Online Incremental Cost Sharing Mechanisms*

Janina Brenner[1] and Guido Schäfer[2]

[1] Institute of Mathematics, Technical University Berlin, Germany
brenner@math.tu-berlin.de
[2] Centrum Wiskunde & Informatica, Amsterdam, The Netherlands
g.schaefer@cwi.nl

Abstract. We propose an online model for general demand cost sharing games and identify critical properties for group-strategyproofness and weak group-strategyproofness of cost sharing mechanisms for these games. We define *incremental* online cost sharing mechanisms which can be derived from competitive algorithms. Based on our general results, we develop online cost sharing mechanisms for several binary demand and general demand cost sharing games derived from network design and scheduling problems. Our results complement the work on incremental mechanisms by Moulin.

1 Introduction

The pivotal point of mechanism design is to achieve a global goal even though part of the input information is owned by selfish players. In cost sharing, the aim is to share the cost of a common infrastructure in a fair manner while the players' valuations for the service are private information. Based on bids and request types, a cost sharing mechanism has to determine a service allocation and distribute the incurred cost among the served players. A mechanism is called *strategyproof* if it provokes all players to reveal their true valuations, assuming that they act strategically. It is *group-strategyproof* if this is the case even when players can form coalitions to arrange their strategies (precise definitions are given in Section 2).

During the last ten years, there has been substantial research on *binary demand* cost sharing games, where a service allocation determines simply whether or not a player is served (see e.g. [1, 4, 9, 11, 12]). We consider the *general demand* setting in which players require not only one but several *levels* of service, e.g. several distinct connections to a source, multiple executions of some job, or different qualities of service. This setting has been studied by Moulin [11], Roughgarden et al. [10], and Bleischwitz et al. [2].

To the best of our knowledge, all previous studies of cooperative cost sharing games considered *offline* settings where the entire instance is known in advance. However, several natural cost sharing games are inherently online in the sense

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

that players arrive over time and request different levels of service. In this paper, we define the first *online* model for general demand cost sharing games. In its most general form, we assume that every player arrives several times to request an additional service level. Upon the arrival of a player, the online mechanism immediately determines a price for her new request. Once a player rejected the offered price for some service level she will not be offered a higher service level in the future. We require that at each point of time, the sum of the collected cost shares approximates the current cost.

In many cost sharing games, the common service is represented by a combinatorial optimization problem like minimum Steiner tree, machine scheduling, etc., which defines a cost for every possible service allocation. Our model is general enough to fit the online versions of most of these problems. Updates of the current solution are then restricted according to the online properties of the underlying problem, e.g. an online Steiner tree mechanism is usually not allowed to resell previously bought edges, and a scheduling mechanism cannot change processes that lie in the past.

We provide a simple yet effective method to derive online cost sharing mechanisms from competitive online algorithms [3]. Given an online optimization problem and a ρ-competitive algorithm for this problem, we define an *incremental* online mechanism for the corresponding cost sharing game which is ρ-budget balanced at all times. In the binary demand case, these mechanisms are automatically weakly group-strategyproof; in the general demand case, this is true if the marginal costs of the underlying cost function are increasing. Moreover, we prove necessary and sufficient conditions for group-strategyproofness of online cost sharing mechanisms.

Moulin's work [11] is closely related to our results. He defines incremental mechanisms for general demand cost sharing games in the offline setting. He characterizes group-strategyproofness of fully budget balanced cost sharing mechanisms in two cases depending on the underlying cost function. If the cost function is supermodular and marginal costs are increasing, essentially all group-strategyproof mechanisms are incremental mechanisms. On the other hand, if the cost function is submodular and marginal costs are decreasing, all cross-monotonic cost sharing methods for binary demand games yield group-strategyproof mechanisms, but only sequential stand alone mechanisms (a particularly simple subclass of incremental mechanisms) can be group-strategyproof for general demand cost sharing games.

We complement Moulin's results in the following way: First, our incremental mechanisms are defined slightly differently than his. In both cases, there is an order on all players, which in our case is predefined by the online order of arriving requests. Also, both define a player's cost share as the eponymous *incremental* cost caused by adding her service request to the current solution. However, in the borderline case in which a player's bid equals the offered price, our mechanism accepts the request while Moulin rejects it. This small difference causes incremental mechanisms for cost functions with increasing marginal costs to require complementary conditions for group-strategyproofness. While Moulin's incremental mechanisms require *supermodular* cost functions, ours need *submodular* cost functions to be group-strategyproof. Putting both results together, we can draw a larger picture: Every cost function with increasing marginal costs whose marginal costs depend monotonically on other players can be turned into a group-strategyproof incremental cost sharing mechanism (of either type).

