected players. Typically, costs stem from solutions to a combinatorial optimization problem and are defined only implicitly. In this work, $C(S)$ is the value of a minimum-cost solution for the instance induced by the player set $S \subseteq [n]$. There are two major obstacles to recover this cost exactly: First, computing the *optimal cost* $C(S)$ may take exponential time, and the service provider therefore resorts to an approximate solution with *actual* cost $C'(S) \geq C(S)$. Second, already the GSP requirement places restrictions on the possible cost-shares. Nonetheless, the total charge of a mechanism (and, analogously, of a cost-sharing method) should be reasonably bounded.

Definition 4. *A mechanism $M = (Q \times x)$ is* β-budget-balanced *(β-BB, for $\beta \geq 1$) if for all $\boldsymbol{b} \in \mathbb{Q}^n : C'(Q(\boldsymbol{b})) \leq \sum_{i \in Q(\boldsymbol{b})} x_i(\boldsymbol{b}) \leq \beta \cdot C(Q(\boldsymbol{b}))$.*

As a quality measure for the choice of a set of served players, we use *optimal* and *actual social costs* $SC_{\boldsymbol{v}}, SC'_{\boldsymbol{v}} : 2^{[n]} \to \mathbb{Q}_{\geq 0}$, respectively, where $SC_{\boldsymbol{v}}(S) := C(S) + \sum_{i \in [n] \setminus S} \max\{0, v_i\}$ and $SC'_{\boldsymbol{v}}(S) := C'(S) + \sum_{i \in [n] \setminus S} \max\{0, v_i\}$. The cost incurred by the served players and the valuations of the rejected players should be traded off as good as possible:

Definition 5. *A mechanism $M = (Q \times x)$ is* γ-efficient *(γ-EFF, for $\gamma \geq 1$) if for all true valuations $\boldsymbol{v} \in \mathbb{Q}^n : SC'_{\boldsymbol{v}}(Q(\boldsymbol{v})) \leq \gamma \cdot \min_{T \subseteq [n]}\{SC_{\boldsymbol{v}}(T)\}$.*

Often, costs exhibit a special structure that can be exploited. In this work, we focus on *subadditive costs* where for all $S, T \subseteq [n] : C(S) + C(T) \geq C(S \cup T)$.

1.3 Related Work

The essentially only general technique for the design of GSP mechanisms consists of finding a cross-monotonic cost-sharing method ξ for use with Moulin_ξ [15]. Yet, already achieving good BB is challenging for many problems [13].

The notion of social cost is due to Roughgarden and Sundararajan [18], who initiated a sequence of works in which Moulin mechanisms with not just good BB but also reasonable EFF were given [19,3,5,11]. However, cross-monotonic cost shares have limitations that hence no Moulin mechanism can overcome [13,3,18]. Thus, there is great need for alternatives. Acyclic mechanisms, as introduced by Mehta et al. [14], are one alternative framework that performs better with respect to BB and EFF. Yet, acyclic mechanisms are not necessarily GSP but only known to be WGSP. Non-Moulin GSP mechanisms were given by Bleischwitz et al. [2]; yet only for a limited scenario with symmetric costs.

Besides optimizing BB and EFF, several works have put efforts into characterization results that help understanding the fundamentals of cost sharing [15,13,17]. On the negative side, long-standing results [10] imply that, in general, mechanisms cannot fulfill GSP, 1-BB, and 1-EFF all at once. In fact, already for symmetric cost functions, there are in general no GSP, 1-BB mechanisms [2].

For previous results on the makespan cost-sharing problem (costs stem from a schedule for the selected players' jobs) see Table 1.

1.4 Contribution

We introduce the new behavioral assumption that coalitions do *not* form if some member would lose service. Yet, coalitions *do already* form if at least one player wins the service. Being reminded of collectors, we call resistance against collective collusion in the new sense *group-strategyproof against collectors* (CGSP).

- In Section 2, we give the formal definition of CGSP and show that it is strictly stronger than WGSP but incomparable to GSP. Moreover, we prove that – contrary to WGSP – any CGSP mechanism induces unique cost-shares.
- In Section 3, we give an algorithm for computing CGSP mechanisms that we call "egalitarian" due to being inspired by Dutta and Ray's [7] "egalitarian solutions". We achieve 1-BB for arbitrary costs and additionally $2H_n$-EFF for the very natural (and rather large) class of subadditive costs.
- In Section 4, we show that our egalitarian mechanisms are a subclass of acyclic mechanisms and that all acyclic mechanisms are CGSP.
- In Section 5, we present a framework for coping with the computational complexity of egalitarian mechanisms. Besides the use of approximation algorithms, the key idea here are "monotonic" cost functions that must not increase when replacing a player by another one with a smaller number.
- In Section 6, we give applications that underline the power of our new approach. For makespan cost-sharing problems, our results are given in Table 1.

Omitted proofs are given in the extended version of this paper.

Table 1. BB and EFF guarantees of best known polynomial-time mechanisms for makespan cost-sharing problems

	GSP mechanisms			**CGSP mechanisms**	
Problem	from	BB	EFF[1]	BB	EFF
general	[1]	$2d$	$\Omega(\log n)$	2	$4H_n$
identical machines[2]	[1]	$\frac{2m}{m+1}$	$\Omega(n)$	$\left\{ 1+\varepsilon \right.$	$\left. 2(1+\varepsilon)H_n \right\}$
	[3]	$\frac{2m-1}{m}$	H_n+1	$\left\{ \frac{4}{3}-\frac{1}{3m} \right.$	$\left. 2(\frac{4}{3}-\frac{1}{3m})H_n \right\}$
identical jobs	[2]	$\frac{\sqrt{17}+1}{4}$	$\Omega(\log n)$	1	$2H_n$

n, m, d: number of jobs, machines, and different processing times, respectively

[1] $\Omega(n)$ is due to [3], $\Omega(\log n)$ follows from the instance with n identical jobs and machines where $\boldsymbol{v} = (\frac{1}{i} - \varepsilon)_{i=1}^{n}$.

