Best Cost-Sharing Rule Design for Selfish Bin Packing

Changjun Wang∗ Guochuan Zhang†

Abstract

In selfish bin packing, each item is regarded as a selfish player, who aims to minimize the cost-share by choosing a bin it can fit in. To have a least number of bins used, cost-sharing rules play an important role. The currently best known cost sharing rule has a price of anarchy (PoA) larger than 1.45, while a general lower bound 4/3 on PoA applies to any cost-sharing rule under which no items have the incentive to move unilaterally to an empty bin. In this paper, we propose a novel and simple rule with a PoA matching the lower bound of 4/3, thus completely resolving this game. The new rule always admits a Nash equilibrium and its price of stability (PoS) is one. Furthermore, the well-known bin packing algorithm BFD (Best-Fit Decreasing) is shown to achieve a strong equilibrium, implying that a stable packing with an asymptotic approximation ratio of 11/9 can be produced in polynomial time. As an extension of the designing framework, we further study a variant of the selfish scheduling game, and design a best coordination mechanism achieving PoS = 1 and PoA = 4/3 as well.

1 Introduction

Bin packing, one of the fundamental combinatorial optimization models, is a sort of resource-sharing problems. Given a sufficient number of identical resource slots (bins), assign a set of resource requests (items) to a minimum number of slots such that the total amount of requests in each assigned slot is bounded by the given capacity (usually normalized as one). As one of the very first NP-hard problems, bin packing has been extensively studied, while new results are still coming. The most common measure asymptotic approximation ratio evaluates the worst-case ratio between the cost of an approximation algorithm and the optimal cost in the sense that the optimal cost is arbitrarily large. Along this line, the well-known bin packing algorithm BFD (Best Fit Decreasing), which will be formally introduced in Section 4, admits an asymptotic approximation ratio 11/9 [23]. Namely,

\[ BFD(L) \leq \frac{11}{9} OPT(L) + c \]

is valid for any bin packing instance L, where c is a constant, BFD(L) is the number of bins used by BFD, OPT(L) is the optimum number of bins, on L. An APTAS (asymptotic polynomial time approximation scheme) was provided by Fernandez de la Vega and Lueker [17], which admits an asymptotic approximation ratio arbitrarily close to one.

Improving the approximation on bin packing has not yet stopped. There is no evidence for bin packing to rule out a polynomial time algorithm which uses at most one more bin than an optimum packing, assuming P ≠ NP. There have been a lot of efforts in reducing the number of extra bins used, among which the best known algorithm uses O(log OPT(L)) more bins than an optimal packing, owing to the elegant work by Hoberg and Rothvoss [21].

∗ Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China; wcj@amss.ac.cn
† College of Computer Science, Zhejiang University, Hangzhou, China; zgc@zju.edu.cn
In the last two decades, game settings were introduced to classical combinatorial optimization problems. In selfish bin packing, each bin has a unit cost, which is shared by all items using the same bin. Each item is handled by an agent (or simply regards an item as a selfish player), who aims to pay the least cost-share. Instead of packing algorithms, cost-sharing rules play an important role in this game, based on which the agents play selfishly to choose a bin. A packing solution is a Nash equilibrium if no items have an incentive to move unilaterally to a different bin with enough space. We also say such a packing is stable. To measure the efficiency of such a bin packing game, we still consider the number of bins used as the social objective. We are interested in how much loss a stable packing may have in comparison with an optimal packing in the classical setting without game issues (called social optimum). To this end we use PoA (Price of Anarchy) \cite{25, 28} and PoS (Price of Stability) \cite{1} to evaluate the game, which are defined as the worst case asymptotic ratio between the number of bins used in a worst (for PoA) and a best (for PoS) Nash equilibrium and the social optimum, respectively. For a selfish bin packing game $G$, let $OPT(G)$ be the social optimum, and $NE(G)$ be the set of Nash equilibria. For a stable packing $\pi \in NE(G)$, denote by $n(\pi)$ the number of bins used in $\pi$. Then PoA and PoS for the bin packing game are formally defined as \cite{14}

$$PoA = \limsup_{OPT(G) \to \infty} \sup_{G} \max_{\pi \in NE(G)} \frac{n(\pi)}{OPT(G)}$$

and

$$PoS = \limsup_{OPT(G) \to \infty} \sup_{G} \min_{\pi \in NE(G)} \frac{n(\pi)}{OPT(G)}.$$  