Based on our general results, we develop online cost sharing mechanisms for several binary demand and general demand cost sharing games derived from network design and scheduling problems.

2 General Demand Cost Sharing

In this section, we formally introduce cost sharing games. In Section 2.1, we define *general demand* cost sharing games [2, 10, 11], a generalization of the well-studied binary demand case. In Section 2.2, we expand this framework to an online scenario.

2.1 The Offline Model

Let U be a set of players that are interested in a common service. In a *general demand* cost sharing game, every player has valuation for a finite number of service levels, i.e. the maximum service level requested is bounded by a constant $L \in \mathbb{N}$. Each player $i \in U$ has a *valuation* vector $v_i \in \mathbb{R}_+^L$, where $v_{i,l}$ denotes how much more (additive) player i likes service level l compared to service level $l-1$. The valuation vectors are private information, i.e. v_i is known to i only. Additionally, each player i announces a *bid* vector $b_i \in \mathbb{R}_+^L$. $b_{i,l}$ represents the maximum price player i is willing to pay for service level l (in addition to service level $l-1$).

An *allocation* of goods or service to the set of players U is denoted by a vector $\boldsymbol{x} \in \mathbb{N}_0^U$, where $x_i \in \mathbb{N}_0$ indicates the level of service that player i obtains; here $x_i = 0$ represents that i does not receive any good or service. Note that as a characteristic of this model, only subsequent service levels can be allocated to a player (i.e. if a player obtains service level l, then she also obtains service levels $1, \ldots, l-1$). We denote by $\boldsymbol{e}_i \in \mathbb{N}_0^U$ the ith unit vector.

The *servicing cost* of an allocation $\boldsymbol{x} \in \mathbb{N}_0^U$ is given by a cost function $C : \mathbb{N}_0^U \to \mathbb{R}_+$. We assume that C is non-decreasing in every component and $C(\mathbf{0}) = 0$ for the all-zero allocation $\mathbf{0}$. In the examples we study, the common service is represented by a combinatorial optimization problem like e.g. Steiner tree, machine scheduling, etc. In these cases, we will define $C(\boldsymbol{x})$ as the cost of an offline optimal solution to the underlying optimization problem.

A general demand *cost sharing mechanism* solicits the bid vectors b_i from all players $i \in U$, and computes a service allocation $\boldsymbol{x} \in \mathbb{N}_0^U$ and (non-negative) payments $\phi_{i,l} \in \mathbb{R}$ for every player $i \in U$ and service level $l \leq L$. We assume that the mechanism complies with the following standard assumptions:

1. *Individual rationality*: A player is charged only for service levels that she receives, and for any service level, her payment is at most her bid, i.e. for all i, l: $\phi_{i,l} = 0$ if $x_i < l$ and $\phi_{i,l} \leq b_{i,l}$ if $x_i \geq l$.

2. *No positive transfer*: A player is not paid for receiving service, i.e. $\phi_{i,l} \geq 0$ for all i, l.

For notational convenience, we define $v_{i,0} = \phi_{i,0} = 0$ for all players $i \in U$.

Let $\bar{C}(\boldsymbol{x})$ denote the cost of the actually computed solution for allocation $\boldsymbol{x}$. A cost sharing mechanism is *β-budget balanced* if the total payment obtained from all players deviates by at most a factor $\beta \geq 1$ from the total cost, i.e.

$$\bar{C}(\boldsymbol{x}) \leq \sum_{i \in U} \sum_{l=1}^{L} \phi_{i,l} \leq \beta \cdot C(\boldsymbol{x}).$$

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

We assume that players act strategically and every player's goal is to maximize her own utility. The *utility* of player i is defined as

$$u_i(\boldsymbol{x}, \boldsymbol{\phi}) := \sum_{l=1}^{x_i} (v_{i,l} - \phi_{i,l}).$$

Since the outcome $(\boldsymbol{x}, \boldsymbol{\phi})$ computed by the mechanism solely depends on the bids $\boldsymbol{b}$ of the players (and not on their true valuations), a player may have an incentive to declare a bid vector b_i that differs from her true valuation vector v_i. We say that a mechanism is *strategyproof* if bidding truthfully is a dominant strategy for every player. That is, for every player $i \in U$ and every two bid vectors $\boldsymbol{b}, \boldsymbol{b}'$ with $b_i = v_i$ and $b_j = b'_j$ for all $j \neq i$, we have $u_i(\boldsymbol{x}, \boldsymbol{\phi}) \geq u_i(\boldsymbol{x}', \boldsymbol{\phi}')$, where $(\boldsymbol{x}, \boldsymbol{\phi})$ and $(\boldsymbol{x}', \boldsymbol{\phi}')$ are the solutions output by the mechanism for bid vectors $\boldsymbol{b}$ and $\boldsymbol{b}'$, respectively. Note that in our model, a player cannot lie about the types or arrival times of her requests.

