Incentives for Early Arrival in Cost Sharing

Junyu Zhang
ShanghaiTech University
Shanghai, China

Yao Zhang
Kyushu University
Fukuoka, Japan

Yaoxin Ge
ShanghaiTech University
Shanghai, China

Dengji Zhao*
ShanghaiTech University
Shanghai, China

Hu Fu
Shanghai University of Finance and Economics
Shanghai, China

Zhihao Gavin Tang
Shanghai University of Finance and Economics
Shanghai, China

Pinyan Lu
Shanghai University of Finance and Economics
Shanghai, China

ABSTRACT

In cooperative games, we study how values created or costs incurred by a coalition are shared among the members within it, and the players may join the coalition in a online manner such as investors invest a startup. Recently, Ge *et al.* [10] proposed a new property called incentives for early arrival (I4EA) in such games, which says that the online allocation of values or costs should incentivize agents to join early in order to prevent mutual strategic waiting. Ideally, the allocation should also be fair, so that agents arriving in an order uniformly at random should expect to get/pay their Shapley values. Ge *et al.* [10] showed that not all monotone value functions admit such mechanisms in online value sharing games.

In this work, we show a sharp contrast in online *cost* sharing games. We construct a mechanism with all the properties mentioned above, for every monotone cost function. To achieve this, we first solve 0-1 valued cost sharing games with a novel mechanism called *Shapley-fair shuffle cost sharing mechanism (SFS-CS)*, and then extend SFS-CS to a family called *generalized Shapley-fair shuffle cost sharing mechanisms (GSFS-CS)*. The critical technique we invented here is a mapping from one arrival order to another order so that we can directly apply marginal cost allocation on the shuffled orders to satisfy the properties. Finally, we solve general valued cost functions, by decomposing them into 0-1 valued functions in an online fashion.

KEYWORDS

Cost Sharing; Early Arrival; Online Cooperative Games

ACM Reference Format:
Junyu Zhang, Yao Zhang, Yaoxin Ge, Dengji Zhao*, Hu Fu, Zhihao Gavin Tang, and Pinyan Lu. 2025. Incentives for Early Arrival in Cost Sharing. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

1 INTRODUCTION

In a classical cost sharing game, a fixed group of players receive a service as a coalition, and the cost of the service is divided among the players. This game models many real-world applications such as electricity or water supply networks [1, 11, 21]. They have been well studied and many solutions have been proposed to address different requirements [3, 13, 20]; e.g. players should have incentives to stay together to share the cost [22], players' shares should be fair [18], or they should change consistently as members join or leave the coalition [12, 19].

Most traditional cost sharing games distribute the cost offline, i.e., the mechanism knows all the players and their cost function in advance. However, in many applications, players typically do not all arrive at the same time; rather, they join sequentially. Upon the arrival of a new player, an irrevocable decision has to be made on the division of the current cost, without knowledge about players that arrive in the future.

For example, a shirt factory produces shirts for different customers with different brands and their orders do not arrive at the same time. However, the unit cost per shirt may decrease when more shirts are produced at one period (considering the cost of assembling a new product line). On the other hand, when new orders arrive, the factory cannot just wait forever to get more orders to start. Therefore, the existing customers have to bear the production costs. If so, the unit cost of early arrival orders might be higher than the late arrival orders. This will incentivize customers to strategically wait each other. From the factory point of view, we need to incentivize customers to place orders as soon as they need, so that the factory can fully utilize its capacity.

Except for the *incentives for early arrivals* (I4EA), the solution also needs to satisfy some other properties. The cost share to a player should be non-increasing when more players are joining (*online individual rationality* (OIR)). One intuition for OIR is that a player made his decision according to the cost allocation on his arrival, if the cost is going to increase with uncertainty in the future, the player may not even want to join the coalition in the first place. Finally, the cost share should be considered fair, otherwise, allocating all the costs to the last comer trivially satisfies I4EA and

*Corresponding author, zhaodj@shanghaitech.edu.cn

OIR. As a measure of fairness, we require that, if the players arrive in a uniformly random order, each player's expected cost share is his Shapley value (*Shapley fairness* (SF)).

When the cost function is supermodular, letting each newly joined player pay her marginal cost satisfies the three properties — OIR, I4EA, and SF; but for more general cost functions, it is not clear that a mechanism exists with all three properties. In fact, for *value* sharing games, Ge *et al.* [10] showed that not all value functions admit such online mechanisms.

In sharp contrast, in this work we show that in online cost sharing games, every monotone cost function admits an online mechanism that is I4EA, OIR, and SF.

Specifically, we propose a new mechanism called the *Shapley-fair shuffle cost sharing mechanism (SFS-CS)* to solve **all** 0-1 cost sharing games. SFS-CS is further extended to a class called the *generalized Shapley-fair shuffle cost sharing mechanism (GSFS-CS).* The key technique used in GSFS-CS is shuffling the original arriving order into a new order which essentially helps us decide who should pay the cost. The shuffling is the key to guaranteeing Shapley fairness, which is also quite involved.

In summary, our main contributions advance the state of the art as follows:

- For 0-1 valued monotone online cost sharing games, we propose a mechanism called *Shapley-fair shuffle cost sharing mechanism (SFS-CS)* to satisfy OIR, I4EA, and SF.
- Extending SFS-CS, we propose a class of mechanisms (GSFS-CS), which gives more flexibility on allocating the cost.
- For general online cost sharing games, we propose decomposing a general cost function into 0-1 valued cost functions to utilize GSFS-CS to satisfy the same properties.

The remainder of the paper is organized as follows. Section 1.1 introduces the related work and Section 2 defines the model and all the desirable properties. We then propose SFS-CS to solve all the 0-1 monotone cost sharing games in Section 3 and extend it to a general class (GSFS-CS) in Section 4. Finally, we extend the solution to general monotone cost sharing games in Section 5.

1.1 Other Related Work

Many studies have already considered online cost sharing. For example, online multicast cost sharing, where players arrive one by one and each connects to the root by greedily choosing a path minimizing its cost, was studied in [7]. It focuses on the price of anarchy of the Nash equilibrium. Besides, cost sharing games with private valuations in an online setting was studied in [5] and they give a perfect characterization of both weakly group-strategyproof and group-strategyproof online cost sharing mechanisms for this model. Furthermore, some studies considered the application of online cost sharing for demand-responsive transport systems and horizontal supply chains [9, 27]. None of them has considered the incentives for early arrival.

There are many online mechanism design problems on other topics. For example, the online coalition formation game is studied in [6, 8], where the players with preference for different coalitions arrive online. The main objective is to effectively allocate players who join asynchronously into groups, with the overarching goal of maximizing the collective social welfare. Besides, [4, 16, 17] studied the auction mechanism design in dynamic settings, where players with private valuations of items will arrive or change over time. Additionally, mechanism design with diffusion incentives is also a new trend [14, 15, 24–26]. This area of research focuses on incentivizing people to invite their neighbors in a social network to participate in an auction or a collaboration. We consider a different setting for cost sharing, where the players can decide when to arrive. To improve time efficiency, the goal is to guarantee that the players are benefited from early arrival.

Many studies also considered the cost sharing problem of minimizing the cost in the network [2, 13, 20]. They consider the minimum spanning tree problem and aim to allocate the cost of the spanning tree among the players. Furthermore, cost sharing under private valuation and connection control was studied in [23], where players have private valuation about the service and the connection control of the edge. They all consider the offline setting while we focus on the online cost sharing.

