The Power of Two Prices: Beyond Cross-Monotonicity*

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

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

Abstract. Assuming strict consumer sovereignty (CS*), when can cost-sharing mechanisms simultaneously be group-strategyproof (GSP) and β-budget-balanced (β-BB)? Moulin mechanisms are GSP and 1-BB for submodular costs. We overcome the submodularity requirement and instead consider arbitrary—yet symmetric—costs:

- Already for 4 players, we show that symmetry of costs is not sufficient for the existence of a GSP and 1-BB mechanism. However, for only 3 players, we give a GSP and 1-BB mechanism.
- We introduce *two-price cost-sharing forms* (2P-CSFs) that define players' cost shares and present a novel mechanism that is GSP given any such 2P-CSF. For subadditive costs, we give an algorithm to compute 2P-CSFs that are $\frac{\sqrt{17}+1}{4}$-BB (≈ 1.28). This result is then shown to be tight for 2P-CSFs. Yet, this is a significant improvement over 2-BB, which is the best Moulin mechanisms can achieve.
- We give applications to the minimum makespan scheduling problem.

A key feature of all our mechanisms is a preference order on the set of players. Higher cost shares are always payed by least preferred players.

1 Introduction and Model

1.1 Motivation

Consider a computing center with a large cluster of parallel machines that offers (uninterrupted) processing times. First, potential customers submit a maximum payment they would be willing to contribute for having their jobs processed. Then, solely based on these messages, the computing center uses a (commonly known) algorithm to determine both the served customers and their eventual payments. In addition, it computes a schedule for the accepted jobs.

Deciding on prices only after receiving *binding bids* by customers enables the computing center to determine an outcome that both ensures recovery of its own cost as well as competitive prices in that its surplus is always relatively small.

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

In fact, if the computing center is a public institution, this *budget-balance* might even be a legal requirement. Since prices are bid-dependent, selfishness of customers elevates the computing center's scheduling problems to a game-theoretic context: How can the service provider ensure *group-strategyproofness* such that (coalitional) strategic bidding corresponds to submitting true *valuations*?

The above scenario is an example for *cost sharing*: A service provider offers some service which $n \in \mathbb{N}$ selfish players are interested in. This interest is defined by a player's true valuation $v_i \in \mathbb{Q}$ for being served, which is private information. The protocol used for negotiation is simple: The service provider elicits bids $b_i \in \mathbb{Q}$ from the players that are supposed to indicate their (alleged) valuations for being served. It uses these bids b_i as a proxy for the true v_i's and employs a commonly known *cost-sharing mechanism* to determine both the set of served players and their payments, referred to as *cost shares* in the following.

1.2 The Model

Notation. For $n \in \mathbb{N}_0$, let $[n] := \{1, \ldots, n\}$ and $[n]_0 := [n] \cup \{0\}$. For all $A \subseteq [n]$ and all $i \in A$, let $\textsc{Rank}(i, A) := |\{j \in A \mid j \leq i\}|$. Given $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{Q}^n$ and $A \subseteq [n]$, let $\boldsymbol{x}_A := (x_i)_{i \in A} \in \mathbb{Q}^{|A|}$ and $\boldsymbol{x}_{-A} := \boldsymbol{x}_{[n] \setminus A}$. Let $(\boldsymbol{x}_{-A}, \boldsymbol{y}_A) \in \mathbb{Q}^n$ denote the vector where the components in $\boldsymbol{x}$ for A are replaced by the respective ones from $\boldsymbol{y}$. For $\boldsymbol{z} \in \mathbb{Q}^m$, $(\boldsymbol{x}; \boldsymbol{z}) \in \mathbb{Q}^{n+m}$ is the vector containing the components of both $\boldsymbol{x}$ and $\boldsymbol{z}$. By convention, the vector of the players' true valuations is always $\boldsymbol{v} \in \mathbb{Q}^n$, whereas an actual bid vector is denoted $\boldsymbol{b} \in \mathbb{Q}^n$.

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 cost-sharing mechanisms are required to fulfill three standard properties:

- *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 in our model, VP implies that players may opt to not participate (by submitting a negative bid). This property together with CS is referred to as *strict consumer sovereignty* (CS*).

We assume the utility $u_i : \mathbb{Q}^n \to \mathbb{Q}$ of any player $i \in [n]$ 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, mechanisms should elicit truthful bids ($\boldsymbol{b} = \boldsymbol{v}$) even if players may collude:

Definition 2. *A mechanism is* group-strategyproof *(GSP) if for every true valuation vector* $\boldsymbol{v} \in \mathbb{Q}^n$ *and any coalition* $K \subseteq [n]$ *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* $i \in K$.

The provider's cost of serving a set of players $Q \subseteq [n]$ is defined by a *cost function* $C : 2^{[n]} \to \mathbb{Q}_{\geq 0}$ with $C(A) = 0 \iff A = \emptyset$. Of particular interest are:

- *Symmetric costs*: Costs depend only on the cardinality, i.e., for all $A, B \subseteq [n]$ with $|A| = |B|$: $C(A) = C(B)$. In this case, we usually define costs solely by a function of the cardinality, $c : [n]_0 \to \mathbb{Q}_{\geq 0}$, where $C(A) := c(|A|)$.
- *Subadditive costs*: The marginal cost of adding a set of players is never more than the stand-alone cost of this set. That is, for any two sets $A, B \subseteq [n]$: $C(A) + C(B) \geq C(A \cup B)$.
- *Submodular costs*: For all $A, B \subseteq [n]$: $C(A) + C(B) \geq C(A \cup B) + C(A \cap B)$.