In *cooperative* mechanism design, it is assumed that players can form coalitions in order to coordinate their bids. A mechanism is *group-strategyproof* if no coordinated bidding of a coalition $S \subseteq U$ can ever strictly increase the utility of some player in S without strictly decreasing the utility of another player in S. More formally, for every coalition $S \subseteq U$ and every two bid vectors $\boldsymbol{b}, \boldsymbol{b}'$ with $b_i = v_i$ for every $i \in S$ and $b_i = b'_i$ for every $i \notin S$,

$$u_i(\boldsymbol{x}', \boldsymbol{\phi}') \geq u_i(\boldsymbol{x}, \boldsymbol{\phi}) \quad \forall i \in S \quad \Longrightarrow \quad u_i(\boldsymbol{x}', \boldsymbol{\phi}') = u_i(\boldsymbol{x}, \boldsymbol{\phi}) \quad \forall i \in S.$$

A mechanism is *weakly group-strategyproof* if no coordinated bidding can ever strictly increase the utility of *every* player in a coalition. That is, for every coalition $S \subseteq U$ and every two bid vectors $\boldsymbol{b}, \boldsymbol{b}'$ with $b_i = v_i$ for every $i \in S$ and $b_i = b'_i$ for every $i \notin S$,

$$\exists\, i \in S: \; u_i(\boldsymbol{x}', \boldsymbol{\phi}') \leq u_i(\boldsymbol{x}, \boldsymbol{\phi}).$$

Intuitively, weak group-strategyproofness suffices if we assume that players adopt a slightly more conservative attitude with respect to their willingness of joining a coalition: While in the group-strategyproof setting a player will participate in a coalition even if her utility is not affected, she only participates if she is strictly better off in the weakly group-strategyproof setting.

2.2 The Online Model

We now extend general demand cost sharing games into an online setting [3]. As mentioned before, most cost sharing games studied both in this paper and the existing literature are based on underlying combinatorial optimization problems. This motivates us to define online cost sharing games very generally with varying online characters inherited by the respective online combinatorial optimization problems. However, our mechanisms will always be required to fix the payment for a requested service at the point of time when it is revealed; the online solution can be modified as in the underlying online optimization problem.

Following Borodin et al. [3], we describe an online list model as a basic example. Note that for certain problems like online scheduling, jobs may have release dates which will then be treated as arrival times of the respective requests. Let (i, l) denote player i's request for service level l. In the basic list model, service requests arrive according to an online list. Upon arrival, the player reveals the type of her new request (the input information for the underlying combinatorial optimization problem) and her bid $b_{i,l}$. The mechanism must immediately decide whether to accept this request and at what price, without any knowledge about future requests. In addition, the mechanism has to maintain a (possibly suboptimal) feasible solution for the current service allocation. The space of feasible transitions will in general guarantee that the solution cost can only increase in subsequent steps.

Let $\boldsymbol{x}^t$ denote the current allocation after processing request $t \in T = \{1, 2, \ldots\}$. Let $\bar{C}(\boldsymbol{x}^t)$ denote the cost of the actually computed solution for $\boldsymbol{x}^t$. We call an online cost-sharing mechanism *β-budget balanced at all times* for some $\beta \geq 1$ if for all requests $t \in T$:

$$\bar{C}(\boldsymbol{x}^t) \leq \sum_{i \in U} \sum_{l=1}^{x_i^t} \phi_{i,l} \leq \beta \cdot C(\boldsymbol{x}^t).$$

The conditions of individual rationality and no positive transfer as well as the different forms of incentive compatibility transfer in a straightforward way.

3 Incremental Online Mechanisms

In this section, we define a simple but very general method for turning competitive online algorithms into *incremental* online cost sharing mechanisms. It is closely related to our results on singleton mechanisms in the offline setting [4]. Given a combinatorial optimization problem $\mathcal{P}$ and a ρ-competitive online algorithm ALG for this problem, we define an online mechanism for the corresponding cost sharing problem which is ρ-budget balanced at all times. This mechanism is weakly group-strategyproof if the algorithm's marginal costs are increasing, which is gratuitous in the binary demand case. We also extract conditions on ALG under which the derived mechanism fulfills the stronger property of group-strategyproofness.