2 THE MODEL

An online cost sharing game is given by a triple (N, c, π), where N is a set of players, $c : 2^N \to \mathbb{R}_{\geq 0}$ is a cost function, and $\pi \in \Pi(N)$ is a permutation of N ($\Pi(N)$ denotes the set of all permutations of N). Players arrive sequentially, in the order given by π. For a coalition $S \subseteq N$, $c(S)$ is the cost caused by S. c is *normalized* if $c(\emptyset) = 0$, and is *monotone* if $\forall T \subseteq S \subseteq N, c(S) \geq c(T)$. Throughout this work, we consider normalized and monotone cost functions. The following gives some formal notations and definitions.

- Given an order π, we say $j \prec_\pi i$ if player j arrives strictly earlier than player i. All these players and i herself form a set $p(i, \pi)$, i.e., $p(i, \pi) := \{j \mid j \prec_\pi i\} \cup \{i\}$.
- For a subset $S \subseteq N$, c *restricted to* S, written as $c_{|S}$, is the set function $c_{|S} : 2^S \to \mathbb{R}_{\geq 0}$ defined by $c_{|S}(T) = c(T), \forall T \subseteq S$; π *restricted to* S, written as $\pi_{|S}$, is the permutation of S defined by $i \prec_{\pi_{|S}} j$ iff $i \prec_\pi j$, for all $i, j \in S$.
- A subset $S \subseteq N$ is a *prefix* of π if S is the set of first $|S|$ players to arrive according to π. We denote this as $S \sqsubseteq \pi$.
- For an online cost sharing game (N, c, π) and a prefix $S \sqsubseteq \pi$, the *local cost sharing game on* S is the game $(S, c_{|S}, \pi_{|S})$.

Definition 2.1 (Marginal Cost). Given a cost function c, player i's **marginal cost (MC)** to a coalition $S \ni i$ is

$$\mathtt{MC}(i, c, S) := c(S) - c(S \setminus \{i\}).$$

Definition 2.2 (Shapley Value). Given a cost function c, player i's **Shapley value** is

$$\mathtt{SV}_i(c) := \frac{1}{|N|!} \sum_{\pi \in \Pi(N)} \mathtt{MC}(i, c, p(i, \pi)) .$$

For a monotone cost function, the marginal cost of any player in any coalition is non-negative. Hence, the Shapley value is also non-negative.

For a cost sharing game to be online, we would like that, at any point of time, when the set of players that have arrived is S, the cost caused by S should be allocated irrevocably among the players in S, and this allocation should be conducted using only information on c and π restricted to S. The next definitions formalize this.

Definition 2.3. A **cost sharing policy** ϕ maps a cost sharing game (N, c, π) to an $|N|$-tuple of allocations, so that $\phi_i(N, c, \pi) \geq 0$ is player i's cost share, and $\sum_i \phi_i(N, c, \pi) = c(N)$.

An **online cost sharing mechanism** is given by a cost sharing policy ϕ for each stage of the game, so that after the arrival of each prefix $S \sqsubseteq \pi$, each player $i \in S$ gets a share of $\phi_i(S, c_{|S}, \pi_{|S})$.

When more players join, we expect that the additional costs caused by them should not be shared by the players arriving before them. In other words, we require each player's cost share to weakly decrease as more players arrive.

Definition 2.4. An online cost sharing mechanism ϕ is **online individually rational** (OIR) for cost function c if, for any arrival order π and any $T, S \sqsubseteq \pi$ with $T \subseteq S$, we have $\phi_i(T, c_{|T}, \pi_{|T}) \geq \phi_i(S, c_{|S}, \pi_{|S})$ for every player $i \in T$.

In fact, OIR is an online variant of cross-monotonic [12]. Both require when new agents join a coalition, the cost share of the existing agents are non-increasing.

To prevent players from strategically waiting, we would like that a player's cost share should weakly increase if she chooses to unilaterally delay her arrival. Formally,

Definition 2.5. An online cost sharing mechanism ϕ is **incentivizing for early arrival** (I4EA) if for any player i, $\phi_i(N, c, \pi_1) \leq \phi_i(N, c, \pi_2)$ for all π_1 and π_2 such that $\pi_{1|N\setminus\{i\}} = \pi_{2|N\setminus\{i\}}$ and $p(i, \pi_1) \subset p(i, \pi_2)$.

Proposition 2.6. *An online cost sharing mechanism ϕ is I4EA if and only if for any player i, $\phi_i(N, c, \pi_1) \leq \phi_i(N, c, \pi_2)$ for any two arrival orders π_1 and π_2 with only player i being delayed one position, i.e., $\pi_1 = [\dots, i, j, \dots]$ and $\pi_2 = [\dots, j, i, \dots]$.*

We would like an online cost sharing mechanism to be fair. To this end, we require a player's expected cost share to be equal to her Shapley value if the arrival order is uniformly at random.

Definition 2.7. An online cost sharing mechanism ϕ is **Shapley-fair** (SF) for a cost function c if for each player $i \in N$,

$$\frac{1}{|N|!} \sum_{\pi \in \Pi(N)} \phi_i(N, c, \pi) = \mathsf{SV}_i(c).$$

In this work, we aim to design online cost sharing mechanisms that are OIR, I4EA, and SF.

3 0-1 VALUED COST FUNCTIONS

In this section, we study cost functions that take values only 0 or 1. Section 3.1 introduces the definition of the shuffle rule and shuffle-based cost sharing mechanism. We then propose our mechanism in Section 3.2 and show its desirable properties in Section 3.3.

3.1 Shuffle-Based Cost Sharing Mechanisms

For 0-1 valued cost functions, for any arrival order π, there is at most one player whose arrival makes the current coalition's cost jump from 0 to 1. We call this player the *marginal player*.

Definition 3.1. Given a 0-1 valued cost function c and an order π, player $i \in \pi$ is called the **marginal player** if $\mathsf{MC}(i, c, p(i, \pi)) = 1$.

As a first attempt, we consider a cost sharing game where $N = \{A, B, C\}$ and $c(S) = 1$ if and only if $A \in S$ or $\{B, C\} \subseteq S$. A possible allocation that satisfies I4EA, OIR, and SF is illustrated in Figure 1. For each order, we first shuffle the order and apply marginal cost allocation in the shuffled order. For example, we shuffle $[A, B, C]$ to $[B, C, A]$, in which C is the marginal player and bears the cost.

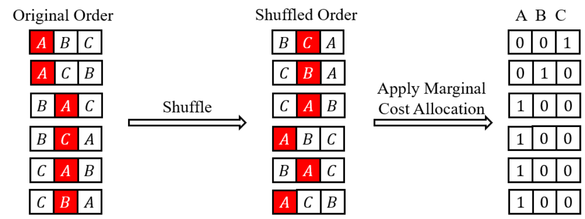

Figure 1: The player in red is the marginal player.