A mechanism should be budget-balanced with respect to the incurred cost:

Definition 3. *A mechanism $M = (Q \times x)$ is* β-budget-balanced (β-BB, for $\beta \geq 1$) w.r.t. cost C *if for all* $\boldsymbol{b} \in \mathbb{Q}^n : C(Q(\boldsymbol{b})) \leq \sum_{i \in Q(\boldsymbol{b})} x_i(\boldsymbol{b}) \leq \beta \cdot C(Q(\boldsymbol{b}))$.

GSP is a strong property in that cost shares selected by a GSP mechanism only depend on the set of served players and not on the bids [13]. This gives rise to:

Definition 4. *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 *induces* a unique cost-sharing method ξ, by setting $\xi(A) := x(\boldsymbol{b})$ where $b_i < 0$ if $i \notin A$ and $b_i = b_i^+$ if $i \in A$. Three special properties of a cost-sharing method ξ are of importance in this work:

- *β-Budget-balance (w.r.t. C)*: For all $A \subseteq [n] : C(A) \leq \sum_{i \in A} \xi_i(A) \leq \beta \cdot C(A)$.
- *Cross-monotonicity*: For any player $i \in [n]$, cost shares are non-increasing as the set of players gets larger, i.e., for all $A, B \subseteq [n] : \xi_i(A \cup B) \leq \xi_i(A)$.
- *Preference order* (for symmetric costs): ξ has a succinct representation by vectors $\boldsymbol{\xi}^j \in \mathbb{Q}^j_{\geq 0}$ for all $j \in \mathbb{N}$: $\xi_i(A) := \xi^{|A|}_{\textsc{Rank}(i,A)}$ if $i \in A$ and 0 otherwise. Example: If $n = 5$, $A = \{2, 4\}$, and $\boldsymbol{\xi}^2 = (2, 1)$, then $\xi(A) = (0, 2, 0, 1, 0)$.

While the first two properties are standard in the realm of cost sharing, the third one is essentially "new" and crucial for our results. A preference order ensures that the cost share of any player i in a set of served players $A \subseteq [n]$ only depends on the rank of i in A and the cardinality of A.

In his seminal work [13], Moulin gave the straightforward mechanism Moulin_ξ that is GSP given any cross-monotonic cost-sharing method ξ. Moulin_ξ repeatedly rejects players whose bids are below their current cost shares until all remaining players can afford their payments. For any GSP mechanism M with induced cross-monotonic cost shares ξ, Moulin_ξ produces the same utility for each player as M. Therefore, we call any GSP mechanism with cross-monotonic cost shares a *Moulin mechanism*.

1.3 Related Work

Moulin [13] completely characterizes the impact of submodular costs on GSP and 1-BB: Any GSP mechanism that is 1-BB w.r.t. submodular costs is a Moulin mechanism. Conversely, for any submodular cost function $C : 2^{[n]} \rightarrow \mathbb{Q}_{\geq 0}$, a cross-monotonic 1-BB cost-sharing method always exists. Besides this result, characterizations have also been obtained for so-called *upper-continuous* mechanisms [8,14], which are a superclass of Moulin mechanisms.

Essentially all known GSP mechanisms are Moulin mechanisms; confer, e.g., [1,7,11,2,9,10,3,6,4]. For scheduling, when costs are defined as the optimal makespan, Bleischwitz and Monien [3] give $\frac{2m}{m+1}$-BB cross-monotonic cost-sharing methods in case of identical jobs (m is the number of machines). They show that no Moulin mechanism can perform better. For arbitrary processing times, they give a $2d$-BB cross-monotonic cost-sharing method, where d is the number of different processing times, and show tightness up to a factor of 2.

In the (implausible) case that players have no means of refusing service (technically: when bids are required to be non-negative and CS* is not required), Immorlica et al. [8] present simple 1-BB and GSP mechanisms.

Recently, trading off *social cost* and BB has become an active direction of research. For details and a definition confer [12,4,6] and their references.

1.4 Contribution

Assuming strict consumer sovereignty (CS*), we obtain the following results:

1. In the vein of previous characterization attempts [13,8], we study the impact of symmetric costs on GSP and 1-BB in Section 3: Already for 4 players, we show that symmetry of costs is not sufficient for the existence of a GSP and 1-BB mechanism any more. However, for only 3 players, we give a GSP and 1-BB mechanism (based on the techniques from Section 2).
2. For symmetric costs, we introduce *two-price cost-sharing forms* (2P-CSFs) in Section 2 that define players' cost shares. The attribute 'two-price' is to indicate that for any set of served players, there are at most two different cost shares. For any 2P-CSF F, we present a novel mechanism $\mathit{MechCSF}_F$ that is GSP. This is analogous to Moulin_ξ which is GSP if the cost-sharing method ξ is cross-monotonic. The usefulness of our new technique lies in the fact that 2P-CSFs do not necessarily represent cross-monotonic cost-sharing methods. Hence, for certain classes of cost functions, 2P-CSFs allow for better budget approximations than cross-monotonic cost-sharing methods. In particular, for symmetric and subadditive costs, we give an algorithm to compute 2P-CSFs that are $\frac{\sqrt{17}+1}{4}$-BB. We show that, in general, this is the best 2P-CSFs can yield. Yet, this significantly improves over 2-BB, which is the best possible for cross-monotonic cost shares [3].
3. We apply our technique to the scheduling problem of minimizing makespan on related machines in Section 2.3. We obtain a quadratic-time algorithm for computing GSP and $(\frac{\sqrt{17}+1}{4} \cdot d)$-BB mechanisms, where d is the number of different processing times. This beats the previously best-known BB of $2d$ [3]. We are able to extend our techniques to guarantee GSP and 1-BB for the case of identical jobs and identical machines and a specific (non-symmetric) scheduling setting on 3 identical machines and processing times 1 and 2. Unfortunately, the same approach fails for 4 identical machines.