[2] CGSP Mechanisms: Upper result based on PTAS with running time exponential in $\frac{1}{\varepsilon}$, lower result achieved with practical algorithm

2 Collectors' Behavior

In the demand for group-strategyproofness lies an implicit modeling assumption that is common to most recent works on cost-sharing mechanisms: First, a player is only willing to be untruthful and join a coalition of false-bidders if this does *not* involve sacrificing her own utility. Second, a coalition always requires an initiating player whose utility *strictly* increases.

Clearly, there are other reasonable behavioral assumptions on coalition formation. We introduce and study the following: First, besides not giving up utility, a player would not sacrifice service, either. (Although her utility is zero both when being served for her valuation and when not being served.) Second, it is sufficient for coalition formation if the initiating player gains *either* utility *or* service. While we consider this behavior very human, it especially reminds us of collectors. We hence denote a mechanism's resistance against coalitions in this new sense as *group-strategyproof against collectors.*

Definition 6. *A mechanism is* group-strategyproof against collectors *(CGSP) if for all* $\boldsymbol{v} \in \mathbb{Q}^n$ *and* $K \subseteq [n]$ *there is no* $\boldsymbol{b} \in \mathbb{Q}^n$ *with* $\boldsymbol{b}_{-K} = \boldsymbol{v}_{-K}$ *such that*

1. $u_i(\boldsymbol{b}) \geq u_i(\boldsymbol{v})$ *and* $i \notin Q(\boldsymbol{v}) \setminus Q(\boldsymbol{b})$ *for all* $i \in K$ *and*
2. $u_i(\boldsymbol{b}) > u_i(\boldsymbol{v})$ *or* $i \in Q(\boldsymbol{b}) \setminus Q(\boldsymbol{v})$ *for at least one* $i \in K$.

We remark that CGSP in a model with quasi-linear utilities is equivalent to GSP in a changed model where a preference of being served for the price of valuation over not being served is internalized in the utilities. To illustrate the interrelation between CGSP and GSP, we introduce a property which is a relaxation of both, called *weakly group-strategyproof against collectors* (WCGSP). Here, (2.) of Definition 6 is replaced by "$u_i(\boldsymbol{b}) > u_i(\boldsymbol{v})$ for at least one $i \in K$".

Lemma 1. *The following implications hold.*

$$\begin{array}{l} GSP \Longrightarrow \\ \qquad\qquad WCGSP \Longrightarrow WGSP \\ CGSP \Longrightarrow \end{array}$$

We remark that Theorem 5 will imply that $Moulin_\xi$ is both GSP and CGSP. Interestingly, already WCGSP is sufficient for a mechanism to induce unique cost shares:

Theorem 1. *Let $M = (Q \times x)$ be a mechanism that satisfies WCGSP. Then, for any two $\boldsymbol{b}, \boldsymbol{b}' \in \mathbb{Q}$ with $Q(\boldsymbol{b}) = Q(\boldsymbol{b}')$, it holds that $x(\boldsymbol{b}) = x(\boldsymbol{b}')$. This result holds even if we restrict our model to non-negative bids and only require CS.*

The proof of Theorem 1 uses ideas from an analogous result in [15]. However, Theorem 1 is stronger since GSP and *strict* CS are relaxed to WCGSP and CS. Conversely, WGSP mechanisms do not always induce (unique) cost-sharing methods, even if we demand 1-BB:

Lemma 2. *For any non-decreasing cost function $C : 2^{[3]} \to \mathbb{Q}_{\geq 0}$, there is a WGSP and 1-BB mechanism $M_C = (Q \times x)$ such that there are bids $\boldsymbol{b}, \boldsymbol{b}' \in \mathbb{Q}_{>0}$ with $Q(\boldsymbol{b}) = Q(\boldsymbol{b}')$, but $x(\boldsymbol{b}) \neq x(\boldsymbol{b}')$.*

3 Egalitarian Mechanisms

Egalitarian mechanisms borrow an algorithmic idea proposed by Dutta and Ray [7] for computing "egalitarian solutions". Given a set of players $Q \subseteq [n]$, cost shares are computed iteratively: Find the *most cost-efficient* subset S of the players that have not been assigned a cost share yet. That is, the quotient of the marginal cost for including S divided by $|S|$ is minimal. Then, assign each player in S this quotient as her cost share. If players remain who have not been assigned a cost share yet, start a new iteration.

Before getting back to most cost-efficient subsets in Section 3.2, we generalize Dutta and Ray's idea by making use of a more general *set selection function* σ and *price function* ρ. Specifically, let $Q \subseteq [n]$ be the set of players to be served. For some fixed iteration, let $N \subsetneq Q$ be the subset of players already assigned a cost-share. Then, $\sigma(Q, N)$ selects the players $S \subseteq Q \setminus N$ who are assigned the cost share $\rho(Q, N)$. We require σ and ρ to be *valid*:

Definition 7. *Set selection and price functions σ and ρ are* valid *if, for all $N \subsetneq Q, Q' \subseteq [n]$:*

1. $\emptyset \neq \sigma(Q, N) \subseteq Q \setminus N$,
2. $Q' \subseteq Q$ *and* $\sigma(Q, N) \subseteq Q' \Longrightarrow \sigma(Q, N) = \sigma(Q', N)$ *and* $\rho(Q, N) = \rho(Q', N)$,
3. $Q' \subseteq Q \Longrightarrow \rho(Q, N) \leq \rho(Q', N)$,
4. $0 \leq \rho(Q, N) \leq \rho(Q, N \cup \sigma(Q, N))$.