Adding a layer of sophistication to this idea, one may extend it to a family of mechanisms that retain its Shapley fairness: given a permutation π, if we use a *bijection* to map π to another permutation π', and let each player in π pay her marginal cost in π', the resulting mechanism must be SF. The idea is natural enough, but for it to be useful, one has to be able to calculate the bijection in an online fashion; on top of that, the bijection must satisfy other properties for the resulting mechanism to be OIR and I4EA. Our main result is that, somewhat surprisingly, a desirable mechanism based on such a bijection exists and can be computed efficiently. We first define bijections usable in such mechanisms:

Definition 3.2. Given a player set N and cost function c, a **shuffle rule** is a function $\mathsf{shuf} : \bigcup_{S \subseteq N} \Pi(S) \to \bigcup_{S \subseteq N} \Pi(S)$, satisfying

(a) for each $S \subseteq N$, shuf restricted to $\Pi(S)$ is a bijection from $\Pi(S)$ to $\Pi(S)$, and this bijection depends only on $c_{|S}$; and
(b) for any $T \subseteq N$, $\pi \in \Pi(T)$, and $S \sqsubseteq \pi$, $\mathsf{shuf}(\pi_{|S}) = (\mathsf{shuf}(\pi))_{|S}$. Algebraically, $\mathsf{shuf}(\cdot)$ and projection to any prefix commutes.

Given this definition of shuffle rule, the following is a well defined online mechanism:
A **shuffle-based cost sharing mechanism** ϕ is given by a shuffle rule shuf, so that after the arrival of each prefix $S \sqsubseteq \pi$, each player $i \in S$ gets a share $\phi_i(S, c_{|S}, \pi_{|S}) = \mathsf{MC}(i, c_{|S}, p(i, \mathsf{shuf}(\pi_{|S})))$.

Note that since shuf is a bijection on each $\Pi(S)$ for $S \subseteq N$, shuf is a bijection on $\bigcup_{S \subseteq N} \Pi(S)$. Its inverse shuf^{-1} exists with $\mathsf{shuf}^{-1}(\mathsf{shuf}(\pi)) = \pi$ for any $\pi \in \bigcup_{S \subseteq N} \Pi(S)$.

Proposition 3.3. *A shuffle-based cost sharing mechanism is SF.*

Proof. Write $\pi' = \mathsf{shuf}(\pi)$. Since shuf is bijective, we have

$$\frac{1}{|N|!} \sum_{\pi \in \Pi(N)} \phi_i(N, c, \pi) = \frac{1}{|N|!} \sum_{\pi' \in \Pi(N)} \mathsf{MC}(i, c, p(i, \pi'))$$
$$= \frac{1}{|N|!} \sum_{\pi \in \Pi(N)} \mathsf{MC}(i, c, p(i, \pi)) = \mathsf{SV}_i(c).$$

□

Definition 3.4 (Group Size Monotone). For a 0-1 valued cost function c, a shuffle rule shuf is **group size monotone** if for any $S \subseteq N$, any $\pi \in \Pi(S)$ with player i being the marginal player in $\mathtt{shuf}(\pi)$, it is the case that for any $T \sqsubseteq \pi$ with $i \in T$, i is also the marginal player in $\mathtt{shuf}(\pi_{|T})$.

Proposition 3.5. *A shuffle-based cost sharing mechanism is OIR if its shuffle rule* shuf *is group size monotone.*

Proof. We only need to show that, for any arrival order $\pi \in \Pi(N)$ and any player i, if i is not the marginal player in the image ordering at a certain point, she never becomes the marginal player in the image ordering as more players join. Suppose for prefix $T \sqsubseteq \pi$, i is not the marginal player in $\mathtt{shuf}(\pi_{|T})$. Then as more players join, when the set of players is $S \supseteq T$, i cannot be the marginal player in $\mathtt{shuf}(\pi_{|S})$ either, by the group size monotonicity of shuf. □

Definition 3.6 (Flip Monotone). For a 0-1 valued cost function, a shuffle rule shuf is **flip monotone** if for any two arrival orders with two adjacent players flipped, i.e., $\pi_1 = [\dots, j, i, \dots]$ and $\pi_2 = [\dots, i, j, \dots]$, if player i is the marginal player in $\mathtt{shuf}(\pi_2)$, then i is the marginal player in $\mathtt{shuf}(\pi_1)$ as well.

By Proposition 2.6, the following is immediate.

Proposition 3.7. *A shuffle-based cost sharing mechanism is I4EA if its shuffle rule* shuf *is flip monotone.*

Theorem 3.8. *A shuffle-based cost sharing mechanism is SF, OIR, and I4EA if its shuffle rule* shuf *is group size monotone and flip monotone.*

In Section 3.2, we construct a group size monotone and flip monotone shuffle rule sfs-shuf and the corresponding *Shapley-fair shuffle cost sharing mechanism* (SFS-CS).

3.2 Shapley-fair Shuffle Cost Sharing Mechanism

We first make some observations and then illustrate some attempts at a desirable shuffle rule, before giving our final construction.

Given an arrival order π, we refer to its image under shuf as the image ordering. When a new player i joins an existing coalition S, the relative orderings of the players in S in the image ordering must remain unchanged by definition of shuffle rule; therefore, in the image ordering $\mathtt{shuf}(\pi_{|S\cup\{i\}})$, i must be inserted into $\mathtt{shuf}(\pi_{|S})$. This suggests that a construction of any shuffle rule can go by stages; at the arrival of each new player, one only needs to decide where to insert i in the image ordering. We therefore maintain and grow an image ordering π' as players join; when there is no danger of confusion, we treat π' as a variable, and use it to denote $\pi'_{|S}$.

Intuitively, both group size monotonicity and flip monotonicity suggest that if a late comer i in π can be made a marginal player in the image ordering, i should be made so. We will indeed follow this intuition in our construction; in fact, among all the positions in the image ordering that make i the marginal player, we choose the earliest such position. The following example shows, however, that when a late comer cannot be made a marginal player in the image ordering, one needs to be very careful choosing her position, because the bijectivity of shuf may be at stake.

Example 3.9. Consider the 0-1 valued cost sharing game with $N = \{A, B, C\}$, and $c(S) = 1$ if and only if $S \supseteq \{A, B\}$. For the arrival order $\pi_1 = [A, B, C]$, when A arrives, $\pi'_1 = [A]$; then B arrives; since $c(\{A, B\}) = 1$, to satisfy OIR, π'_1 must be $[A, B]$. When C arrives, she is not a marginal player in π'_1 no matter where we insert her. Does it make a difference where we insert her?

- If we insert C to the start of π'_1, we get $\pi'_1 = [C, A, B]$ (see Figure 2). Consider another two orders, $\pi_2 = [A, C, B]$ and $\pi_3 = [C, A, B]$. Since shuf is a bijection from $\Pi(\{A, C\})$ to itself, after the second player joins, one of the image orderings of π_2 and π_3 is $[C, A]$, and the other is $[A, C]$. B is the last player in both π_2 and π_3, and by group size monotonicity must be made the marginal player in π'_2 and π'_3 when she joins. The image ordering $[C, A]$ must now become $[C, A, B]$, contradicting shuf being a bijection. Therefore π_1 cannot be mapped to $[C, A, B]$.
- By a similar argument, C cannot be inserted at the end of π'_1.

Therefore, π'_1 has to be $[A, C, B]$ when C joins.