All our algorithms are based on one basic idea: They compute mechanisms with preference-ordered cost shares. Most players are charged a reasonable lower cost share (at least the minimum average per-player cost over all possible sets of players) while some less preferred players have to reimburse the remaining cost.

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

2 Two Price Cost-Sharing Forms

Before looking at *two* prices, we state what can be achieved with *one* price:

Lemma 1. *For any non-decreasing, symmetric, and subadditive cost function* $c : [n]_0 \to \mathbb{Q}_{>0}$*, there is a GSP and* 2*-BB mechanism that always charges all served players equally. If* c *is also submodular, this mechanism is even* 1*-BB. However, for any* $\varepsilon > 0$*, there is a non-decreasing, symmetric, and subadditive cost function for which no such GSP and* $(2 - \varepsilon)$*-BB mechanism exists.*

In the following, we give GSP mechanisms that perform better with respect to BB. They use at most two different cost shares for any set of served players.

Definition 5. *A* two-price cost-sharing form *(2P-CSF) is a tuple* $F = (n, \boldsymbol{h}, \boldsymbol{l}, \boldsymbol{d})$ *where for each cardinality* $i \in [n]_0$ *of the set of served players*

- $h_i \in \mathbb{Q}_{>0}$ *is the* ***h**igher,* $l_i \in \mathbb{Q}_{>0}$ *(with* $l_i < h_i$*) is the* ***l**ower cost share, and*
- $d_i \in [i]_0$ *is the number of* ***d**isadvantaged players paying* h_i*.*

Note that $d_0 = 0$ by definition and neither h_0 nor l_0 are actually used; cardinality 0 is included only to avoid undesired case analyses later on. A 2P-CSF $(n, \boldsymbol{h}, \boldsymbol{l}, \boldsymbol{d})$ is a succinct representation of vectors $\boldsymbol{\xi}^i \in \mathbb{Q}^i$, $i \in [n]$, which define the preference ordered cost-sharing method $\xi : 2^{[n]} \to \mathbb{Q}^n$ by

$$\boldsymbol{\xi}^i := (\underbrace{h_i, \ldots, h_i}_{d_i}, \underbrace{l_i, \ldots, l_i}_{i-d_i \text{ elements}}) .$$

We call a contiguous range $\{s, s+1, \ldots, t\} \subseteq [n]_0$ of cardinalities with $d_s = d_{t+1} = 0$ (let $d_{t+1} := 0$ if $t = n$), and $d_k > 0$ for $k \in \{s+1, \ldots, t\}$ a *segment*. That is, only at the beginning of a segment there is no disadvantaged player paying the higher cost share. Furthermore, a segment is maximal with this property.

Definition 6. *A 2P-CSF* $(n, \boldsymbol{h}, \boldsymbol{l}, \boldsymbol{d})$ *is* valid *if for each cardinality* $i \in [n]$*:*

- *Lower cost shares are non-increasing (in the cardinality) and stay the same within a segment, i.e.,* $l_i \leq l_{i-1}$ *and* $(l_i < l_{i-1} \Longrightarrow d_i = 0)$*.*
- *Higher cost shares may only increase at the beginning of a segment, i.e.,* $h_i > h_{i-1} \Longrightarrow d_i = 0$*.*
- *Adding a single player may not increase the number of disadvantaged players by more than 1. Moreover, if the higher cost share decreases, only one disadvantaged player may remain, i.e.,* $d_i \leq d_{i-1} + 1$ *and* $(h_i < h_{i-1} \Longrightarrow d_i \leq 1)$*.*

We define $\gamma : [n]_0 \to \mathbb{Q}_{\geq 0}$, $\gamma(i) := d_i \cdot h_i + (i - d_i) \cdot l_i$ as the *recovered cost.*

2.1 GSP Mechanisms for Two-Price Cost-Sharing Forms

Algorithm 1 (Computing mechanism $\mathit{MechCSF}_F$, for a 2P-CSF F).
Input: 2P-CSF $F = (n, \boldsymbol{h}, \boldsymbol{l}, \boldsymbol{d})$, bid vector $\boldsymbol{b} \in \mathbb{Q}^n$
Output: set of players $Q \in 2^{[n]}$, vector of cost shares $\boldsymbol{x} \in \mathbb{Q}^n_{\geq 0}$

1: $k := \max\left\{ i \in [n]_0 \;\middle|\; |\{j \in [n] \mid b_j \geq l_i\}| \geq i \right\}$; $l := l_k$ ▷ Find segment
2: $Q := \{i \in [n] \mid b_i \geq l\}$ ▷ Players still in the game
3: $I := \{i \in [n] \mid b_i = l\}$ ▷ **I**ndifferent players
4: $D := \emptyset$ ▷ **D**isadvantaged players with higher cost share
5: **loop**
6: $\quad q := \max\{i \in [|Q|]_0 \mid d_i = |D|\}$ ▷ Max # players if players in D pay $> l$
7: $\quad$ **if** $q \geq |Q \setminus I|$ **then** ▷ Possible to serve all non-indifferent players?
8: $\quad\quad Q := Q \setminus \{|Q| - q$ smallest elements of $I\}$ ▷ Remove "excess"
9: $\quad\quad$ **break**
10: $\quad \ell := \min(Q \setminus D)$ ▷ Least preferred player not yet in D
11: $\quad$ **if** $b_\ell \geq h_{|Q|}$ **then** $D := D \cup \{\ell\}$ ▷ Make disadvantaged
12: $\quad$ **else** $Q := Q \setminus \{\ell\}$; $I := I \setminus \{\ell\}$
13: $x_i := h_{|Q|}$ for $i \in D$; $x_i := l$ for $i \in Q \setminus D$; $x_i := 0$ for $i \in [n] \setminus Q$

Informally, Algorithm 1 works as follows.