3.1 Definition

Let ALG be a ρ-competitive algorithm for an online combinatorial optimization problem $\mathcal{P}$. Consider an instance $\mathcal{I}$ of $\mathcal{P}$. The incremental online algorithm induced by ALG works as follows: Requests arrive according to $\mathcal{I}$. Each time a new request arrives, we simulate ALG on the online instance induced by the requests that have previously been accepted plus the new item. The player who made the new request is offered the additional level of service for the incremental price caused by the update in the competitive algorithm. If her bid $b_{i,l}$ is larger or equal to this price, the request is accepted and added to the current online instance. Otherwise, the request is rejected and all further appearances of player i are deleted from the online list. Algorithm 1 gives a more formal description.

Algorithm 1: Incremental online mechanism induced by ALG.

Input: ρ-competitive online algorithm ALG for $\mathcal{P}$
Output: allocation vector $\boldsymbol{x} = (x_i)_{i \in U}$, payment vector $\boldsymbol{\phi} = (\phi_{i,l})_{i \in U, l \leq L}$

Initialize $\boldsymbol{x}^0 = \mathbf{0}$
forall *requests* $t \in T$ **do**
  Read out type and bid $b_{i,l}$ of newly arrived request (i, l).
  Simulate ALG to compute $p = \bar{C}(\boldsymbol{x}^{t-1} + \boldsymbol{e}_i) - \bar{C}(\boldsymbol{x}^{t-1})$.
  if $b_{i,l} \geq p$ **then** set $\boldsymbol{x}^t = \boldsymbol{x}^{t-1} + \boldsymbol{e}_i$ and $\phi_{i,l} = p$
  else set $\boldsymbol{x}^t = \boldsymbol{x}^{t-1}$ and $\phi_{i,l} = 0$, and ignore all further appearances of player i.
end
Output allocation vector $\boldsymbol{x}$ and payments $\boldsymbol{\phi}$

3.2 Budget Balance

It is straightforward to see that the budget balance factor of the incremental online mechanism is inherited from the competitive ratio of the input algorithm:

Lemma 1. *The incremental online mechanism is ρ-budget balanced at all times.*

Proof. In every iteration t of the mechanism, we have $\sum_{i \in U} \sum_{l=1}^{x_i^t} \phi_{i,l} = \bar{C}(\boldsymbol{x}^t)$, since every accepted player pays exactly the incremental cost for adding her to the current set of served players. Since ALG is a ρ-competitive algorithm, we obtain

$$\bar{C}(\boldsymbol{x}^t) = \sum_{i \in U} \sum_{l=1}^{x_i^t} \phi_{i,l} \leq \rho \cdot C(\boldsymbol{x}^t),$$

which proves ρ-budget balance at all times. □

4 Incentive Compatibility

The following characterizations are true not only for incremental online mechanisms but for all online mechanisms that accept requests if their announced bid is greater *or equal* to the offered price.

4.1 Strategyproofness

In order to achieve strategyproofness, we have to bound the increase in marginal valuations of individual players. As expressed by Fact 1, this is an important precondition to prevent players from overbidding for a lower level to obtain positive utility for a higher one. In previous works on general demand cost sharing [2, 10], players' valuations were assumed to be non-increasing. However, we can slightly relax this condition by introducing a positive factor λ:

Definition 1. *A valuation vector* $v_i \in \mathbb{R}^L$ *is* λ-decreasing *if for all* $1 \le l \le L$,

$$v_{i,l} \le \lambda \cdot v_{i,l-1}.$$

Given λ-decreasing valuations for all players, the incremental mechanism is guaranteed to be weakly group-strategyproof if and only if the induced cost shares grow faster than the valuations (the proof is given in the Section 4.2):

Definition 2. *A cost sharing mechanism has* λ-increasing prices *if for every bid vector* $\boldsymbol{b}$ *and player* $i \in U$*, the price for any service level* $1 \le l \le L$ *is at least* λ *times the price for the previous service level, i.e.*

$$\phi_{i,l}(\boldsymbol{b}) \ge \lambda \cdot \phi_{i,l-1}(\boldsymbol{b}).$$

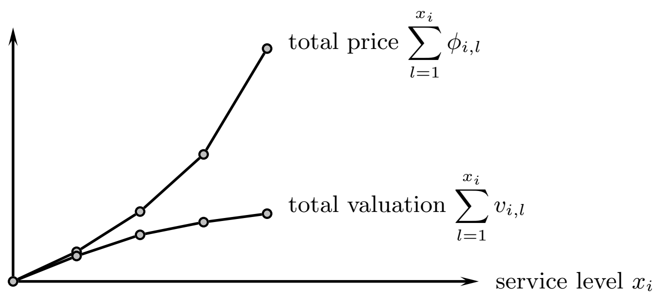

Fig. 1. An example for λ-decreasing valuations and λ-increasing prices with $\lambda = 1$

We would like to remark that the above conditions can be further generalized by letting λ vary for every player (and/or level) or adding constant terms to the right hand sides. However, the following fact emphasizes that a set of conditions similar to the above are indeed necessary to achieve strategyproofness.