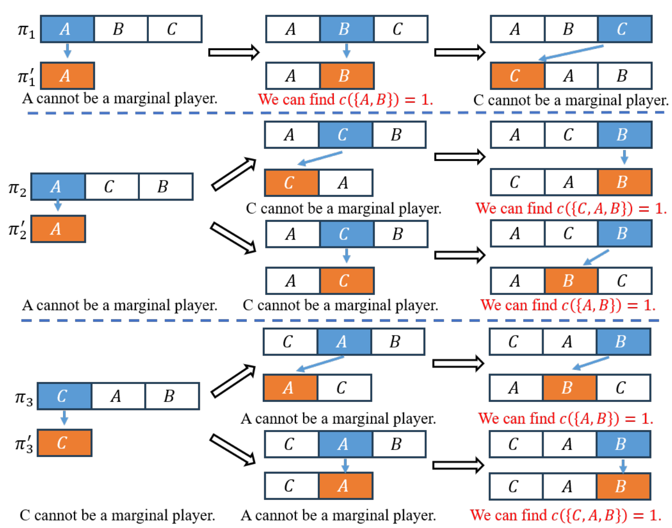

Figure 2: An attempt on constructing a shuf **for cost sharing game mentioned in Example 3.9.**

It turns out that in Example 3.9, there is a shuffle rule mapping π_1 to $[A, C, B]$ that has all the desirable properties. The construction of sfs-shuf below shows that this is not a coincidence. Specifically, when a newcomer i cannot be made the marginal player in the image ordering π', sfs-shuf inserts i in π' before *i's predecessor in π*. (In Example 3.9, the predecessor of C in π is B, hence C is placed before B in π'.)

We now formally construct sfs-shuf. The iterative construction explicitly gives the image under sfs-shuf of an order π restricted to each prefix $S \sqsubseteq \pi$. It should be clear that, in this mapping, sfs-shuf uses only $\pi_{|S}$ and $c_{|S}$, and does produce an order on S. The fact that it is bijective is nontrivial, and is shown in Section 3.3.

The shuffle rule `sfs-shuf` maps any order π to an image ordering π' given by the following iterative procedure:

- The image ordering π' is initialized to be the first player in π. New players arriving according to π are iteratively inserted into π'. Let i be the next player to arrive in π.
- **Case 1.** If i can be inserted into π' so that i becomes the marginal player in π', she is inserted into the earliest such position. Formally, let $\mathcal{P}$ be the set $\{S \mid S \sqsubseteq \pi', \mathrm{MC}(i, c, S \cup \{i\}) = 1\}$, then $\mathcal{P} \neq \emptyset$. Let S^* be the member in $\mathcal{P}$ with the smallest cardinality. Update π' so that i is inserted after S^*. (Note that S^* may be the empty set, in which case i becomes the first player in π'.)
- **Case 2.** If there is no way to insert i into π' to make her the marginal player, update π' so that i is inserted before her predecessor in π. (By predecessor we mean the player coming right before i in π.)

Definition 3.10. The *Shapley-fair shuffle cost sharing mechanism* (SFS-CS) is the shuffle-based cost sharing mechanism given by the shuffle rule `sfs-shuf`.

We show in Section 3.3 the proofs of the main theorem.

Theorem 3.11. *For all 0-1 valued monotone cost sharing games, SFS-CS is SF, OIR, and I4EA.*

Before the proof, we give an example of `sfs-shuf`.

Example 3.12. Consider the 0-1 valued cost sharing game with $N = \{A, B, C, D, E, F, G\}$. For any T, $c(T) = 1$ if and only if $\exists S \subseteq T, S \in \{\{A, C\}, \{B, C\}, \{B, D, E\}, \{E, F\}\}$ (see Figure-3). The left side is the original order π and for each joining player (colored by blue), the right side shows the construction process of the image ordering π'. Note that the players colored by red are the marginal players, and the players colored by green are the corresponding related players (see Definition 3.14 in Section 3.3). For the arrival order $\pi = [A, B, C, D, E, F, G]$, when A arrives, $\pi' = [A]$; then B arrives, $\mathcal{P} = \emptyset$. B is inserted before A in π' and $\pi' = [B, A]$. When C arrives, $\mathcal{P} = \{\{B\}, \{B, A\}\}$ and $\pi' = [B, C, A]$. Then D arrives and $\mathcal{P} = \emptyset$. So D is inserted before C in π', i.e., $\pi' = [B, D, C, A]$. The arrivals of E, F, and G are handled similarly. E is the marginal player in the final image ordering, so her share is 1 and everyone else's is 0.

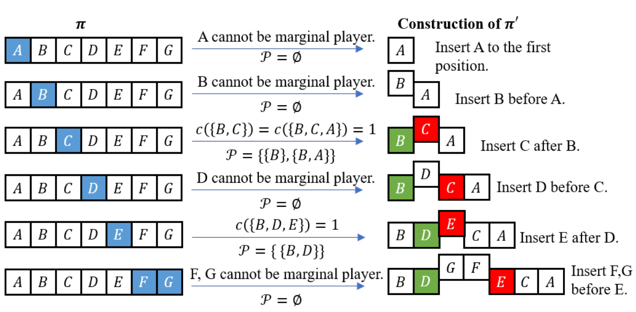

Figure 3: An example of `sfs-shuf`.

Then we prove Theorem 3.11 in Section 3.3.

3.3 Proofs of Properties of SFS-CS

3.3.1 OIR. By Proposition 3.5, it suffices to show that, if a player i is not the marginal player in the image ordering π' at a certain moment, then she does not become the marginal player in π' when a new player j joins. Obviously, the only case that needs an argument is when j is inserted into π', the cost of the new coalition is 1 and j is not the marginal player in π'; by definition of `sfs-shuf`, this happens only if j cannot be made the marginal player no matter where she is inserted in π'. For the sake of contradiction, suppose i becomes the marginal player in π' after j joins. This can happen only when j is inserted before i in π'. Let S be the prefix of π' up to i before j joins. If $c(S) = 1$, then $c(S \setminus \{i\}) = 1$ (since i was not the marginal player before j joins), and $c(S \setminus \{i\} \cup \{j\}) \geq c(S \setminus \{i\}) = 1$, and so i still cannot be the marginal player. Therefore, $c(S) = 0$. If i becomes the marginal player after j joins, we must have $c(S \cup \{j\}) = 1$. But that means j can be the marginal player if she is placed right after i, a contradiction to the condition that j cannot be a marginal player no matter where she is inserted. This shows that i cannot become the marginal player when a new player joins.

3.3.2 SF. By Proposition 3.3, it suffices to show that `sfs-shuf` is a bijection. We do this by constructing an inverse mapping for `sfs-shuf`. Given an ordering $\pi' \in \Pi(S)$, we show that a unique $\pi \in \Pi(S)$ can be found such that `sfs-shuf`$(\pi) = \pi'$. To this end, it suffices to show that we can uniquely identify the last player in π, which allows us to iteratively reconstruct π.

- If $c(S) = 0$, there cannot be a marginal player in π', in which case π is simply the reverse of π'.
- If $c(S) = 1$, let i be the marginal player in π'.
  - If $c(\{i\}) = 1$, then
    (a) Either i is the last comer in π, in which case i must be the first player in π';
    (b) Or i is not the last comer in π, but the marginal player in π' has not changed since i's arrival in π. In this case, the players before i in π' are precisely the players arriving after i in π, and their order in π' is precisely reverse to their order in π. Hence the first player in π' is the last player in π.
  - If $c(\{i\}) \neq 1$, we also consider the following two cases:
    (a') Either i is the last comer in π;
    (b') Or i is not the last comer, but the marginal player in π' has not changed since i's arrival in π.