1. Find largest $k \in \mathbb{N}$ such that there are k players who bid at least the respective lower cost share $l_k =: l$. Note that k is already in the correct segment.
2. Reject all players who do not even bid l.
3. If possible, include all players still in the game with bid $b_i > l$ for price l, by rejecting a suitable subset of the indifferent players ($b_i = l$); then stop.
4. Otherwise, include the least preferred player (with lowest number) for the current higher cost share or, if she bids less than that, reject her. Go to 3.

The intuition is that including the least preferred player for the current higher cost share never harms the other players. Instead, it may even benefit in that more players can be served for l afterwards. Once a player is included for a higher cost share, this cost share remains fixed during the further execution.

Theorem 1. *Let $F = (n, \boldsymbol{h}, \boldsymbol{l}, \boldsymbol{d})$ be a valid 2P-CSF. Then $\mathit{MechCSF}_F$ is GSP and can be computed by Algorithm 1 in time $O(n^2)$.*

Proof (Sketch). For any input $\boldsymbol{b} \in \mathbb{Q}^n$, denote by $k(\boldsymbol{b})$ and $l(\boldsymbol{b})$ the values of k and l in line 1 of Algorithm 1. Moreover, set $s(\boldsymbol{b}) := \max\{j \in [k(\boldsymbol{b})]_0 \mid d_j = 0\}$ to the beginning of the segment that $k(\boldsymbol{b})$ is in. Note that $s(\boldsymbol{b}) \leq |Q(\boldsymbol{b})| \leq k(\boldsymbol{b})$.

Denote by $\boldsymbol{v} \in \mathbb{Q}^n$ the true valuation vector and let $K \subseteq [n]$ be a successful coalition, i.e., there is a bid vector $\boldsymbol{b} \in \mathbb{Q}^n$ with $\boldsymbol{b}_{-K} = \boldsymbol{v}_{-K}$ such that $u_j(\boldsymbol{b}) \geq u_j(\boldsymbol{v})$ for all $j \in K$ and $u_i(\boldsymbol{b}) > u_i(\boldsymbol{v})$ for at least one $i \in K$.

First assume $s(\boldsymbol{b}) > s(\boldsymbol{v})$. Then also $s(\boldsymbol{b}) > k(\boldsymbol{v})$, thus $|Q(\boldsymbol{b})| > k(\boldsymbol{v})$. Hence, there is a player $j \in Q(\boldsymbol{b})$ with $v_j < l(\boldsymbol{b})$ because $k(\boldsymbol{v})$ would not have been maximal in line 1 otherwise. Since $j \in Q(\boldsymbol{b})$, it holds that $b_j \geq x_j(\boldsymbol{b}) \geq l(\boldsymbol{b}) > v_j$, so we have $j \in K$ and $u_j(\boldsymbol{b}) < 0 \leq u_j(\boldsymbol{v})$, a contradiction.

Now consider $s(\boldsymbol{b}) < s(\boldsymbol{v})$. Then also $k(\boldsymbol{b}) < s(\boldsymbol{v})$. Define $L := \{j \in [n] \mid b_j \geq l(\boldsymbol{v})\}$. Clearly, $|L| < s(\boldsymbol{v})$ as $k(\boldsymbol{b})$ would not have been maximal in line 1 otherwise. Now, let M be a set of $s(\boldsymbol{v}) - |L|$ players $j \in [n] \setminus L$ with $l(\boldsymbol{v}) \leq v_j$. Such a set M exists. Define a new bid vector $\boldsymbol{b}' \in \mathbb{Q}^n$ by $b'_j := l(\boldsymbol{v})$ for $j \in M$ and $b'_j := b_j$ otherwise. Then $s(\boldsymbol{b}') = |Q(\boldsymbol{b}')| = k(\boldsymbol{b}') = s(\boldsymbol{v})$ and $u_j(\boldsymbol{b}') \geq u_j(\boldsymbol{b})$ for all players $j \in [n]$. W.l.o.g., we may thus assume $s(\boldsymbol{b}) = s(\boldsymbol{v})$ in the following. For clarity let $l := l(\boldsymbol{b}) = l(\boldsymbol{v})$.

Since $u_i(\boldsymbol{b}) > u_i(\boldsymbol{v})$, we have $i \in Q(\boldsymbol{b})$ and $x_i(\boldsymbol{b}) < h_{|Q(\boldsymbol{v})|}$ and ($i \notin Q(\boldsymbol{v})$ or $x_i(\boldsymbol{v}) = h_{|Q(\boldsymbol{v})|}$). At least one of the following has to be fulfilled:

- The number of disadvantaged players paying the higher cost share increased from $\boldsymbol{v}$ to $\boldsymbol{b}$. That is, $\exists j \in [i-1] : b_j \geq h_{|Q(\boldsymbol{b})|}$ and $h_{|Q(\boldsymbol{v})|} > v_j$.
- A player j that the mechanism prefers to i (and who got the service for l) waived being served. That is, $\exists j \in \{i+1, \dots, n\} : b_j \leq l < v_j$ and $j \notin Q(\boldsymbol{b})$.
- The total number of players served for the lower cost share l increased. That is, $\exists j \in [n] : b_j \geq l > v_j$ and $j \in Q(\boldsymbol{b})$.