### 1.1 Related Work

The selfish bin packing game was initiated by Bíló \cite{3}, who analyzed a natural proportional cost-sharing rule, under which an item pays the cost proportional to its size. It was proved that there exists a Nash equilibrium achieving the social optimum, implying that PoS $= 1$, while PoA falls in the interval $[1.6, 1.67]$. Later two groups were focused on narrowing the interval. Epstein and Kleiman \cite{14}, and Yu and Zhang \cite{30} independently got a lower bound of around 1.6416, while Epstein and Kleiman \cite{14} reached a better upper bound of 1.6428. The gap is very small but still there. Epstein et al. \cite{16} further considered the case that item sizes are restricted and a parametric bound on PoA was derived.

Another natural rule, called the equally sharing rule, simply asks the items in the same bin to pay the same cost regardless of their sizes. Ma et al. \cite{26} showed that under this rule PoA is in $[1.6901, 1.7]$, which is even worse than the proportional rule. Very recently, Dósa and Epstein \cite{12} reduced an upper bound of PoA down to below 1.7.

Both the two natural rules do not seem promising as the inefficiency ($PoA$) is too big. It is thus well motivated to design better cost-sharing rules to improve the efficiency of Nash equilibria. There have appeared a number of interesting results in recent years. Nong et al. \cite{27} proposed a rule, where the payoff of an item is a function of its own size and the largest item size in its bin. They obtained a better $PoA$ of 1.5. Later, Chen et al. \cite{8} designed another cost-sharing mechanism with PoA being between 1.47407 and 1.4748. Zhang and Zhang \cite{31} kept the proportional rule for those bins without large items (whose size is larger than 1/2), while offering a discount function for the items staying with a large item. They derived a better cost-sharing rule with $PoA$ at most $22/15 \approx 1.467$, the same upper bound for the proportional rule applying to bin packing games without large items \cite{16}. Dósa et al. \cite{13} dealt with the problem in a different way. They applied
the proportional rule to the item weights instead of the item sizes. By carefully defining the item weights, they proved the PoA is within 1.4528 and 1.4545 [13].

Recall that the classical bin packing problem admits an APTAS, while the currently best-known upper bound on PoA for selfish bin packing is 1.4545. It leaves a big gap. Does there exist a cost-sharing rule, under which the PoA can be arbitrarily close to one? Unfortunately the answer is negative for a wide class of the bin packing games. Dósa et al. [13] presented an instance showing that the PoA is at least 4/3 if the cost-sharing rule does not encourage a packed item to move to an empty bin. Such an assumption is quite natural, as long as no items will pay a cost more than one. There are also quite a few results in the literature considering different variants of selfish bin packing, such as selfish bin covering [4], selfish vector packing [15].

There is also an extensive study in the literature that shares a similar favor as our work: how to design prior game rules to minimize the inefficiency induced by selfish players. The line of research on coordination mechanisms was introduced by Christodoulou et al. [9], and subsequently studied by Immorlica et al. [22], Caragiannis [5], Kollias [24], Azar et al. [2], and Cole et al. [11]. Most work on coordination mechanisms concerns scheduling games in which \( n \) players assign a respective job to one of \( m \) machines with a goal to minimize its own completion time, and seeks how the price of anarchy varies with the choice of local machine scheduling policies (i.e., the order in which to process jobs assigned to the same machine). Another mainstream of such studies is on the design of cost-sharing protocols for resource selection games [6, 29, 20, 18, 19], where selfish players choose a set of resources to use, and the cost of each resource is a function of the set of players choosing the resource. In these works, the model settings are somewhat different from the bin packing game and their allowable cost-sharing methods are more flexible. Therefore the proved bounds and corresponding approaches therein do not apply to the bin packing game. Moreover, the design of scheduling policy and the design of cost-sharing protocols are believed to be fundamentally different [29, 20]. As far as we know, no work has shown that these two rule-designs are closely related. Interestingly, however, in this paper, we will show that the best cost-sharing rule in some specific resource selection games and the best scheduling policy in some specific scheduling games are essentially equivalent.