In case (a) and (a'), by definition of `sfs-shuf`, i is in the earliest position in π' that makes her the marginal player. In particular, in case (a), i has no predecessor in π', and in case (a'), let j be the predecessor of i in π', we must have $c(p(j, \pi')) = 0$, and for any $k \prec_{\pi'} j$, $c(p(k, \pi') \cup \{i\}) = 0$.

In case (b) and (b'), let T be the set of players arriving after i in π, then players in T are inserted before i in π', in the order precisely reverse to $\pi_{|T}$. In case (b'), if we still let j be the predecessor of i in π' at the moment right after i's arrival in π (i.e., before the arrival of anyone in T), then we have $c(p(j, \pi') \cup \{i\}) = 1$, and the person right after j in π' is the last one to arrive in π.

To summarize, to distinguish case (a') and (b'), let $j \prec_{\pi'} i$ be the first player in π' with $c(p(j, \pi') \cup \{i\}) = 1$. If j is the predecessor of i in π', then i is the last one to arrive in π; otherwise, the player right after j is the last one to arrive in π. Therefore, we can always uniquely identify the last player in π, and iteratively reconstruct π from π'. Example 3.15 provides an illustration.

Two notions in the analysis of cases (a') and (b') are used again in the next section. We formally define them as follows.

Definition 3.13. Given π' with the marginal player i and $c(\{i\}) = 0$, i's predecessor in π' right after i's insertion into π' is said to be i's **related player**.

In the proof above, j is i's related player and she can be identified as the first player in π' before i such that $c(p(j, \pi') \cup \{i\}) = 1$.

Definition 3.14. Given π' with the marginal player i and $c(\{i\}) = 0$, i's predecessor in π' right after i's insertion into π' is said to be i's **related player**.

In the proof above, the set T is the late arriving set. This set can be identified in π' as the set of players before i if $c(\{i\}) = 1$, or, if $c(\{i\}) = 0$, as the set of players between i and her related player j. By this notation, we can simplify the reconstruction process by finding the players in late arriving set, reversing them, and putting them at the end of reconstruction order.

Example 3.15. Considering the game in Example 3.12, with a final image ordering $\pi' = [B, D, G, F, E, C, A]$, we give the procedure to recover the original order (see Figure 4). The left side marks the marginal players (red) and corresponding related players (green) through this process, and the right side shows the identified last players (blue) in each iteration. Initially, the marginal player is E and the related player is D, so that the player G right after D is the one who arrives last in π. Similarly, we can find the player F is the one who arrives last among the remaining players. Next, the player right after the related player D is player E herself, so that E is the last one arriving. Following steps are all similar and we can finally find the original order π.

3.3.3 I4EA. Before we prove I4EA, we introduce two lemmas. Lemma 3.16 states that if player i is the marginal player of the image ordering when she joins, all the players after her in the original order are inserted before her in the image ordering (see Figure 5).

Lemma 3.16. *Given an order π and image ordering π' of* `sfs-shuf`*, suppose player i is the marginal player of $\pi'_{|p(i,\pi)}$ (i.e., the image ordering when i just joins). $\forall j \in \pi$ satisfying $i \prec_\pi j$, we have $j \prec_{\pi'} i$.*

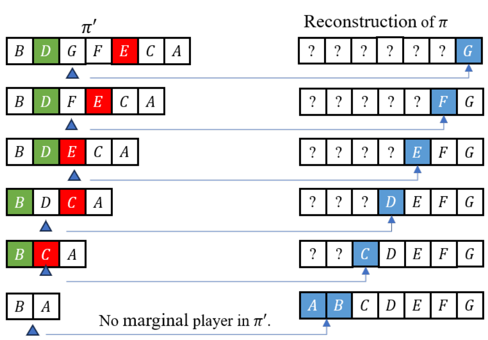

Figure 4: The reconstruction process of π from π'.

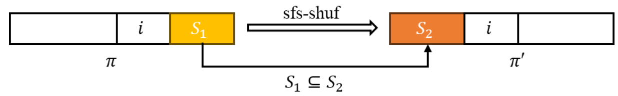

Figure 5: π is the original order. π' is the image ordering. i is the marginal player of π'. Lemma 3.16 states $S_1 \subseteq S_2$.

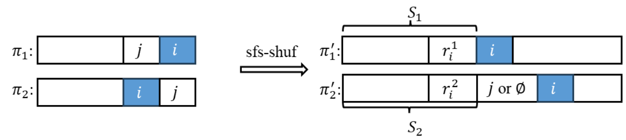

Figure 6: π_1, π_2 are the original orders, where only last two players flip. π'_1 and π'_2 are the corresponding image orderings. i is the marginal player of π'_1 and π'_2. r^1_i and r^2_i are the corresponding related players. Lemma 3.17 states $S_1 \setminus \{j\} \subseteq S_2$.

Proof. Consider player i's next player j, there are two cases.

- **Case 1.** If j cannot be the marginal player, she is inserted before her predecessor i of π, i.e., $j \prec_{\pi'} i$.
- **Case 2.** If j can be the marginal player, we prove the statement by contradiction. Assuming that $i \prec_{\pi'} j$, we have $c(p(j, \pi'_{|p(j,\pi)})) \geq c(p(i, \pi'_{|p(i,\pi)})) = 1$ and $c(p(j, \pi'_{|p(j,\pi)}) \setminus \{j\}) \geq c(p(i, \pi'_{|p(i,\pi)})) = 1$. Hence,

$$\begin{aligned}\texttt{MC}(j, c, p(j, \pi'_{|p(j,\pi)})) &= c(p(j, \pi'_{|p(j,\pi)})) - c(p(j, \pi'_{|p(j,\pi)}) \setminus \{j\}) \\ &= c(p(i, \pi'_{|p(i,\pi)})) - c(p(i, \pi'_{|p(i,\pi)})) \\ &= 1 - 1 = 0,\end{aligned}$$

which leads to a contradiction.

For j's next player k, (i) if j cannot be the marginal player, there are two cases.

- **Case 1.** If k cannot be the marginal player, she is inserted before her predecessor j of π, i.e., $k \prec_{\pi'} j \prec_{\pi'} i$.
- **Case 2.** If k can be the marginal player, the analysis is the same as the Case 2 above so that $k \prec_{\pi'} i$.

(ii) If j is the marginal player, the analysis of k is the same as the analysis of j above so that $k \prec_{\pi'} j \prec_{\pi'} i$.

For the following players, we can get the conclusion recursively. □

We can use Figure 6 to visually illustrate the insights of Lemma 3.17.

Lemma 3.17. *Given two orders $\pi_1 = [\dots, j, i]$, and $\pi_2 = [\dots, i, j]$, where only adjacent i and j exchange. π'_1 and π'_2 are the corresponding image orderings of* `sfs-shuf`*. If player i is the marginal player of both π'_1 and π'_2, and let r^1_i and r^2_i be the related players of π'_1 and π'_2, then we have $p(r^1_i, \pi'_1) \setminus \{j\} \subseteq p(r^2_i, \pi'_2)$.*

Proof. For player j, there are two cases.