In either case, player j is part of the coalition K and $u_j(\boldsymbol{b}) < u_j(\boldsymbol{v})$. $\square$

2.2 $\frac{\sqrt{17}+1}{4}$-BB Two-Price Cost-Sharing Forms for Subadditive Costs

Theorem 2. *Let $\varepsilon > 0$. For any non-decreasing, symmetric, and subadditive cost function $c : [n]_0 \to \mathbb{Q}_{\geq 0}$, there is a $\left(\frac{\sqrt{17}+1}{4} + \varepsilon\right)$-BB 2P-CSF. Moreover (if c is given as an array of n function values), it can be computed in time $O(n)$.*

Proof. Algorithm 2 computes a 2P-CSF for increasing cardinalities. Fix an arbitrary $\beta \geq \frac{\sqrt{17}+1}{4}$. The ε in the formulation of the theorem is solely to account for the fact that we require bids and cost shares to be rational.

Algorithm 2 (Computing a 2P-CSF for subadditive costs).
Input: subadditive cost function $c : [n]_0 \to \mathbb{Q}_{\geq 0}$
Output: valid 2P-CSF $(n, \boldsymbol{h}, \boldsymbol{l}, \boldsymbol{d})$

$l_0 := l_1 := \beta \cdot c(1)$; $h_0 := h_1 := \infty$; $d_0 := d_1 := 0$; $f := 1$
for $i := 2, \dots, n$ **do**
  if $\beta \cdot \frac{c(i)}{i} \leq l_f$ **then** $l_i := \beta \cdot \frac{c(i)}{i}$; $h_i := \infty$; $d_i := 0$; $f := i$
  else
    $l_i := l_{i-1}$; $h_i := \min\{\beta \cdot c(i) - (i-1) \cdot l_i, h_{i-1}\}$
    if $h_i + (i-1) \cdot l_i < c(i)$ **then**
      $d_i := 2$
    else if $h_i + (i-1) \cdot l_i \geq 2 \cdot c(f)$ **then**
      $d_i := 1$
    else if $h_i \geq (\beta^2 - \beta) \cdot c(f)$ **and** $i \geq 4$ **then**
      $d_i := 1$
      if $(\beta^2 - \beta) \cdot c(f) + (i-1) \cdot l_i \geq c(i)$ **then** $h_i := (\beta^2 - \beta) \cdot c(f)$
    else
      $d_i := 0$; $h_i := \infty$

Here, "∞" is a placeholder for any "sufficiently large" value (a value strictly larger than $\beta \cdot c(f)$ is sufficient) to simplify the presentation. Let $(n, \boldsymbol{h}, \boldsymbol{l}, \boldsymbol{d})$ be the output 2P-CSF. Since $d_0 = 0$ and $\gamma(0) = c(0)$, consider an arbitrary cardinality $i \in [n]$. We first show validity. Clearly, $d_i \in \{0, 1, 2\}$.

- $l_i \leq l_{i-1}$ and $(l_i < l_{i-1} \Longrightarrow d_i = 0)$ since $l_i \neq l_{i-1}$ only if l_i was set in line 3.
- Moreover, line 5 ensures $d_i > 0 \Longrightarrow h_i \leq h_{i-1}$.
- $d_i \leq d_{i-1} + 1$ and $(d_i = 2 \Longrightarrow h_i = h_{i-1})$. Clearly, we only need to consider $d_i = 2$: Then $h_i < c(i) - (i-1) \cdot l_i$ because line 6 evaluated to true and thus $h_i = \min\{\beta \cdot c(i) - (i-1) \cdot l_i, h_{i-1}\} = h_{i-1}$. Now assume $d_i > d_{i-1} + 1$, i.e., $d_{i-1} = 0$. Then, $h_i = h_{i-1} = \infty$, a contradiction to $h_i < c(i) - (i-1) \cdot l_i$.

We now prove β-BB. If $d_i = 1$, $c(i) \leq \gamma(i) \leq \beta \cdot c(i)$ due to lines 5 and 6. For $d_i \in \{0, 2\}$, we define $f := \max\{j \in [i] \mid \beta \cdot \frac{c(j)}{j} = l_i\}$ to be the last cardinality previous or equal to i for which the lower cost share was set in lines 1 or 3. Furthermore, let $g := \min(\{j \in \{i+1, i+2, \ldots, n\} \mid \beta \cdot \frac{c(j)}{j} \leq l\} \cup \{n+1\})$ be the next such cardinality after i (or $g = n+1$ if f is the largest such cardinality). It is $f \leq i < g \leq 2f$. Otherwise, f would not be maximal, due to $\frac{c(2f)}{2f} \leq \frac{2 \cdot c(f)}{2f} = \frac{c(f)}{f}$ because of subadditivity. Since c is non-decreasing, $c(i) \leq c(2f) \leq 2 \cdot c(f)$. Set $h'_i := \min\{\beta \cdot c(i) - (i-1) \cdot l_i, h_{i-1}\}$. We will make use of the following property:

$$d_{i-1} = 1 \text{ and } \gamma(i-1) \geq 2 \cdot c(f) \Longrightarrow \forall j \in \{i, i+1, \ldots, g-1\} : d_j = 1\,. \qquad (1)$$

Proof of (1): If $h'_i = h_{i-1}$, then $h'_i + (i-1) \cdot l_i = \gamma(i-1) + l_i$. If $h'_i = \beta \cdot c(i) - (i-1) \cdot l_i$, then $h'_i + (i-1) \cdot l_i = \beta \cdot c(i) \geq \beta \cdot c(i-1) \geq \gamma(i-1)$. In any case, $h'_i + (i-1) \cdot l_i \geq \gamma(i-1) \geq 2 \cdot c(f) \geq c(i)$, so $h_i = h'_i$ and $d_i = 1$ by line 8. Inductively, (1) follows. Consider now $d_i = 2$ and $d_i = 0$:

For $d_i = 2$, we first show $h_i = (\beta^2 - \beta) \cdot c(f)$. Define $i' := \max\{j \in [i] \mid d_j = 1\}$. By validity, $f < i' < i$ and $h_{i'} = h_i$. Since line 8 evaluated to false for cardinality i' because of (1), line 10 must have evaluated to true, implying $h_i = h'_i \geq (\beta^2 - \beta) \cdot c(f)$. Now assume "$>$". Let $s := \max\{j \in [i] \mid d_i = 0\}$ be the start of the segment that i is in. By validity, $f \leq s \leq i - 2$, $d_{s+1} = 1$, and $h_{s+1} \geq h_i$. Lines 8 and 12 evaluated to false for cardinality $(s+1)$, meaning that $\beta^2 \cdot c(f) \leq (\beta^2 - \beta) \cdot c(f) + s \cdot l_{s+1} < c(s+1)$. Then, however, $h_{s+1} + s \cdot l_{s+1} = \beta \cdot c(s+1) > \beta^3 \cdot c(f) > 2 \cdot c(f)$, a contradiction. Hence,

$$\gamma(i) = 2h_i + (i-2) \cdot \beta \cdot \frac{c(f)}{f} \geq (2\beta^2 - \beta) \cdot c(f) \geq 2 \cdot c(f) \geq c(i)\,.$$

On the other hand, $\beta \cdot c(f) < \beta^2 \cdot c(f) = h_i + \beta \cdot c(f) < h_i + (i-1) \cdot l_i < c(i)$, where the last inequality holds since line 6 evaluated to true. Hence,

$$\gamma(i) = 2h_i + (i-2) \cdot l_i < h_i + c(i) < (\beta - 1) \cdot c(i) + c(i) = \beta \cdot c(i)\,.$$

Now let $d_i = 0$. Since line 10 evaluated to false, $h'_i < (\beta^2 - \beta) \cdot c(f)$. We first show that $h'_i = \beta \cdot c(i) - (i-1) \cdot l_i$. Assume otherwise. Then, $h'_i = h_{i-1}$ and

$d_{i-1} = 1$ since $h_{i-1} \notin \{\infty, (\beta^2 - \beta) \cdot c(f)\}$. Yet, line 8 evaluated to false for $(i-1)$ because of (1). Thus, $h'_i = h_{i-1} \geq (\beta^2 - \beta) \cdot c(f)$ by line 10. Contradiction.

Now $\beta \cdot c(i) < (\beta - 1) \cdot \beta \cdot c(f) + (i-1) \cdot l_i$. Furthermore, $\beta \cdot c(f) \leq \gamma(i)$ and $(i-1) \cdot l_i \leq \gamma(i)$. Putting everything together gives

$$c(i) = \frac{\beta \cdot c(i)}{\beta} \leq \frac{(\beta - 1) \cdot \gamma(i) + \gamma(i)}{\beta} = \gamma(i)\,.$$

Moreover, $\gamma(i) = i \cdot l_i \leq \beta \cdot c(i)$ as otherwise $l_f = l_i > \beta \cdot \frac{c(i)}{i}$. □

Theorem 3. *For all $\varepsilon > 0$, there is a non-decreasing, symmetric, and subadditive cost function $c : [n]_0 \to \mathbb{Q}_{\geq 0}$ for which no valid $\left(\frac{\sqrt{17}+1}{4} - \varepsilon\right)$-BB 2P-CSF exists.*

Proof (Sketch). Fix $\alpha := \frac{\sqrt{17}+1}{4}$. W.l.o.g., let $0 < \varepsilon \leq \alpha - 1$ and set $\beta := \alpha - \varepsilon$. Finally, let $k, l \in \mathbb{N}$ with $l > \ln_\alpha \frac{\alpha - 1}{\varepsilon}$ and $k > \frac{(l+1)\cdot\beta}{\varepsilon} = \frac{(l+1)\cdot\alpha}{\varepsilon} - (l+1)$. Set $m := k + l + 1$ and $n := m + 1$ and consider the cost function $c : [n]_0 \to \mathbb{Q}_{\geq 0}$:

i	$1 \cdots k$	$k+1$	$k+2$	$\cdots$	$k+l$	m	n
$c(i)$	$1 \cdots 1$	$\alpha - \frac{\alpha-1}{\alpha^1}$	$\alpha - \frac{\alpha-1}{\alpha^2}$	$\cdots$	$\alpha - \frac{\alpha-1}{\alpha^l}$	α	2

Clearly, c is subadditive. Now assume there is a valid 2P-CSF $(n, \boldsymbol{h}, \boldsymbol{l}, \boldsymbol{d})$ which is β-BB. It can be shown that $d_m \geq 1$ and $d_n = d_m + 1$. Let $s := \max\{j \in [n] \mid d_j = 0\}$ be the start of the segment that cardinality n is in. Clearly, $s < m$ and $h_n \leq h_{s+1}$. By case analysis, $h_{s+1} \leq \beta \cdot c(s+1) - c(s) < \alpha^2 - \alpha$. Thus $\gamma(n) \leq h_n + \beta \cdot \alpha < 2 \cdot \alpha^2 - \alpha = 2 = c(n)$, a contradiction to β-BB. □