Fact 1 *A general demand online mechanism is not strategyproof if cost shares do not increase faster than valuations in terms of service levels.*

Proof. Assume that $\phi_{i,l+1}(\boldsymbol{b}) = \lambda \cdot (\phi_{i,l}(\boldsymbol{b}) - \delta)$ for some player i and service level l, and $\delta > 0$. By consumer souvereignity we have $x_i(\boldsymbol{b}) \ge l$. We define player i's valuations as $v_{i,k} := \phi_{i,k}(\boldsymbol{b})$ for $k < l$, $v_{i,l} := \phi_{i,l}(\boldsymbol{b}) - \epsilon$ and $v_{i,l+1} = \lambda \cdot v_{i,l}$.

With this valuation vector, player i obtains positive utility in the run on $\boldsymbol{b}$ for the combination of service levels l and $l+1$: $u_{i,l}(\boldsymbol{b}) + u_{i,l+1}(\boldsymbol{b}) = (\phi_{i,l}(\boldsymbol{b}) - \epsilon) - \phi_{i,l}(\boldsymbol{b}) + \lambda \cdot (\phi_{i,l}(\boldsymbol{b}) - \epsilon) - \lambda \cdot (\phi_{i,l}(\boldsymbol{b}) - \delta) = \lambda\delta - (1+\lambda)\epsilon > 0$ for sufficiently small ϵ. On the other hand, player i would get zero total utility if she bid truthfully. $\square$

4.2 Weak Group-Strategyproofness

We now prove that under the above conditions, every online incremental mechanism is weakly group-strategyproof.

Theorem 1. *With λ-decreasing valuations and λ-increasing prices, an online cost sharing mechanism is weakly group-strategyproof.*

Proof. Fix a coalition $S \subseteq U$ and a bid vector $\boldsymbol{b}$ with $b_i = v_i$ for all $i \in S$. Assume for contradiction that all members of the coalition can strictly increase their utilities by changing their bids to $\boldsymbol{b'}$ (while $b_i = b'_i$ for all $i \notin S$). Let (i,l) be the first request for which the mechanism makes different decisions in the runs on $\boldsymbol{b}$ and $\boldsymbol{b'}$. We have $\phi_{i,l}(\boldsymbol{b}) = \phi_{i,l}(\boldsymbol{b'})$ since all previous decisions of the mechanism were equal in both runs. There are two possible cases:

1. $v_{i,l} < \phi_{i,l} \leq b'_{i,l}$. Because of λ-decreasing valuations and λ-increasing prices, we have $\ldots \leq \lambda^{-2} v_{i,l+2} \leq \lambda^{-1} v_{i,l+1} \leq v_{i,l} < \phi_{i,l}(\boldsymbol{b'}) \leq \lambda^{-1}\phi_{i,l+1}(\boldsymbol{b'}) \leq \lambda^{-2}\phi_{i,l+2}(\boldsymbol{b'}) \leq \ldots$, and hence player i has negative utility for service levels l and higher in the run on $\boldsymbol{b'}$, whereas the utility for each level is non-negative when bidding truthfully.
2. $b'_{i,l} < \phi_{i,l} \leq v_{i,l}$. Then, player i gets only $l-1$ levels of service in the run on $\boldsymbol{b'}$, whereas she gets additional utility by accepting level l in the run on $\boldsymbol{b}$.

Consequently, player i gets less or equal utility in the run on $\boldsymbol{b'}$, a contradiction to the assumption. □

Remark that for binary demand cost sharing games, both Definitions 1 and 2 are always fulfilled since there is only one level of service. Hence, binary demand incremental cost sharing mechanisms are always weakly group-strategyproof.

4.3 Group-Strategyproofness

In order to ensure the stronger notion of group-strategyproofness, we need to prevent that *dropping out*, like in the second case in the proof of Theorem 1, can help subsequent players. Towards this end, we need the following property of cross-monotonicity, which is equivalent to Moulin's submodular costs condition.