Based on valid σ and ρ, we define $Egal_{\sigma,\rho} : \mathbb{Q}^n \to 2^{[n]} \times \mathbb{Q}^n_{\geq 0}$:

Algorithm 1 (Computing Egalitarian Mechanisms $Egal_{\sigma,\rho}(\boldsymbol{b})$).
Input: valid set selection and price functions σ, ρ; bid vector $\boldsymbol{b} \in \mathbb{Q}^n$
Output: set of served players $Q \in 2^{[n]}$; cost-share vector $\boldsymbol{x} \in \mathbb{Q}^n_{\geq 0}$

1: $Q := [n]$; $N := \emptyset$; $\boldsymbol{x} := 0$
2: **while** $N \neq Q$ **do**
3: $\quad S := \sigma(Q, N)$, $a := \rho(Q, N)$
4: $\quad Q := Q \setminus \{i \in S \mid b_i < a\}$
5: $\quad$ **if** $S \subseteq Q$ **then** $x_i := a$ for all $i \in S$; $N := N \cup S$

Theorem 2 (Corollary of Theorems 5, 6)**.** *Egalitarian mechanisms are CGSP.*

3.1 Efficiency of Egalitarian Mechanisms

Definition 8. *Let $C : 2^{[n]} \to \mathbb{Q}_{\geq 0}$ be a cost function, ρ be a price function, and $\beta > 0$. Then, ρ is called β-*average *for C if for all $N \subsetneq Q \subseteq [n]$ and all $\emptyset \neq A \subseteq Q \setminus N$, it holds that $\rho(Q, N) \leq \beta \cdot \frac{C(A)}{|A|}$.*

Lemma 3. *Let $C : 2^{[n]} \to \mathbb{Q}_{\geq 0}$ be a cost function and σ and ρ be valid set selection and price functions such that ρ is β-average for C. Moreover, let $A \subseteq [n]$ and $\boldsymbol{b} \in \mathbb{Q}^n$ be a bid vector with $b_i \geq \beta \cdot \frac{C(A)}{|A|}$ for all $i \in A$. Then, $(Q \times x) := Egal_{\sigma,\rho}$ serves at least one player $i \in A$, i.e., $A \cap Q(\boldsymbol{b}) \neq \emptyset$.*

Theorem 3. *Let $C : 2^{[n]} \to \mathbb{Q}_{\geq 0}$ be a non-decreasing cost function and σ and ρ be valid set selection and price functions such that ρ is β-average for C. Then, if $Egal_{\sigma,\rho}$ always recovers at least the actual cost C', it is $(2\beta \cdot H_n)$-EFF.*

Proof. Let $(Q \times x) := Egal_{\sigma,\rho}$ and $\boldsymbol{v} \in \mathbb{Q}^n$ be the true valuation vector. Denote $Q := Q(\boldsymbol{v})$, $\boldsymbol{x} := x(\boldsymbol{v})$. Moreover, let $P \subseteq [n]$ be a set that minimizes optimal social cost, i.e., $P \in \arg\min_{T \subseteq [n]}\{SC_{\boldsymbol{v}}(T)\}$. W.l.o.g., we may assume that $v_i \geq 0$ for all $i \in [n]$ because C is non-decreasing. We have

$$SC'_{\boldsymbol{v}}(Q) \leq \sum_{i \in Q \cap P} x_i + \sum_{i \in Q \setminus P} \underbrace{x_i}_{\leq v_i} + \sum_{i \in [n] \setminus Q} v_i \leq \sum_{i \in Q \cap P} x_i + \sum_{i \in P \setminus Q} v_i + \sum_{i \in [n] \setminus P} v_i \,,$$

$$\frac{SC'_{\boldsymbol{v}}(Q)}{SC_{\boldsymbol{v}}(P)} \leq \frac{\sum_{i \in Q \cap P} x_i + \sum_{i \in P \setminus Q} v_i + \sum_{i \in [n] \setminus P} v_i}{C(P) + \sum_{i \in [n] \setminus P} v_i} \leq \frac{\sum_{i \in Q \cap P} x_i + \sum_{i \in P \setminus Q} v_i}{C(P)} \,.$$

The last inequality holds since the left fraction is at least 1. Now, consider the iteration k when for the first time Algorithm 1 decides to accept a player $i \in Q \cap P$ (line 5). Fix all variables just after line 3 in that iteration k and indicate them with a subscript k. We have $x_i = a_k = \rho(Q_k, N_k) \leq \beta \cdot \frac{C(Q \cap P)}{|Q \cap P|}$, because $Q \cap P \subseteq Q_k \setminus N_k$. With the same arguments, for the second player $i \in Q \cap P$, we can bound her cost-share $x_i \leq \beta \cdot \frac{C(Q \cap P)}{|Q \cap P| - 1}$, and so forth. Finally, $\sum_{i \in Q \cap P} x_i \leq \beta \cdot H_{|Q \cap P|} \cdot C(Q \cap P)$.

On the other hand, in $P \setminus Q$, there is at least one player i with $v_i < \beta \cdot \frac{C(P \setminus Q)}{|P \setminus Q|}$. Otherwise, due to Lemma 3, we would have $(P \setminus Q) \cap Q \neq \emptyset$, a contradiction.