2.3 Applications to Scheduling

We apply our technique to the scheduling problem of minimizing makespan on related machines. To keep the service provider's task computationally tractable, we want algorithms to be polynomial-time in the size of the scheduling instance plus the players' bids. An instance is given by a tuple $(n, m, \boldsymbol{p}, \boldsymbol{s})$, where $n \in \mathbb{N}$ is the number of players, $m \in \mathbb{N}$ the number of machines, $\boldsymbol{s} \in \mathbb{Q}^m_{>0}$ a vector containing the machine speeds, and $\boldsymbol{p} \in \mathbb{Q}^n_{>0}$ a vector with the processing times. For identical jobs, computing the (symmetric) optimal makespan cost function $c : [n]_0 \to \mathbb{Q}_{\geq 0}$ in time $O(n \cdot \log m)$ is straightforward using LPT [5]. If jobs are not identical, an optimal schedule is in general NP-hard to compute and costs are not symmetric any more. Let d be the number of different processing times. We treat each processing time separately: First compute the costs $c : [n]_0 \to \mathbb{Q}_{\geq 0}$ of an optimal schedule when all processing times are 1. Second, compute the corresponding 2P-CSF F in time $O(n)$. Finally, for each processing time $\phi \in \bigcup\{p_i\}$, compute $MechCSF_F$ (though cost shares from F multiplied with ϕ) with just the bids of all players i with $p_i = \phi$. Each run with n_j players takes $O(n_j^2)$ time. Return the union of all selected players. Since $\forall (n_1, \ldots, n_d) \in [n]_0^d$ with $\sum n_j = n$: $\sum n_j^2 \leq n^2$, this algorithm is quadratic-time, GSP, and $(\frac{\sqrt{17}+1}{4} \cdot d)$-BB both w.r.t. the actual *and* the optimal scheduling cost:

Lemma 2. *For sharing the makespan cost on related machines, there is always a quadratic-time computable, GSP, and* $\left(\frac{\sqrt{17}+1}{4} \cdot d\right)$*-BB cost-sharing mechanism where* d *is the number of different processing times.*

In particular, dividing all cost shares by $\min_{i \in [n]}\{\frac{\gamma(i)}{c(i)}\}$ leads to:

Lemma 3. *For sharing the makespan cost of identical jobs on identical machines there is always a quadratic-time, GSP, and* 1*-BB mechanism.*

Extending our techniques in order to achieve 1-BB for non-symmetric costs seems to be very challenging. For 3 identical machines and processing times either 1 or 2, we manage to construct 1-BB and GSP mechanisms by generalizing the notion of a preference order and making cost shares dependent on the rank as well as the cardinalities of *both* classes of served players. Unfortunately, this approach is not directly extendable to more than three machines.

Theorem 4. *For sharing the makespan cost on 3 identical machines and jobs with processing times* 1 *or* 2*, there is always a GSP and* 1*-BB mechanism.*

Finally, we find it interesting that the costs used for Theorem 2 are optimal makespan costs for $n = m + 1$ identical jobs and speed $\frac{1}{c(i)}$ for machine $i \in [m]$.

3 The Impact of Symmetric Costs on GSP and 1-BB

Theorem 5. *There is no GSP mechanism that is 1-BB w.r.t.* $c : [4]_0 \to \mathbb{Q}_{\geq 0}$ *with* $c(1) := 1, c(2) := 3, c(3) := 6,$ *and* $c(4) := 7$.

Theorem 6. *For any cost function* $c : [3]_0 \to \mathbb{Q}_{\geq 0}$*, there is a 1-BB and GSP mechanism.*

Proof. In the following, we describe how the techniques from Section 2 can be reused to construct an algorithm for computing GSP and 1-BB mechanisms, given any arbitrary symmetric 3-player cost function $c : [3]_0 \to \mathbb{Q}_{\geq 0}$. It turns out that only the case where marginal costs are strictly increasing needs special treatment, i.e., the case $c(3) - c(2) > c(2) - c(1) > c(1)$.

First, for $i \in [3]$, vectors $\boldsymbol{\xi}^i \in \mathbb{Q}^i_{>0}$ are computed, where $\boldsymbol{\xi}^1 := c(1)$ and

$$\boldsymbol{\xi}^2 := \begin{cases} (\frac{c(2)}{2}, \frac{c(2)}{2}) & \text{if } \frac{c(2)}{2} \leq \xi^1_1 = c(1) \\ (c(2) - c(1); \boldsymbol{\xi}^1) & \text{otherwise} \end{cases}$$

$$\boldsymbol{\xi}^3 := \begin{cases} (\frac{c(3)}{3}, \frac{c(3)}{3}, \frac{c(3)}{3}) & \text{if } \frac{c(3)}{3} \leq \xi^2_2 \\ (c(3) - 2 \cdot \xi^2_2, \xi^2_2, \xi^2_2) & \text{otherwise, if } c(3) - c(2) < \xi^2_2 \\ (\xi^2_1, c(3) - c(2), \xi^2_2) & \text{otherwise, if } c(3) - c(2) < \xi^2_1 \\ (c(3) - c(2); \boldsymbol{\xi}^2) & \text{otherwise.} \end{cases}$$

It is a simple observation that the defined cost-sharing vectors correspond to a valid 2P-CSF if at most two prices are used for each cardinality. Hence, we can call Algorithm 1 as a subroutine. In the following, we consider the only remaining case $\xi^3_1 > \xi^3_2 > \xi^3_3$. We distinguish:

– $\xi_1^3 = \xi_1^2$, i.e., $\xi_1^3 = c(2) - c(1) > \xi_2^3 = c(3) - c(2) > \xi_3^3 = c(1)$:
  Here, we can actually reuse Algorithm 1 as well, by replacing references to the cost-sharing form by their equivalent in terms of cost-sharing vectors:
  - Replace line 1 by "$l := c(1)$"
  - In line 6, replace "$d_i = |D|$" with "$\xi^i_{|D|+1} = l$".
  - In line 11, replace $h_{|Q|}$ with $\xi^{|Q|}_{|D|+1}$.