Consider a fixed instance of an online optimization problem $\mathcal{P}$ and let $\phi_{i,l}(\boldsymbol{b})$ denote the price that player i is offered for service level l when $\boldsymbol{b}$ is the bid vector input to the mechanism. Throughout this section, we assume λ-decreasing valuations and λ-increasing prices.

Definition 3. *An online mechanism is* cross-monotonic *if for every player $i \in U$ and service level l, the offered price does not decrease when a subset of requests are accepted in previous iterations, i.e.*

$$\phi_{i,l}(\boldsymbol{b'}) \geq \phi_{i,l}(\boldsymbol{b})$$

for all bid vectors $\boldsymbol{b}, \boldsymbol{b'}$ such that $x^{t-1}(\boldsymbol{b'}) \leq x^{t-1}(\boldsymbol{b})$, where (i,l) is request t.

We call an online algorithm ALG cross-monotonic *if the induced incremental online mechanism is cross-monotonic.*

With this property, the online incremental cost sharing mechanism is group-strategyproof; the main ideas for the proof are: First, dropping out can never help others since it only increases cost shares of subsequent bidders. Second, the first member of a coalition who overbids for an additional level of service can only decrease her utility by doing this, since prices increase more than valuations in terms of service levels.

Theorem 2. *An online mechanism is group-strategyproof if it is cross-monotonic.*

Proof. Fix a coalition $S \subseteq U$ and a bid vector $\boldsymbol{b}$ with $b_i = v_i$ for all $i \in S$. Assume that every member of the coalition increases or maintains her utility when the coalition changes their bids to $\boldsymbol{b'}$ (while $b_i = b'_i$ for all $i \notin S$).

We first prove that $\boldsymbol{x}^t(\boldsymbol{b'}) \leq \boldsymbol{x}^t(\boldsymbol{b})$ for all $t \in T$. Assume for contradiction that there is a request which is accepted in the run on $\boldsymbol{b'}$ but not in the run on $\boldsymbol{b}$. Let (i, l) be the earliest such request, say request t. That is, $\boldsymbol{x}^\tau(\boldsymbol{b'}) \leq \boldsymbol{x}^\tau(\boldsymbol{b})$ for all $\tau < t$. By cross-monotonicity of ALG, we have $\phi_{i,l}(\boldsymbol{b'}) \geq \phi_{i,l}(\boldsymbol{b})$. Since players outside the coalition submit the same bids in both runs, player i must be a member of the coalition to gain service in the run on $\boldsymbol{b'}$. But then, $\phi_{i,l}(\boldsymbol{b'}) \geq \phi_{i,l}(\boldsymbol{b}) > b_{i,l} = v_{i,l}$ and hence by λ-decreasing valuations and λ-increasing prices, player i has negative utility for service levels l and higher in the run on $\boldsymbol{b'}$. Since $\boldsymbol{x}^\tau(\boldsymbol{b'}) \leq \boldsymbol{x}^\tau(\boldsymbol{b})$ for all $\tau < t$, by cross-monotonicity $\phi_{i,k}(\boldsymbol{b'}) \geq \phi_{i,k}(\boldsymbol{b})$ for all $k < l$ as well, and therefore $u_i(\boldsymbol{b'}) < u_i(\boldsymbol{b})$, a contradiction to the first assumption.

We can conclude that $\boldsymbol{x}^t(\boldsymbol{b'}) \leq \boldsymbol{x}^t(\boldsymbol{b})$ for all $t \in T$. Hence, $\phi_{i,l}(\boldsymbol{b'}) \geq \phi_{i,l}(\boldsymbol{b})$ for all i, l by cross-monotonicity. This means that

$$u_i(\boldsymbol{b'}) = \sum_{l=1}^{x_i(\boldsymbol{b'})} (v_{i,l} - \phi_{i,l}(\boldsymbol{b'})) \leq \sum_{l=1}^{x_i(\boldsymbol{b})} (v_{i,l} - \phi_{i,l}(\boldsymbol{b})) = u_i(\boldsymbol{b})$$

for all i and l, hence we obtain group-strategyproofness. □

We prove next that the statement in Theorem 2 actually holds in an if and only if fashion, even in the binary demand case.