1.2 Our Contributions

In this paper, we revisit selfish bin packing by proposing a novel and simple cost-sharing scheme, which is based on the local item sizes in a bin. Basically, we introduce a threshold parameter \( 0 < \Lambda \leq 1 \), and stack up the items in decreasing order of their sizes in a bin. If the total size is at least \( \Lambda \), then the items pay the cost based on a cost density function ranging in \([0, \Lambda]\); otherwise, the bottom item (the largest one) has to pay some extra cost. The cost-sharing rules satisfy many nice and natural properties. By setting \( \Lambda = \frac{3}{4} \), we show that there always exists an optimum packing which is a Nash equilibrium, implying that \( PoS = 1 \). Then we prove that the PoA is at most \( \frac{4}{3} \), matching the lower bound, and thus resolving the wide class of games. Finally, we turn to the classical BFD algorithm which works perfectly under our cost-sharing rule by setting \( \Lambda = \frac{3}{2} \). BFD always produces a strong Nash equilibrium, implying that we can output in \( O(n \log n) \) time a stable packing whose asymptotic approximation ratio is \( \frac{11}{9} (\leq \frac{4}{3}) \). It is also currently the best bound of a Nash equilibrium carried out in polynomial time.

Furthermore, as an extension, we study a variant of the selfish scheduling game, and design a best coordination mechanism with \( PoS = 1 \), \( PoA = \frac{4}{3} \). We show that the best cost-sharing rule in the selfish bin packing is essentially equivalent to the best scheduling policy of this scheduling game.

Organization of the Paper. In the next section, we formally introduce the bin packing game
model and the new cost-sharing rules together with several observations. Section 3 presents the analysis of the rule by showing the main results on PoS and PoA. The analysis of algorithm BFD is provided in Section 4. In Section 5 we extend the results to a variant of the selfish scheduling game. Finally, we conclude the paper in Section 6.

2 Game Model and A New Cost-Sharing Scheme

In this section, we formally introduce the game model of selfish bin packing and some definitions. Then we propose our new cost-sharing rules that satisfies many nice properties.

2.1 Definitions

Let $N = \{1, 2, \ldots, n\}$ be a set of items, where each item $i \in N$ has a size $s_i \in (0, 1]$ and is controlled by a selfish agent. For simplicity, in the following we will directly view the items as game players. There are plenty of unit-capacity bins being available to pack the items, but each with an unit opening cost. For a set of items $I \subseteq N$ packed into a bin $B$, we say that $B$ is valid or a valid bin, if $s(I) := \sum_{i \in I} s_i \leq 1$. A packing $\pi$ is defined to be an allocation of the items into some used bins $B_1, \ldots, B_m$ such that $N = B_1 \cup B_2 \cdots \cup B_m$ and each bin $B_k$ is valid, where $B_k (1 \leq k \leq m)$ denotes a bin as well as the set of items allocated in that bin (slightly abusing notation). For the items packed into one bin, they need to share the bin’s opening cost according to a pre-given cost-sharing rule $\Xi$. We use $c_i(\pi) \geq 0$ to represent the cost-share of item $i$ under packing $\pi$. Each item aims to minimize its own cost-share. The strategy $b_i$ of each item $i \in N$ is to select a valid bin to stay in. Then a feasible strategy profile $b$ is equivalent a unique packing $\pi = (B_1, \ldots, B_m)$ with nonempty bins $B_k$, i.e., $b$ and $\pi$ can be transformed by $B_k = \{i \in N : b_i = B_k\}$ for $k = 1, \ldots, m$, or by $b_i = B_k$ for every $i \in B_k$. The social objective is to minimize the number of used bins to pack all the items.