Inductively and by the same lemma, for every $j = 1, \dots, |P \setminus Q| - 1$, there has to be a player $i \in P \setminus Q$ with $v_i < \beta \cdot \frac{C(P\setminus Q)}{|P\setminus Q|-j}$. Finally, $\sum_{i \in P\setminus Q} v_i \leq \beta \cdot H_{|P\setminus Q|} \cdot C(P \setminus Q)$. Since C is non-decreasing, we get

$$\frac{SC'_{\boldsymbol{v}}(Q)}{SC_{\boldsymbol{v}}(P)} \leq \frac{\beta \cdot H_{\max\{|Q\cap P|,|P\setminus Q|\}} \cdot (C(Q \cap P) + C(P \setminus Q))}{C(P)} \leq 2\beta \cdot H_n \,. \qquad \square$$

3.2 Most Cost-Efficient Set Selection

Definition 9. *Let $C : 2^{[n]} \to \mathbb{Q}_{\geq 0}$ be a cost function. The* most cost-efficient *set selection function σ_C and its corresponding price function ρ_C are defined as*

$$\sigma_C(Q, N) := \textit{ lexicographic max in } \arg \min_{\emptyset \neq T \subseteq Q\setminus N} \left\{ \frac{C(N \cup T) - C(N)}{|T|} \right\} ,$$

$$\rho_C(Q, N) := \min_{\emptyset \neq T \subseteq Q\setminus N} \left\{ \frac{C(N \cup T) - C(N)}{|T|} \right\} .$$

Lemma 4. *For any cost function $C : 2^{[n]} \to \mathbb{Q}_{\geq 0}$, σ_C and ρ_C are valid. If C is subadditive then ρ_C is also 1-average for C.*

Theorem 4 (Corollary of Theorem 3 and Lemma 4)**.** *For any costs $C : 2^{[n]} \to \mathbb{Q}_{\geq 0}$, $Egal_{\sigma_C,\rho_C}$ is CGSP and 1-BB. If C is both non-decreasing and subadditive, then $Egal_{\sigma_C,\rho_C}$ is also $2H_n$-EFF.*

We remark here that already "sequential stand-alone mechanisms" [15] achieve CGSP and 1-BB for non-decreasing cost. Yet, they are only $\Omega(n)$-EFF in general.

Clearly, evaluating σ_C can take exponentially many steps (in n). Furthermore, evaluating C may be computationally hard. In Section 5 we thus study how to pick "suitable" cost-efficient subsets in polynomial time. We conclude by showing that our EFF bound is tight up to a factor of 2.

Lemma 5. *For the cost function $C : 2^{[n]} \to \mathbb{Q}_{\geq 0}$ with $C(T) = 1$ for all $\emptyset \neq T \subseteq [n]$, the mechanism $Egal_{\sigma_C,\rho_C}$ is no better than H_n-EFF.*

Lemma 6. *For any $\gamma > 1$, there is a non-decreasing cost function $C : 2^{[4]} \to \mathbb{Q}_{\geq 0}$ for which the efficiency of mechanism $Egal_{\sigma_C,\rho_C}$ is no better than γ.*

4 Acyclic Mechanisms and CGSP

By introducing acyclic mechanisms, Mehta et al. [14] gave a framework for constructing WGSP mechanisms. We prove that acyclic mechanisms are in fact CGSP, and thus remarkably stronger. That egalitarian mechanisms are CGSP will follow from the observation that they are acyclic.

An acyclic mechanism $Acyc_{\xi,\tau} : \mathbb{Q}^n \to 2^{[n]} \times \mathbb{Q}^n_{\geq 0}$ makes use of a cost-sharing method ξ and an *offer function* $\tau : 2^{[n]} \to \mathbb{Q}^n_{\geq 0}$ which specifies a non-negative offer time $\tau_i(Q)$ for every subset $Q \subseteq [n]$ and every player $i \in Q$. Mehta et al. [14] showed that if ξ and ρ satisfy a certain validity requirement, $Acyc_{\xi,\tau}$ is WGSP.

Algorithm 2 (Computing Acyclic Mechanisms $Acyc_{\xi,\tau}(\boldsymbol{b})$).
Input: cost-sharing method ξ; valid offer function τ; bid vector $\boldsymbol{b} \in \mathbb{Q}^n$
Output: set of players $Q \in 2^{[n]}$, vector of cost shares $\boldsymbol{x} \in \mathbb{Q}^n_{\geq 0}$

$Q := [n]$
while $\exists i \in Q$ with $b_i < \xi_i(Q)$ **do**
  Let $j \in \arg\min_{i \in Q}\{\tau_i(Q) \mid b_i < \xi_i(Q)\}$ (use arbitrary tie breaking rule)
  $Q := Q \setminus \{j\}$
$\boldsymbol{x} := \xi(Q)$

Theorem 5. *Acyclic mechanisms are CGSP.*

Theorem 6. *Egalitarian mechanisms are acyclic.*

Theorem 6 is based on the fact that Algorithm 2 computes $Egal_{\sigma,\rho}$ when given the cost-sharing method $\xi^{\sigma,\rho}$ and offer function $\tau^{\sigma,\rho}$ as defined by Algorithm 3.