- **Case 1.** $j \prec_{\pi'_1} r^2_i$. For π'_1, when i is inserted after r^2_i, i is the marginal player. Hence, i cannot be inserted in the later position and we can get $r^1_i \prec_{\pi'_1} r^2_i$ or $r^1_i = r^2_i$. Therefore, $p(r^1_i, \pi'_1) \setminus \{j\} \subseteq p(r^2_i, \pi'_2)$.

- **Case 2.** $r_i^2 \prec_{\pi_1'} j$. When i joins for π_1, `sfs-shuf` must insert i after r_i^2 such that i is the marginal player. Hence, $r_i^1 = r_i^2$ and thus $p(r_i^1, \pi_1') \setminus \{j\} \subseteq p(r_i^2, \pi_2')$.

□

Now we prove I4EA with above Lemmas and Proposition 3.7.

Proof of I4EA. Given $\pi_1 = [\dots, j, i, \dots], \pi_2 = [\dots, i, j, \dots]$, where only adjacent i and j exchange.
(i) If $c(\{i\}) = 1$, i is inserted at the head of the image orderings when she joins. According to Lemma 3.16, $p(i, \pi_1') \subseteq p(i, \pi_2')$. Hence, $c(p(i, \pi_1')) = c(p(i, \pi_2')) = 1$ and $c(p(i, \pi_1') \setminus \{i\}) \leq c(p(i, \pi_2') \setminus \{i\})$, which derives

$$\begin{aligned}\mathsf{MC}(i, c, p(i, \pi_1')) &= c(p(i, \pi_1')) - c(p(i, \pi_1') \setminus \{i\}) \\ &\geq c(p(i, \pi_2')) - c(p(i, \pi_2') \setminus \{i\}) = \mathsf{MC}(i, c, p(i, \pi_2')),\end{aligned}$$

i.e., if i is the marginal player in π_2', she is also that of the π_1'.
(ii) If $c(\{i\}) = 0$, there are three cases for player i in π_1.

- **Case 1.** i is the marginal player for π_1'. In this case, `sfs-shuf` is flip monotone.
- **Case 2.** i is the marginal player for $\pi'_{1|p(i,\pi_1)}$ but is not for π_1'. Let $S_1 = \{k \mid k \in \pi_1, i \prec_{\pi_1} k\}$. For $k \in S_1$, we have $k \prec_{\pi_1'} i$ (according to Lemma 3.16). We then prove flip monotone in this case by contradiction. Assume that `sfs-shuf` is not flip monotone, i.e., i is the marginal player for π_2' but is not the marginal player for π_1'. Similarly, let $S_2 = \{k \mid k \in \pi_2, i \prec_{\pi_2} k\}$. For $k \in S_2$, we have $k \prec_{\pi_2'} i$ (according to Lemma 3.16). We can observe that $S_1 \cup \{j\} = S_2$. Let r_i^1 and r_i^2 denote the related players of $\pi'_{1|p(i,\pi_1)}$ and $\pi'_{2|p(j,\pi_2)}$, we have $p(r_i^1, \pi'_{1|p(i,\pi_1)}) \setminus \{j\} \subseteq p(r_i^2, \pi'_{2|p(j,\pi_2)})$ (according to Lemma 3.17). Hence,
$$\begin{aligned}p(i, \pi_1') \setminus \{i\} &= S_1 \cup p(r_i^1, \pi'_{1|p(i,\pi_1)}) \\ &\subseteq S_2 \cup p(r_i^2, \pi'_{2|p(j,\pi_2)}) \\ &= p(i, \pi_2') \setminus \{i\}.\end{aligned}$$
Since i is not the marginal player for π_1', $c(p(i, \pi_2') \setminus \{i\}) \geq c(p(i, \pi_1') \setminus \{i\}) = 1$, i.e., i is not the marginal player for π_2', which leads to a contradiction.
- **Case 3.** i is not the marginal player for $\pi'_{1|p(i,\pi_1)}$. There are two cases for player j.
  - **Case 3.1.** j is not the marginal player of $\pi'_{1|p(j,\pi_1)}$. Hence, for π_2, i still cannot be the marginal player of $\pi'_{2|p(i,\pi_2)}$ without j's participation, i.e., `sfs-shuf` is flip monotone.
  - **Case 3.2.** j is the marginal player for $\pi'_{1|p(j,\pi_1)}$. We then prove flip monotone in this case by contradiction. Assume that `sfs-shuf` is not flip monotone, i.e., i is the marginal player for π_2' but is not the marginal player for π_1'. Let r_i^2 denote the corresponding related player. Since i is not the marginal player for $\pi'_{1|p(i,\pi_1)}$, we can get $j \prec_{\pi_1'} r_i^2$; otherwise, i can be inserted after r_i^2 such that i is the marginal player for $\pi'_{1|p(i,\pi_1)}$. Therefore, for π_2, j can be inserted in the same location of π_2' with that of π_1' such that j is the marginal player for $\pi'_{2|p(j,\pi_2)}$, i.e., i is not the marginal player for π_2', which leads to a contradiction.

Taking all above together, we can conclude that SFS-CS is I4EA.

□

4 A CLASS OF SHUFFLE-BASED COST SHARING MECHANISMS

In this section, we propose a class of shuffle-based cost sharing mechanisms based on SFS-CS. Notice that the key point of SF is to ensure that the shuffle rule is a bijection. Recalling the process of `sfs-shuf` in Example 4.1 below, we can see more possibilities.

Example 4.1. Consider the 0-1 valued cost sharing game with $N = \{A, B, C, D, E\}$. For any T, we have $c(T) = 1$ if and only if $\exists S \subseteq T, S \in \{\{A\}, \{B, D\}, \{C, E\}\}$(see Figure 7). The left side is the original order π and for each joining player (colored by blue), the middle and right side show the image orderings given by these methods. The players colored by red are the marginal players, and the players colored by green are the corresponding related players. For the arrival order $\pi = [A, B, C, D, E]$, when player C arrives, `sfs-shuf` inserts her before her predecessor B. Actually, in this step, we can observe that inserting player C between player B and A is also available. Intuitively, it will not affect how we find the late arriving set from the image ordering π' and we can recover the order of the late arriving set by recording the insertion positions; hence, its properties will not be hurt. Interestingly, on the other hand, it may change the player who finally bears the cost; as in this example, player E cannot be the marginal player in the new image orderings when she arrives.

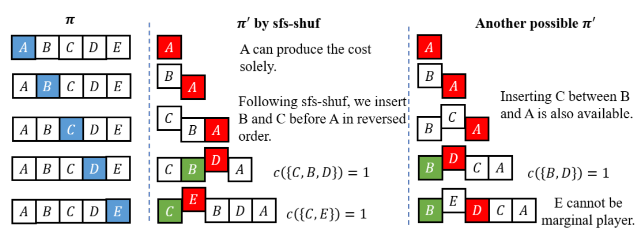

Figure 7: An example of `sfs-shuf` and another possible mapping.

By the given intuitions, the key is how to decide the rearrangements of the following arrived players who cannot be new marginal players. For the case 2 of `sfs-shuf`, it produces a reverse order of their original order, while actually, any bijective rearrangement can yield a valid shuffle rule. Formally, we define the following coordinate functions to decide those rearrangements as follows.