**Definition 1.** A strategy profile $b$ (or a packing $\pi$) is called a Nash equilibrium (NE, for short) if no item can reduce its cost by unilaterally changing its strategy, i.e., $c_i(b) \leq c_i(b_i', b_{-i})$, $\forall i \in N$.

Designing cost sharing rules to improve the efficiency of equilibrium (i.e., minimize PoA or PoS) for different resource selection games has been widely studied in literature [6, 29, 20, 18, 31, 19]. In these studies, many desirable properties have been proposed to restrict the design space of feasible cost sharing rules. Below we define them and some new natural ones in this context.

**Definition 2 (Properties of Cost Sharing Rules).** A cost sharing rule $\Xi$ is:

1. **stable,** if it induces only games that admit at least one NE.
2. **budget-balanced,** if for every bin $B$ of any packing $\pi$, $\sum_{i \in B} c_i(\pi) = 1$.
3. **local,** if for any two different packings $\pi \neq \pi'$ that both contain a bin $B$ with the same set of packed items, and every item $i \in B$, it holds $c_i(\pi) = c_i(\pi')$.
4. **monotone,** if for any two bins $B, B'$ satisfying $B \subseteq B'$ (suppose w.l.o.g. $B$ is a bin of packing $\pi$ and $B'$ is with $\pi'$) and every item $i \in B$, $c_i(\pi) \geq c_i(\pi')$.
5. **fair,** if for any two items $i, j$ that are packed in the same bin of any packing $\pi$, it holds $(s_i - s_j) \cdot (c_i(\pi) - c_j(\pi)) \geq 0$.
6. **polynomial time computable,** if for every packing $\pi$, the cost-shares $c_i(\pi)$ of all $i \in N$ can be computed in polynomial time of the input size.
Now we discuss the above properties in more detail. Note that the NEs in Definition 1 are in fact pure strategy Nash equilibria. Thus a stable cost sharing rule ensures the existence of a pure strategy Nash equilibrium in the induced game of selfish bin packing. Budget-balance is a natural, straightforward requirement for a cost sharing rule in the economics literature. Locality means that for any bin, the cost-shares of the items in this bin depend only on the bin itself (i.e., the set of items in this bin), and disregard the information of other bins. A local cost sharing rule ensures that the opening costs can be distributed in a decentralized and local manner, that is, the bin does not have to know any cost-shares of other items or the allocation for other bins. Monotonicity stating that moving new items into a bin should not increase the cost-shares of previous items of this bin is a natural property in selfish bin packing as the cost of a bin is always a constant. Since the capacity of each bin is limited, in a fair cost-sharing rule, for the items packed in the same bin, an item with a larger size should not pay less than a smaller item. Polynomial-time computability of the rules is crucial to make the games run efficiently.

2.2 A new cost-sharing scheme

In this subsection, we will propose our new cost-sharing scheme. First recall that under the classical propositional cost-sharing rule, the items are incentivized to move into feasible bins with more total loads. Differently from that, the rough idea of our new rules is to motivate each item $i \in N$ to choose a feasible bin with more total loads of larger items (than $i$’s size). Even though this motivation seems no more effective (at getting the bins more loaded) than the propositional rule, we will finally show that this kind of strategic behaviors will result in the best possible NE efficiency. However, to realize this kind of incentive and make the nice properties in Definition 2 satisfied, the new cost-sharing scheme needs to be well designed.