Algorithm 3 (Computing $\xi^{\sigma,\rho}(Q)$ and $\tau^{\sigma,\rho}(Q)$).
Input: valid set selection and price functions σ, ρ; set of players $Q \subseteq [n]$
Output: cost-sharing vector $\boldsymbol{\xi} \in \mathbb{Q}^n_{\geq 0}$; offer-time vector $\boldsymbol{\tau} \in \mathbb{Q}^n_{\geq 0}$

$N := \emptyset$; $\boldsymbol{\xi} := 0$; $\boldsymbol{\tau} := 0$
while $N \neq Q$ **do**
  $S := \sigma(Q, N)$, $a := \rho(Q, N)$
  $\xi_i := a$ and $\tau_i := 1 + \max_{j \in Q}\{\tau_j\}$ for all $i \in S$; $N := N \cup S$

We remark that the mechanisms given by Devanur et. al. [6] are not just acyclic (see [14]) but also egalitarian. Using the terminology as in [6], they could be computed by Algorithm 1 by letting $\sigma(Q, N)$ be the next set that "goes tight" after all players in N have been "frozen" and all in $[n] \setminus Q$ have been dropped.

5 A Framework for Polynomial Time Computation

In this section, we show how to solve all of the service provider's tasks in polynomial time by using egalitarian mechanisms with a set selection function that picks the most cost-efficient set *w.r.t. costs of approximate solutions.* Formally, a (cost) minimization problem is a tuple $\Pi = (D, \boldsymbol{S} = (S_I)_{I \in D}, \boldsymbol{f} = (f_I)_{I \in D})$, where D is the set of problem instances (**d**omain) such that for any **i**nstance $I \in D$, S_I is the set of feasible **s**olutions, and $f_I : S_I \to \mathbb{Q}_{\geq 0}$ is a function mapping any solution to its cost.

We write a cost-sharing problem as $\Phi = (\Pi, \textsc{Inst})$, where $\textsc{Inst} : 2^{[n]} \to D$ denotes the function mapping a subset of the n players to the induced instance of Π. In particular, Φ implicitly defines the optimal cost $C : 2^{[n]} \to \mathbb{Q}_{\geq 0}$ by $C(T) := \min_{Z \in S_{\textsc{Inst}(T)}}\{f(Z)\}$. Moreover, for any algorithm $\textsc{alg}$ that computes feasible solutions for Π, we define $C_{\textsc{alg}} : 2^{[n]} \to \mathbb{Q}_{\geq 0}$, $C_{\textsc{alg}}(T) := f(\textsc{alg}(\textsc{Inst}(T)))$.

Resorting to approximate solutions does, of course, not yet remedy the need to iterate through all available subsets in order to pick the most cost-efficient one. The basic idea therefore consists of using an (approximation) algorithm ALG that is *monotonic* (see, e.g., [16]): Seemingly favorable changes to the input must not worsen the algorithm's performance. In the problems considered here, every player is endowed with a size (e.g., processing time in the case of scheduling) and reducing a player's size must not increase the cost of the algorithm's solution. We can then simply number the players in the order of their size such that $C_{\text{ALG}}(\min_{|U|} T) \leq C_{\text{ALG}}(U)$ for all $U \subseteq T \subseteq [n]$. Finding the most cost-efficient set then only requires iterating through all possible cardinalities.

We generalize this basic idea such that only a (polynomial-time computable) monotonic bound C_{mono} on C_{ALG} is needed whereas ALG itself does not need to be monotonic any more.

Definition 10. *Let $\Phi = (\Pi, \text{INST})$ be a cost-sharing problem. A tuple $R := (\text{ALG}, C_{\text{mono}})$ is a* β-relaxation for Φ *if* ALG *is an approximation algorithm for Π and $C_{\text{mono}} : 2^{[n]} \to \mathbb{Q}_{\geq 0}$ is a cost function such that the following holds:*

– *For all $T \subseteq [n]$: $C_{\text{ALG}}(T) \leq C_{\text{mono}}(T) \leq \beta \cdot C(T)$.*
– *For all $U \subseteq T \subseteq [n]$: $C_{\text{mono}}(\min_{|U|} T) \leq C_{\text{mono}}(U)$.*

Note that C_{mono} does not necessarily have to be subadditive (as required for $2H_n$-EFF in Section 3), even if C is. Thus, some additional care is needed.

Given a β-relaxation $R := (\text{ALG}, C_{\text{mono}})$, we define set selection and price functions σ_R and ρ_R recursively as follows. For $N \subsetneq Q \subseteq [n]$, let $\xi^{\sigma_R,\rho_R}(N)$ as computed by Algorithm 3. Furthermore, let

$$k := \max\left\{\arg\min_{i\in[|Q\setminus N|]}\left\{\frac{C_{\text{mono}}(N \cup MIN_i\,(Q\setminus N)) - \sum_{i\in N}\xi_i^{\sigma_R,\rho_R}(N)}{i}, \frac{C_{\text{mono}}(MIN_i\,(Q\setminus N))}{i}\right\}\right\},$$

and $S := MIN_k\,(Q \setminus N)$. Then, $\sigma_R(Q, N) := S$ and

$$\rho_R(Q,N) := \min\left\{\frac{C_{\text{mono}}(N \cup S) - \sum_{i\in N}\xi_i^{\sigma_R,\rho_R}(N)}{k}, \frac{C_{\text{mono}}(S)}{k}\right\}.$$

Note that this recursion is well-defined. Computing $\sigma_R(Q, N)$ and $\rho_R(Q, N)$ requires $\xi^{\sigma_R,\rho_R}(N)$ for which only $\sigma_R(N, \cdot)$ and $\rho_R(N, \cdot)$ is needed (unless $N = \emptyset$). Yet, $N \subsetneq Q$ by assumption.