Definition 4.2 (Coordinate function). Given a player set N and a player $i \in N$, a **coordinate function** of i is a bijection function $\mathsf{cd}_i : \bigcup_{S \subseteq N \setminus \{i\}} \Pi(S) \to \bigcup_{S \subseteq N \setminus \{i\}} \Pi(S)$, and for any $T \subseteq N \setminus \{i\}$, $\pi \in \Pi(T)$, and $S \sqsubseteq \pi$, $\mathsf{cd}_i(\pi_{|S}) = (\mathsf{cd}_i(\pi))_{|S}$.

When the marginal player i of π' exists, we use cd_i to decide the rearrangements of those players who cannot be new marginal players after i. When there is no marginal player, we use cd_0 to represent the rearrangements before the first marginal player appears.

We choose different coordinate functions and get a class of shuffle rules described as follows, which extends the case 2 of `sfs-shuf`.

The gsfs-shuf maps any order π to an image ordering π' given by the following iterative procedure:

- The image ordering π' is initialized to be the first player in π. Let i be the next player to arrive in π.
- **Case 1.** If i can be inserted into π' so that i becomes the marginal player in π', she is inserted into the earliest such position.
- **Case 2.** If there is no way to insert i into π' to make her the marginal player, there are two cases for π'.
  - If the marginal player j exists, update π' so that $i \in \mathsf{LA}(\pi')$ and $\pi'_{|\mathsf{LA}(\pi')} = \mathsf{cd}_j(\pi_{|\mathsf{LA}(\pi')})$.
  - Otherwise, update $\pi' = \mathsf{cd}_0(\pi_{|p(i,\pi)})$.

Definition 4.3. The *generalized Shapley-fair shuffle cost sharing mechanism* (GSFS-CS) is the shuffle-based cost sharing mechanism given by the shuffle rule gsfs-shuf.

Now we can get the reconstruction of gsfs-shuf. Compared to the reconstruction of sfs-shuf, the only difference is that we use the inverse of cd_i to reposition the corresponding late arriving set.

Since cd_0 and cd_i are bijective, it can be easily verified that gsfs-shuf is still bijective. Lemma 3.16 and Lemma 3.17 still hold and I4EA can be obtained in the same approach; it can be directly validated by noticing that those proofs only require players who cannot be marginal player are inserted between the marginal player and the related player. Hence, GSFS-CS is SF, OIR, and I4EA.

Theorem 4.4. *For all 0-1 valued monotone cost sharing games, GSFS-CS is OIR, I4EA, and SF.*

5 EXTENSION TO GENERAL COST FUNCTIONS

So far we have proposed a class of mechanisms satisfying all our requirements on 0-1 valued monotone cost sharing games. In this section, we show how GSFS-CS can be applied to general valued monotone cost sharing games. According to [10], an online value sharing mechanism on 0-1 valued monotone game can be extended to general valued setting while maintaining the properties by the following two steps: (1) decompose a general valued monotone function into positive-weighted 0-1 valued monotone components in an online fashion, and (2) run the mechanism simultaneously on each component game and determine each player's share as the weighted sum of her shares from those games. We show that GSFS-CS can be extended to general valued setting through greedy-monotone decomposition (GM), which is an online decomposition algorithm proposed in [10] and meets the requirements. Furthermore, this extended mechanism can satisfy SF, OIR, and I4EA on any monotone cost sharing games.

Lemma 5.1. *Given (N, c, π), the output $D(c) = \{(g_k, \mu_k)\}$ of GM-decomposition is a set of pairs where $c = \sum_k \mu_k g_k$. Note that $\{g_k\}$ are 0-1 valued monotone functions and $\{\mu_k\}$ are non-negative coefficients.*

Now we propose the extended GSFS-CS based on GM formally. The mechanism firstly does GM-decomposition on input cost function c. Then it calculates the cost share in each 0-1 cost monotone games by GSFS-CS and accumulates them with coefficients to be the cost share in c.

Definition 5.2. The **extended generalized Shapley-fair cost sharing mechanisms** (eGSFS-CS) is defined by

$$\bar{\phi}_i(S, c_{|S}, \pi_{|S}) = \sum_{(g_k,\mu_k)\in D(c_{|S})} \mu_k \phi_i^{\mathsf{GSFS-CS}}(S, g_k, \pi_{|S})$$

where $\phi_i^{\mathsf{GSFS-CS}}$ is the cost sharing policy of GSFS-CS.

The properties of eGSFS-CS are maintained and we prove them below.

Theorem 5.3. *eGSFS-CS is SF, OIR, and I4EA.*

Proof. **SF**: Since the Shapley value satisfies *additivity*, we have

$$\bar{\phi}_i(N, c, \pi) = \sum_{(g_k,\mu_k)\in D(c)} \mu_k \phi_i^{\mathsf{GSFS-CS}}(N, g_k, \pi) = \sum_{(g_k,\mu_k)\in D(c)} \mu_k \mathsf{SV}_i(g_k) = \mathsf{SV}_i(c).$$

OIR: Given π, for any $T, S \sqsubseteq \pi$ with $T \subseteq S$, we have

$$\bar{\phi}_i(S, c_{|S}, \pi_{|S}) = \sum_{(g_k,\mu_k)\in D(c_{|S})} \mu_k \phi_i^{\mathsf{GSFS-CS}}(S, g_k, \pi_{|S}) \le \sum_{(g_k,\mu_k)\in D(c_{|T})} \mu_k \phi_i^{\mathsf{GSFS-CS}}(T, g_{k|T}, \pi_{|T}) = \bar{\phi}_i(T, c_{|T}, \pi_{|T}).$$

I4EA: For $\pi_1 = [\dots, i, j, \dots]$ and $\pi_2 = [\dots, j, i, \dots]$, where only adjacent i and j exchange, we have

$$\bar{\phi}_i(N, c, \pi_1) = \sum_{(g_k,\mu_k)\in D(c)} \mu_k \cdot \phi_i^{\mathsf{GSFS-CS}}(N, g_k, \pi_1) \le \sum_{(g_k,\mu_k)\in D(c)} \mu_k \cdot \phi_i^{\mathsf{GSFS-CS}}(N, g_k, \pi_2) = \bar{\phi}_i(N, c, \pi_2).$$

□

6 FUTURE WORK

There are several future directions worth investigation. For 0-1 valued monotone cost sharing games, one may consider characterizing all mechanisms satisfying the desirable properties. Besides, one player may bear the entire cost in a game. It is also important (especially in practice) to look for a more fair cost allocation in each arrival order. For general monotone cost sharing games, since the time complexity of our decomposition is exponential, one may consider designing polynomial time mechanisms.

ACKNOWLEDGMENTS

This work was supported in part by Science and Technology Commission of Shanghai Municipality (No. 22ZR1442200 and No. 23010503000), Shanghai Frontiers Science Center of Human-centered Artificial Intelligence (ShangHAI), Key Laboratory of Intelligent Perception and Human-Machine Collaboration in ShanghaiTech University, and Key Laboratory of Interdisciplinary Research of Computation and Economics in Shanghai University of Finance and Economics. Yao Zhang was also supported by JST ERATO Grant Number JPMJER2301.

REFERENCES

[1] Gustavo Bergantiños and María Gómez-Rúa. 2010. Minimum cost spanning tree problems with groups. *Economic Theory* 43, 2 (2010), 227–262.