Theorem 3. *An online mechanism is not group-strategyproof if it is not cross-monotonic.*

Proof. Consider an online mechanism that is not cross-monotonic; let $L = 1$. That is, there are bid vectors $\boldsymbol{b}, \boldsymbol{b'}$ with $\boldsymbol{x}^{t-1}(\boldsymbol{b'}) \leq \boldsymbol{x}^{t-1}(\boldsymbol{b})$ and $\phi_i(\boldsymbol{b'}) < \phi_i(\boldsymbol{b})$ for some player i. For simplicity, assume that i is the last player in the online instance. We will define valuations such that there is a coalition S which has an incentive to misreport their valuations.

Define $S := \{j \in U \mid b_j \neq b'_j\} \cup \{i\}$. Assume that all $j \in U \setminus S$ bid $b_j = b'_j$. Now, define $v_j := \phi_j(\boldsymbol{b})$ for all $j \in S$. Note that if all players in S bid truthfully, the outcome of the incremental mechanism is the same as for the bid vector $\boldsymbol{b}$. However, if the coalition changes their bids to $\boldsymbol{b'}$, all players $j \in S \setminus \{i\}$ lose service but retain their previous utility of zero, while player i increases her utility from zero to $\phi_i(\boldsymbol{b}) - \phi_i(\boldsymbol{b'})$. Hence, the mechanism is not group-strategyproof. □

5 Binary Demand Examples

We now apply our framework to competitive online algorithms for several combinatorial optimization problems. In this section, we give examples for *binary demand* cost sharing games, i.e. the maximum service level is $L = 1$ and every player has only one request.

Online Scheduling. Consider the parallel machine scheduling problem with the objective of minimizing the makespan. Any list scheduling algorithm has an approximation factor of at most 2 for this problem. Hence, the online algorithm that adds each arriving job to the machine with the currently least load is 2-competitive. Unfortunately, it is not cross-monotonic as deleting jobs can cause higher or lower completion times for subsequent jobs. Nonetheless, our framework leads to a 2-budget balanced, weakly group-strategyproof online mechanism. Note that in this scenario, jobs do not have release dates and so the online order is not coupled with scheduling time.

Corollary 1. *There is a 2-budget balanced weakly group-strategyproof incremental online mechanism for the minimum makespan scheduling problem on parallel machines* $P||C_{\max}$.

Online Steiner Tree and Forest. Given an undirected graph G with edge costs, connection requests arrive online. In the Steiner forest problem, each request consists of a pair of vertices s_i, t_i; in the Steiner tree problem, all requests have one vertex in common, i.e. $s_i = s_j$ for all $i, j \in U$. The goal is to select a minimum cost set of edges such that each vertex pair is connected by a path.

The online greedy Steiner tree algorithm picks the shortest path to the current tree each time a new commodity arrives. It has a competitive ratio of $\log n$, while the competitive ratio of any online algorithm is shown to be at least $1/2 \log n$ [8]. If additional commodities cannot cause players to switch to a different path, e.g. if G is a tree, dropping out cannot help subsequent players, and the algorithm is cross-monotonic. In the general case, our framework gives a weakly group-strategyproof $\Theta(\log n)$-budget balanced online cost sharing mechanism for the Steiner tree problem, which is asymptotically best possible. The respective greedy algorithm for the online Steiner forest problem achieves an approximation ratio of $O(\log^2 n)$.

Corollary 2. *There is an* $O(\log^2 n)$*-budget balanced weakly group-strategyproof incremental online mechanism for the Steiner forest problem. This mechanism is* $(\log n)$*-budget balanced for the Steiner tree problem. Both mechanisms are group-strategyproof if the input graph is a forest.*

6 Multiple Demand Examples

In this section, we exploit the whole range of our framework by deriving incremental mechanisms for *general demand* cost sharing games. In the first example, players are assumed to arrive only once with the complete list of their requests, while in the second example, the arrival sequence is mixed, i.e. players can takes turns announcing additional requests.

Online Preemptive Scheduling. A common problem in preemptive scheduling is the parallel machine setting in which each job has a release date. The cost of a solution is given by the sum of all completion times. The single machine case is solved optimally by the *shortest remaining processing time* (SRPT) algorithm [13]. SRPT is a 2-approximation algorithm for the parallel machine case [5]. In the corresponding scheduling game, we treat the release date of a job as its arrival time in the cost sharing framework. The cost $\bar{C}$ of an allocation $\boldsymbol{x}$ is the total cost of the SRPT schedule for $\boldsymbol{x}$. Whenever a job arrives, the SRPT solution is updated by adding the new job, and the resulting increase in total cost is set to be the cost share for this request.