Lemma 7. *Let $R = (\text{ALG}, C_{\text{mono}})$ be a β-relaxation for some cost-sharing problem Φ. Then σ_R and ρ_R are valid, and ρ_R is β-average for C.*

To also compute a solution for the instance induced by the players selected by $Egal_{\sigma_R,\rho_R}$, we need:

Definition 11. *Let $\Phi = (\Pi, \text{INST})$ with $\Pi = (D, \boldsymbol{S}, \boldsymbol{f})$ be a cost-sharing problem. Then, Φ is called* mergable *if for all disjoint $T, U \subseteq [n]$, $T \cap U = \emptyset$, and for all $X \in S_{\text{INST}(T)}$ and $Y \in S_{\text{INST}(U)}$, there is a $Z \in S_{\text{INST}(T\cup U)}$ with $f(Z) \leq f(X) + f(Y)$. We denote this operation by $Z = X \oplus Y$.*

Based on σ_R and ρ_R, Algorithm 4 solves all of the service provider's tasks, including computing a feasible solution of the underlying optimization problem. We address the running time afterwards.

Algorithm 4 (Computing Egalitarian Mechanisms via β-Relaxations).
Input: β-relaxation $R = (\text{ALG}, C_{\text{mono}})$; bid vector $\boldsymbol{b} \in \mathbb{Q}^n$
Output: player set $Q \in 2^{[n]}$, cost-share vector $\boldsymbol{x} \in \mathbb{Q}^n_{\geq 0}$, solution $Z \in S_{\text{INST}(Q)}$

1: $\boldsymbol{x} := 0$, $Q := [n]$, $N := \emptyset$, $Z :=$ "empty solution"
2: **while** $N \neq Q$ **do**
3: $\quad S := \sigma_R(Q, N)$; $a := \rho_R(Q, N)$
4: $\quad Q := Q \setminus \{i \in S \mid b_i < a\}$
5: $\quad$ **if** $S \subseteq N$ **then**
6: $\qquad Z := \begin{cases} \text{ALG}(\text{INST}(N \cup S)) & \text{if } C_{\text{mono}}(N \cup S) - \sum_{i \in N} x_i \leq C_{\text{mono}}(S) \\ Z \oplus \text{ALG}(\text{INST}(S)) & \text{otherwise} \end{cases}$
7: $\qquad N := N \cup S$; $x_i := a$ for all $i \in S$

Lemma 8. *Let $R = (\text{ALG}, C_{\text{mono}})$ be a β-relaxation for a mergable cost-sharing problem Φ.*

1. *At the end of each iteration of Algorithm 4, it holds that $\boldsymbol{x} = \xi^{\sigma_R, \rho_R}(N)$.*
2. *Line 3 of Algorithm 4 needs at most $2n$ evaluations of C_{mono}.*
3. *The mechanism computed by Algorithm 4 is β-BB.*

Theorem 7 (Corollary of Lemmata 7, 8)**.** *Let Φ be a mergable cost-sharing problem having a β-relaxation $(\text{ALG}, C_{\text{mono}})$. Then the mechanism computed by Algorithm 4 is CGSP, β-BB, and $(2\beta \cdot H_n)$-EFF. Moreover, Algorithm 4 evaluates C_{mono} for no more than $2n^2$ subsets of $[n]$, makes no more than n (direct) calls to ALG, and the number of merge operations is no more than n.*

6 Applications

We use three approaches for obtaining β-relaxations that are polynomial-time computable in the *succinct representation* of the cost-sharing problem plus the bid vector: Monotonic approximation algorithms (e.g., Theorem 8), a non-monotonic approximation algorithm with a monotonic bound C_{mono} (Theorem 9), and optimal costs that are monotonic and polynomial-time computable (discussed at the end of this section).

A makespan cost-sharing problem is *succinctly represented* by a tuple $(\boldsymbol{p}, \boldsymbol{\varsigma})$ where $\boldsymbol{p} \in \mathbb{N}^n$ contains the processing times $p_1 \leq \cdots \leq p_n$ of the n jobs, and $\boldsymbol{\varsigma} \in \mathbb{N}^m$ contains the speeds of the m machines. If $\boldsymbol{p} = 1$ ($\boldsymbol{\varsigma} = 1$), jobs (machines) are *identical*. Each player owns exactly one job. For any set of served players $S \subseteq [n]$, $C(S)$ is the value of a minimum-makespan schedule for S.

A bin packing cost-sharing problem is succinctly represented by a vector of object sizes $\boldsymbol{s} \in \mathbb{Q}^n_{>0}$ with $s_1 \leq \cdots \leq s_n \leq 1$. The capacity of a single bin is 1. Each player owns exactly one object. For any set of players $S \subseteq [n]$, $C(S)$ is the minimum number of bins needed to serve S.

Note here that we assume that each player is given a unique number $\in [n]$ in advance (outside the scope of Algorithm 4) and that players are sorted according to the respective monotonicity criterion.

Lemma 9. *Any bin packing or makespan cost-sharing problem $\Phi = (\Pi, \textsc{Inst})$ is mergable in time $O(n)$. Moreover,* Inst *is computable in linear time (in the size of the succinct representation of Φ).*

First, we consider identical-machine makespan cost-sharing problems. Their succinct representation is $(\boldsymbol{p}, m)$. The LPT (longest processing time first) heuristic [8] is known to be a $\frac{4m-1}{3m}$-approximation algorithm for this problem. It processes the jobs in decreasing order and assigns each job to the machine on which its completion time will be smallest. Its running time is $O(n \cdot \log n)$ for the sorting phase and $O(n \cdot \log m)$ for the job assignment phase. We show that LPT is monotonic w.r.t. processing times:

Lemma 10. *Let $\boldsymbol{p}, \boldsymbol{p}' \in \mathbb{N}^n$, $i \in [n]$, $p_i > p'_i$, and $\boldsymbol{p}_{-i} = \boldsymbol{p}'_{-i}$. Then it holds that $f(\textsc{lpt}(\boldsymbol{p}, m)) \geq f(\textsc{lpt}(\boldsymbol{p}', m))$.*

Theorem 8 (Corollary of Lemma 10**).** *For any identical-machine makespan cost-sharing problem with succinct representation $(\boldsymbol{p}, m)$, where $p_1 \leq \cdots \leq p_n$, it holds that $(\textsc{lpt}, C_{\textsc{lpt}})$ is a $\frac{4m-1}{3m}$-relaxation and Algorithm 4 runs in time $O(n^3 \cdot \log m)$.*

Besides the previous result, we show how to adapt the PTAS (for identical machines) by Hochbaum and Shmoys [12]. The approach is different to before: Not the PTAS itself is monotonic but a bound computed inside the algorithm.