[2] Gustavo Bergantiños and Juan J. Vidal-Puga. 2007. A fair rule in minimum cost spanning tree problems. *J. Econ. Theory* 137, 1 (2007), 326–352. https://doi.org/10.1016/J.JET.2006.11.001

[3] Gustavo Bergantiños and Juan J. Vidal-Puga. 2007. The optimistic *TU* game in minimum cost spanning tree problems. *Int. J. Game Theory* 36, 2 (2007), 223–239. https://doi.org/10.1007/S00182-006-0069-7

[4] Dirk Bergemann and Juuso Välimäki. 2019. Dynamic mechanism design: An introduction. *Journal of Economic Literature* 57, 2 (2019), 235–274.

[5] Janina A. Brenner and Guido Schäfer. 2010. Online Cooperative Cost Sharing. In *Algorithms and Complexity, 7th International Conference, CIAC 2010, Rome, Italy, May 26-28, 2010. Proceedings (Lecture Notes in Computer Science, Vol. 6078)*, Tiziana Calamoneri and Josep Díaz (Eds.). Springer, 252–263.

[6] Martin Bullinger and René Romen. 2023. Online Coalition Formation Under Random Arrival or Coalition Dissolution. In *31st Annual European Symposium on Algorithms, ESA 2023, September 4-6, 2023, Amsterdam, The Netherlands (LIPIcs, Vol. 274)*, Inge Li Gørtz, Martin Farach-Colton, Simon J. Puglisi, and Grzegorz Herman (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 27:1–27:18.

[7] Moses Charikar, Howard J. Karloff, Claire Mathieu, Joseph Naor, and Michael E. Saks. 2008. Online multicast with egalitarian cost sharing. In *SPAA 2008: Proceedings of the 20th Annual ACM Symposium on Parallelism in Algorithms and Architectures, Munich, Germany, June 14-16, 2008*, Friedhelm Meyer auf der Heide and Nir Shavit (Eds.). ACM, 70–76.

[8] Michele Flammini, Gianpiero Monaco, Luca Moscardelli, Mordechai Shalom, and Shmuel Zaks. 2021. On the Online Coalition Structure Generation Problem. *J. Artif. Intell. Res.* 72 (2021), 1215–1250. https://doi.org/10.1613/JAIR.1.12989

[9] Masabumi Furuhata, Kenny Daniel, Sven Koenig, Fernando Ordóñez, Maged M. Dessouky, Marc-Etienne Brunet, Liron Cohen, and Xiaoqing Wang. 2015. Online Cost-Sharing Mechanism Design for Demand-Responsive Transport Systems. *IEEE Trans. Intell. Transp. Syst.* 16, 2 (2015), 692–707.

[10] Yaoxin Ge, Yao Zhang, Dengji Zhao, Zhihao Gavin Tang, Hu Fu, and Pinyan Lu. 2024. Incentives for Early Arrival in Cooperative Games. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2024, Auckland, New Zealand, May 6-10, 2024*, Mehdi Dastani, Jaime Simão Sichman, Natasha Alechina, and Virginia Dignum (Eds.). International Foundation for Autonomous Agents and Multiagent Systems / ACM, 651–659. https://doi.org/10.5555/3635637.3662917

[11] María Gómez-Rúa and Juan J. Vidal-Puga. 2011. Merge-proofness in minimum cost spanning tree problems. *Int. J. Game Theory* 40, 2 (2011), 309–329. https://doi.org/10.1007/S00182-010-0243-9

[12] Nicole Immorlica, Mohammad Mahdian, and Vahab S. Mirrokni. 2008. Limitations of cross-monotonic cost-sharing schemes. *ACM Trans. Algorithms* 4, 2 (2008), 24:1–24:25. https://doi.org/10.1145/1361192.1361201

[13] Anirban Kar. 2002. Axiomatization of the Shapley Value on Minimum Cost Spanning Tree Games. *Games Econ. Behav.* 38, 2 (2002), 265–277. https://doi.org/10.1006/GAME.2001.0883

[14] Bin Li, Dong Hao, Hui Gao, and Dengji Zhao. 2022. Diffusion auction design. *Artif. Intell.* 303 (2022), 103631.

[15] Bin Li, Dong Hao, Dengji Zhao, and Tao Zhou. 2017. Mechanism Design in Social Networks. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, Satinder Singh and Shaul Markovitch (Eds.). AAAI Press, 586–592. https://doi.org/10.1609/AAAI.V31I1.10605

[16] David C Parkes. 2007. Online Mechanisms. *Algorithmic Game Theory* (2007), 411–439.

[17] Alessandro Pavan, Ilya Segal, and Juuso Toikka. 2014. Dynamic mechanism design: A myersonian approach. *Econometrica* 82, 2 (2014), 601–653.

[18] Lloyd S Shapley. 1953. A value for n-person games. *Contributions to the Theory of Games* 2, 28 (1953), 307–317.

[19] William Thomson. 1995. Population-monotonic allocation rules. In *Social choice welfare and ethics*, WA Barnett, H Moulin, M Salles, and Schofield NJ (Eds.). Cambridge University Press, 79–124.

[20] Christian Trudeau. 2012. A new stable and more responsive cost sharing solution for minimum cost spanning tree problems. *Games Econ. Behav.* 75, 1 (2012), 402–412. https://doi.org/10.1016/J.GEB.2011.09.002

[21] Christian Trudeau and Juan J. Vidal-Puga. 2017. On the set of extreme core allocations for minimal cost spanning tree problems. *J. Econ. Theory* 169 (2017), 425–452. https://doi.org/10.1016/J.JET.2017.03.001

[22] Christian Trudeau and Juan J. Vidal-Puga. 2020. Clique games: A family of games with coincidence between the nucleolus and the Shapley value. *Math. Soc. Sci.* 103 (2020), 8–14. https://doi.org/10.1016/J.MATHSOCSCI.2019.10.002

[23] Tianyi Zhang, Junyu Zhang, Sizhe Gu, and Dengji Zhao. 2023. Cost Sharing under Private Valuation and Connection Control. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023, London, United Kingdom, 29 May 2023 - 2 June 2023*, Noa Agmon, Bo An, Alessandro Ricci, and William Yeoh (Eds.). ACM, 2161–2169.

[24] Yao Zhang and Dengji Zhao. 2022. Incentives to Invite Others to Form Larger Coalitions. In *21st International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2022, Auckland, New Zealand, May 9-13, 2022*, Piotr Faliszewski, Viviana Mascardi, Catherine Pelachaud, and Matthew E. Taylor (Eds.). International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), 1509–1517. https://doi.org/10.5555/3535850.3536018

[25] Dengji Zhao. 2021. Mechanism Design Powered by Social Interactions. In *AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems, Virtual Event, United Kingdom, May 3-7, 2021*, Frank Dignum, Alessio Lomuscio, Ulle Endriss, and Ann Nowé (Eds.). ACM, 63–67.

[26] Dengji Zhao. 2022. Mechanism Design Powered by Social Interactions: A Call to Arms. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, Luc De Raedt (Ed.). ijcai.org, 5831–5835.

[27] Han Zou, Maged M Dessouky, and Shichun Hu. 2021. An online cost allocation model for horizontal supply chains. *Transportation Research Part C: Emerging Technologies* 122 (2021), 102888.