In our model, each player may request multiple executions of their job. E.g. consider a student who asks a copy shop to print and bind several copies of his master's thesis, or a joinery is asked to produce a few of the same individual piece of furniture. In these scenarios, it is very natural to assume that the marginal valuation for each additional copy is decreasing, i.e. $v_{i,l} \geq v_{i,l+1}$ for all i, l. We assume that each player arrives only once with all her requests. In scheduling terms, each player owns several jobs which all have the same release date and processing time. Before subsequent players arrive, the SRPT algorithm schedules all of player i's jobs subsequently, hence each of them delays the same number of jobs, and later copies have larger completion times. Therefore, the general demand incremental online mechanism induced by SRPT has increasing marginal prices, and we obtain:

Corollary 3. *There is a 2-budget balanced weakly group-strategyproof general demand incremental online mechanism for the preemptive scheduling problem with release dates* $P|r_i, pmtn|\sum C_i$*. This mechanism is budget balanced in the single machine case.*

Online Multicommodity Routing. In an online multicommodity routing problem, we are given a directed graph with monotonically increasing cost functions on each arc. Commodities arrive online and request routing of l units of capacity from some vertex to another. We assume that the routing is splittable in integer units. The greedy algorithm which routes each unit of flow separately in an optimal way is $(3+2\sqrt{2})$-competitive for this problem [6, 7]. It is clear that marginal costs are increasing, since the cost functions on each arc grow with increasing traffic. This is true even when players arrive in a mixed order and request to route additional units between their source-destination pair. However, this is a congestion-type game (the more players in the game, the higher the costs per request), and so we cannot expect group-strategyproofness.

Corollary 4. *There is a* $(3+2\sqrt{2})$*-budget balanced weakly group-strategyproof incremental online mechanism for the online multicommodity routing problem in which each player arrives multiple times.*

7 Conclusion

We characterized strategyproofness, weak group-strategyproofness and group-strategyproofness of online cost sharing mechanisms which are directly derived from competitive online mechanisms by employing incremental cost sharing. While for binary demand problems, all such mechanisms are directly weakly group-strategyproof, cost shares for subsequent service levels must increase faster than valuations in the general demand case to achieve this property. In both cases, incremental mechanisms are group-strategyproof only if *dropping out* of previous players cannot help subsequent players. As a consequence, we cannot expect incremental cost sharing mechanisms for scheduling games or other problems with congestion effects to be group-strategyproof, whereas this is easier to achieve for network design problems.

References

1. 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 *LNCS*, pages 175–186, 2006.
2. Y. Bleischwitz and F. Schoppmann. Group-strategyproof cost sharing for metric fault tolerant facility location. In *Proc. of the 1st Int. Sympos. on Algorithmic Game Theory*, volume 4997 of *LNCS*, pages 350–361, 2008.
3. A. Borodin and R. El-Yaniv. *Online computation and competitive analysis*. Cambridge University Press, New York, NY, USA, 1998.
4. J. Brenner and G. Schäfer. Singleton acyclic mechanisms and their applications to scheduling problems. In *Proc. of the 1st Int. Sympos. on Algorithmic Game Theory*, volume 4997 of *LNCS*, pages 315–326, 2008.
5. J. Du, J. Y. T. Leung, and G. H. Young. Minimizing mean flow time with release time constraint. *Theoretical Computer Sci.*, 75(3):347–355, 1990.
6. T. Harks, S. Heinz, and M. Pfetsch. Competitive online multicommodity routing. *Theory of Computing Systems*, 2008.
7. T. Harks, S. Heinz, M. Pfetsch, and T. Vredeveld. Competitive online multicommodity routing with time windows. In *Unpublished*, 2007.
8. M. Imase and B. Waxman. Dynamic Steiner tree problems. *SIAM J. Disc. Math.*, 4(3):369–384, 1991.
9. N. Immorlica, M. Mahdian, and V. S. Mirrokni. Limitations of cross-monotonic cost sharing schemes. In *Proc. of the 16th ACM-SIAM Sympos. on Discrete Algorithms*, pages 602–611, 2005.
10. A. Mehta, T. Roughgarden, and M. Sundararajan. Beyond Moulin mechanisms. In *Proc. of the ACM Conference on Electronic Commerce*, 2007.
11. H. Moulin. Incremental cost sharing: Characterization by coalition strategyproofness. *Soc. Choice Welfare*, 16:279–320, 1999.
12. T. Roughgarden and M. Sundararajan. New trade-offs in cost-sharing mechanisms. In *Proc. of the 38th ACM Sympos. on Theory of Computing*, pages 79–88, 2006.
13. L. Schrage. A proof of the optimality of the shortest remaining processing time discipline. *Operations Res.*, 16:687–690, 1968.