The basic idea of the PTAS is a reduction to bin packing: Given processing times $\boldsymbol{p} \in \mathbb{N}^n$, binary search is employed in order to find a makespan d such that the bin packing instance $\frac{\boldsymbol{p}}{d}$ does *not* need more than m bins of size $(1+\varepsilon)$, whereas the bin packing instance $\frac{\boldsymbol{p}}{d-1}$ *does* need more than m bins. Specifically, the PTAS makes use of an ϵ-dual approximation algorithm $\textsc{bpDual}_\varepsilon$ for the bin packing problem (see [12], pp.149–151). $\textsc{bpDual}_\varepsilon$ outputs solutions that are *dual feasible*; this means that $\textsc{bpDual}_\varepsilon$ may use bins of size $(1+\epsilon)$.

Now, for any bin packing instance $\boldsymbol{s} \in \mathbb{Q}^n_{>0}$, let $S^*_{\boldsymbol{s}} \supseteq S_{\boldsymbol{s}}$ be the set of all dual-feasible solutions and $f^*_{\boldsymbol{s}} : S^*_{\boldsymbol{s}} \to \mathbb{N}$ be a function mapping each dual-feasible solution to its cost, i.e., the number of used bins. We define $g^*_{\boldsymbol{s}} : S^*_{\boldsymbol{s}} \to \mathbb{N}$, $g^*_{\boldsymbol{s}}(Z) := \max\{f^*_{\boldsymbol{s}}(Z), \lceil\sum_{i\in[n]} s_i\rceil\}$. Hence, the crucial property of g^* is to guarantee that $g^*_{\boldsymbol{s}}$ is never less than the number of bins needed for a feasible optimal solution, i.e., when bins have capacity 1. We show that g^* is monotonic.

Lemma 11. *Let $\boldsymbol{s}, \boldsymbol{s}' \in \mathbb{Q}^n_{\geq 0}$ be two vectors of object sizes, $i \in [n]$, $s_i > s'_i$, and $\boldsymbol{s}_{-i} = \boldsymbol{s}'_{-i}$. Then $b := g^*_{\boldsymbol{s}}(\textsc{bpDual}_\epsilon(\boldsymbol{s})) \geq g^*_{\boldsymbol{s}'}(\textsc{bpDual}_\epsilon(\boldsymbol{s}')) =: b'$.*

Our crucial modification of the PTAS is as follows: Letting $\boldsymbol{s} := \frac{\boldsymbol{p}}{d}$, we use the check $g^*_{\boldsymbol{s}}(\textsc{bpDual}_\varepsilon(\boldsymbol{s})) \leq m$ in the binary search (instead of testing $f^*_{\boldsymbol{s}}$ as in the original PTAS). Moreover, we let $\mathit{lower}_\varepsilon(\boldsymbol{p})$ denote the minimum d for which this check evaluates to true and let $\textsc{hs}_\varepsilon$ denote the adapted PTAS. One can easily verify that $\mathit{lower}_\varepsilon(\boldsymbol{p})$ is a lower bound on the optimal makespan and $(1+\varepsilon) \cdot \mathit{lower}_\varepsilon(\boldsymbol{p})$ is an upper bound on the schedule found by $\textsc{hs}_\varepsilon$. Moreover, $\mathit{lower}_\varepsilon(\boldsymbol{p})$ is computed within $\textsc{hs}_\varepsilon$ in polynomial time because monotonicity of g^* ensures that indeed the minimum d is found by the binary search.

Theorem 9 (Corollary of Lemma 11)**.** *For any identical-machine makespan cost-sharing problem with succinct representation* $(\boldsymbol{p}, m)$*, where* $p_1 \leq \cdots \leq p_n$*, let* $C_{\text{mono}}(A) := (1+\varepsilon) \cdot \mathit{lower}_\varepsilon(\textsc{Inst}(A))$*. Then,* $(\textsc{hs}_\epsilon, C_{\text{mono}})$ *is a* $(1+\varepsilon)$*-relaxation and Algorithm 4 runs in time* $O(n^{2+\frac{1}{\varepsilon^2}} \cdot \log \sum_{i \in [n]} p_i)$*.*

Finally, we also obtain 2-relaxations for bin packing and general makespan cost-sharing problems:

Lemma 12. *For any bin packing cost-sharing problem with succinct representation* $\boldsymbol{s}$*, there is a 2-relaxation for* C *and Algorithm 4 runs in time* $O(n^3 \cdot \log n)$*.*

Theorem 10. *For any makespan cost-sharing problem with succinct representation* $(\boldsymbol{p}, \boldsymbol{\varsigma})$*, there is a 2-relaxation for* C *and Algorithm 4 runs in time* $O(n^3 \cdot \log m \cdot \log \sum_{i \in [n]} p_i)$*.*