  With these modifications, the computed mechanism is GSP: Assume true valuations $\boldsymbol{v} \in \mathbb{Q}^3$ and that there is a player $i \in [3]$ who improved for bid vector $\boldsymbol{b} \in \mathbb{Q}^3$. This can only happen if ($i \notin Q(\boldsymbol{v})$ or $x_i(\boldsymbol{v}) > c(1)$) and $\max Q(\boldsymbol{b}) = i$. Hence, there is a more preferred player j (i.e., $j > i$) with $b_j \leq c(1) < v_j$. Thus, j is part of the coalition and $u_j(\boldsymbol{b}) < u_j(\boldsymbol{v})$.

– $\xi_1^3 > \xi_1^2$, i.e., $\xi_1^3 = c(3) - c(2) > \xi_2^3 = c(2) - c(1) > \xi_3^3 = c(1)$:
  Here, including indifferent players never helps. Also, including less preferred players does not help more preferred players. Hence, we can simply go through the players in the order of preference and accept them if this gives them a strictly positive utility and reject them otherwise:

  $Q := \emptyset$
  for $i := 3, 2, 1$ **do**
      if $b_i > \xi_1^{|Q|+1}$ **then** $Q := Q \cup \{i\}$

  This is GSP: Assume true valuations $\boldsymbol{v} \in \mathbb{Q}^3$ and that there is a player $i \in [3]$ who improved for bid vector $\boldsymbol{b} \in \mathbb{Q}^3$. This can only happen if some player $j > i$ bids $b_j < v_i$ such that $j \notin Q(\boldsymbol{b})$ but $j \in Q(\boldsymbol{v})$. Then, however, j is part of the coalition and $u_j(\boldsymbol{b}) < u_j(\boldsymbol{v})$. □

4 Conclusion and Future Work

We regard as the main asset of our work that it is a systematic first step for finding GSP mechanisms that perform better than Moulin mechanisms. Furthermore, we continued the line of characterization efforts by specifically looking at symmetric costs. It came as a surprise that despite their simplicity, these costs do not necessarily allow for GSP and 1-BB mechanisms. While symmetric costs are arguably of limited practical interest, we yet transferred our techniques to the minimum makespan scheduling problem as an application and also to a setting with non-symmetric costs. Clearly, our work has to leave open many issues:

– For symmetric and/or subadditive costs, we still need an exact characterization with respect to the *best* possible BB that GSP mechanisms can achieve.
– Can our techniques be generalized to all/most non-symmetric cost functions? What is the potential of Algorithm 1? What is achievable with more prices? How would more general cost-sharing forms have to be like?
– Finally: Does the better BB (compared to Moulin mechanisms) come at the price of increased social cost (which was not considered in this work)?

Acknowledgment. We would like to thank Marios Mavronicolas for many fruitful discussions. Furthermore, we thank the anonymous referee who provided us with thorough feedback and helpful suggestions.

References

1. Archer, A., Feigenbaum, J., Krishnamurthy, A., Sami, R.: Approximation and collusion in multicast cost sharing. Games and Economic Behaviour **47** (2004) 36–71
2. Becchetti, L., Könemann, J., Leonardi, S., Pál, M.: Sharing the cost more efficiently: improved approximation for multicommodity rent-or-buy. In: Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms. (2005) 375–384
3. Bleischwitz, Y., Monien, B.: Fair cost-sharing methods for scheduling jobs on parallel machines. In: Proceedings of the 6th Italian Conference on Algorithms and Complexity. Volume 3998 of LNCS. (2006) 175–186
4. Brenner, J., Schäfer, G.: Cost sharing methods for makespan and completion time scheduling. In: Proceedings of the 24th International Symposium on Theoretical Aspects of Computer Science. Volume 4393 of LNCS. (2007)
5. Graham, R.: Bounds on multiprocessing timing anomalies. SIAM Journal of Applied Mathematics **17**(2) (1969) 416–429
6. Gupta, A., Könemann, J., Leonardi, S., Ravi, R., Schäfer, G.: An efficient cost-sharing mechanism for the prize-collecting Steiner forest problem. In: Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms. (2007)
7. Gupta, A., Srinivasan, A., Tardos, E.: Cost-sharing mechanisms for network design. In: Proceedings of the 7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems. Volume 3122 of LNCS. (2004) 139–152
8. Immorlica, N., Mahdian, M., Mirrokni, V.: Limitations of cross-monotonic cost sharing schemes. In: Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms. (2005) 602–611
9. Könemann, J., Leonardi, S., Schäfer, G.: A group-strategyproof mechanism for Steiner forests. In: Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms. (2005) 612–619
10. Könemann, J., Leonardi, S., Schäfer, G., van Zwam, S.: From primal-dual to cost shares and back: A stronger LP relaxation for the Steiner forest problem. In: Proceedings of the 32th Int. Colloquium on Automata, Languages, and Programming. Volume 3580 of LNCS. (2005) 930–942
11. Leonardi, S., Schäfer, G.: Cross-monotonic cost-sharing methods for connected facility location games. In: Proceedings of the ACM Conference on Electronic Commerce. (2004) 224–243
12. Mehta, A., Roughgarden, T., Sundararajan, M.: Beyond Moulin mechanisms. To appear in the Proceedings of the 8th ACM Conference on Electronic Commerce (2007)
13. Moulin, H.: Incremental cost sharing: Characterization by coalition strategy-proofness. Social Choice and Welfare **16**(2) (1999) 279–320
14. Penna, P., Ventre, C.: The algorithmic structure of group strategyproof budget-balanced cost-sharing mechanisms. In: Proceedings of the 23rd International Symposium on Theoretical Aspects of Computer Science. Volume 3884 of LNCS. (2006) 337–348