There are several mergable scheduling problems for which optimal costs are monotonic and computable in polynomial time. For instance, for any identical-job makespan cost-sharing problem with succinct representation $(n, \boldsymbol{\varsigma})$, it holds that $(\textsc{lpt}, C_{\textsc{lpt}})$ is a 1-relaxation and Algorithm 4 runs in time $O(n^3 \cdot \log m)$. In the following, we give a selection of further problems (taken from [4] and using the classification scheme introduced by Graham et al. [9]). We restrict our attention to the cases in which only one of the properties p_i, w_i (weight), and r_i (release date) is variable and let the others be fixed with $p_i = 1$, $w_i = 1$, and $r_i = 0$. We get that 1-relaxations exist for:

– Variable processing times: $Q|\text{pmtn}|C_{\max}$, $Q||\sum_i C_i$, $Q|\text{pmtn}|\sum_i C_i$
– Variable weights: $P||\sum_i w_i C_i$, $P|\text{pmtn}|\sum_i w_i C_i$
– Variable release dates: $Q|\text{pmtn}|C_{\max}$

The result for $Q||\sum_i C_i$ especially implies 1-BB for $1||\sum_i C_i$. This is a drastic improvement over Moulin mechanisms, since no cross-monotonic cost-sharing method can be better than $\frac{n+1}{2}$-BB [3].

7 Conclusion and Future Work

The pivotal point of this work is our new modeling assumption on coalition formation. We believe that CGSP is a viable replacement for the often too limiting GSP requirement. Besides this novel structural property, we consider the main asset of our work to be threefold: i) Egalitarian mechanisms; showing existence of CGSP, 1-BB, and $2H_n$-EFF mechanisms for any non-decreasing subadditive costs. ii) Our framework for polynomial-time computation that reduces constructing CGSP, $O(1)$-BB, and $O(\log n)$-EFF mechanisms to finding monotonic approximation algorithms. iii) Showing that acyclic mechanisms are CGSP and thus remarkably stronger than was known before.

An immediate issue left often by our work is, of course, to find more applications of our polynomial-time framework. For instance, it is easy to see that (rooted) Steiner tree cost-sharing problems are mergable and their costs non-decreasing and subadditive; but do they allow for a β-relaxation?

References

1. Bleischwitz, Y., Monien, B.: Fair cost-sharing methods for scheduling jobs on parallel machines. In: Proc. of the 6th Italian Conference on Algorithms and Complexity. LNCS 3998. (2006) 175–186
2. Bleischwitz, Y., Monien, B., Schoppmann, F., Tiemann, K.: The power of two prices: Beyond cross-monotonicity. In: Proc. of the 32th International Symposium on Mathematical Foundations of Computer Science. LNCS 4708. (2007) 657–668
3. Brenner, J., Schäfer, G.: Cost sharing methods for makespan and completion time scheduling. In: Proc. of the 24th International Symposium on Theoretical Aspects of Computer Science. LNCS 4393. (2007) 670–681
4. Bruckner, P.: Scheduling Algorithms. Springer (1995)
5. Chawla, S., Roughgarden, T., Sundararajan, M.: Optimal cost-sharing mechanisms for Steiner forest problems. In: Proc. of the 2nd Internet and Network Economics. Volume 4286 of LNCS. (2006) 112–123
6. Devanur, N.R., Mihail, M., Vazirani, V.V.: Strategyproof cost-sharing mechanisms for set cover and facility location games. Decision Support Systems **39**(1) (2005) 11–22
7. Dutta, B., Ray, D.: A concept of egalitarianism under participation constraints. Econometrica **57**(3) (1989) 615–635
8. Graham, R.: Bounds on multiprocessing timing anomalies. SIAM Journal of Applied Mathematics **17**(2) (1969) 416–429
9. Graham, R., Lawler, E., Lenstra, J., Rinnooy Kan, A.: Optimization and approximation in deterministic sequencing and scheduling: A survey. Annals of Discrete Mathematics **5** (1979) 287–326
10. Green, J., Laffont, J.: Characterizations of satisfactory mechanisms for the revelation of preferences for public goods. Econometrica **45** (1977) 427–438
11. Gupta, A., Könemann, J., Leonardi, S., Ravi, R., Schäfer, G.: An efficient cost-sharing mechanism for the prize-collecting Steiner forest problem. In: Proc. of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms. (2007)
12. Hochbaum, D., Shmoys, D.: Using dual approximation algorithms for scheduling problems: theoretical and practical results. Journal of the ACM **34**(1) (1987) 144–162
13. Immorlica, N., Mahdian, M., Mirrokni, V.: Limitations of cross-monotonic cost sharing schemes. In: Proc. of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms. (2005) 602–611
14. Mehta, A., Roughgarden, T., Sundararajan, M.: Beyond Moulin mechanisms. In: Proc. of the 8th ACM Conference on Electronic Commerce. (2007) 1–10
15. Moulin, H.: Incremental cost sharing: Characterization by coalition strategy-proofness. Social Choice and Welfare **16**(2) (1999) 279–320
16. Murgolo, F.D.: Anomalous behavior in bin packing algorithms. Discrete Applied Mathematics **21**(3) (October 1988) 229–243
17. Penna, P., Ventre, C.: The algorithmic structure of group strategyproof budget-balanced cost-sharing mechanisms. In: Proc. of the 23rd International Symposium on Theoretical Aspects of Computer Science. LNCS 3884. (2006) 337–348
18. Roughgarden, T., Sundararajan, M.: New trade-offs in cost-sharing mechanisms. In: Proc. of the 38th ACM Symposium on Theory of Computing. (2006) 79–88
19. Roughgarden, T., Sundararajan, M.: Optimal efficiency guarantees for network design mechanisms. In: Proc. of the 12th Conference on Integer Programming and Combinatorial Optimization. (2007) 469–483