Local-Size-Based Cost-sharing Scheme. The new framework of cost-sharing rules is parameterized with a constant $\Lambda \in (0, 1]$ and a “cost density” function $f(x)$. Namely, a constant $\Lambda$ and a function $f(x)$ will uniquely determine one cost-sharing rule. Given a constant $\Lambda$, the function $f(x)$ with $x \in [0, 1]$ is a non-negative function satisfying: i). $f(x)$ is decreasing in $[0, \Lambda)$ with $\int_0^\Lambda f(x)dx = 1$; and ii). $f(x) = 0$ for $x \in [\Lambda, 1]$. A simple selection of $f(x)$ is letting $f_o(x)$, where $f_o(x)$ is a linear function,

$$f_o(x) := \begin{cases} \frac{2}{\Lambda} - \frac{2x}{\Lambda^2}, & \text{for } 0 \leq x < \Lambda; \\ 0, & \text{for } \Lambda \leq x \leq 1. \end{cases}$$

The local-size-based (LSB) cost-sharing rules work as follows:

1). Re-index the items in $N$ such that $s_1 \geq s_2 \geq \ldots \geq s_n$ and fix this index (order). If index $i < j$, we call item $i$ “bigger” than item $j$ or $j$ is “smaller” than $i$, denoted by item $i \succ j$ item $j$. Note that a bigger item may have the same size as the smaller item.

2). Given a packing $\pi$, for each bin $B$ of $\pi$, suppose w.l.o.g. that $B = \{i_1, i_2, \ldots, i_l\}$ and $i_1 \succ i_2 \succ \ldots \succ i_l$. Denote by $S^h_B := \sum_{j=1}^h s_{i_j}$ for $h \leq l$ the total size of the biggest $h$ items in bin $B$. Note that $S^l_B = s(B)$. Then the cost share of item $i_h$ for $h = 1, 2, \ldots, l$ is

$$c_{i_h}(\pi) = \begin{cases} \int_0^{S^1_B} f(x)dx + (1 - \int_0^{s(B)} f(x)dx), & \text{for } h = 1; \\ \int_{S^h_B}^{S^{h-1}_B} f(x)dx, & \text{for } h > 1. \end{cases}$$
In a bin, we call the biggest item (the item with smallest index) the bottom item of the bin. In the above rules, except for the bottom item, an item’s cost depends on the total size of bigger items and its own size, disregarding the sizes of smaller items. For the bottom item, if the total load of the bin \( s(B) \) is less than \( \Lambda \), then it should also pay for the extra remaining cost \( \int_{s(B)}^{\Lambda} f(x) dx \). Apparently, every LSB cost-sharing rule is budget-balanced and local.

**Observation 2.1.** The LSB cost-sharing rules are budget-balanced and local.

In addition, we have the following obvious observation about the LSB cost sharing scheme.

**Observation 2.2.** The LSB cost-sharing rules are fair and monotone.

**Proof.** By noting that the density function \( f(x) \) is monotone non-increasing, it is easy to see that the LSB cost-sharing rules are fair. To see the monotonicity, pick any two bins \( B, B' \) satisfying \( B \subseteq B' \). Suppose w.l.o.g. that \( B \) (resp. \( B' \)) is a bin of packing \( \pi \) (resp. \( \pi' \)), and \( j \in B' \setminus B \). Then for any item \( i \in B \), if \( i \) is bigger than \( j \) but not the bottom item of \( B \), then the LSB cost-sharing scheme guarantees that \( i \)'s cost share remains the same, \( c_i(\pi) = c_i(\pi') \); otherwise, \( i \) is either a bottom item of \( B \) or smaller than \( j \), then we have \( c_i(\pi) \geq c_i(\pi') \).

Generally, the LSB rules may not be stable or polynomial time computable. But through selecting some proper \( \Lambda \) and \( f(x) \), we can also guarantee that the corresponding LSB cost-sharing rule is stable and polynomial time computable.

**Theorem 1.** Let constant \( \Lambda \leq 3/4 \) and \( f(x) = f_o(x) \). Then this LSB cost-sharing rule satisfies all the proposed properties in Definition 2.

**Proof.** Having observations 2.1 and 2.2 we only need to show that this cost-sharing rule is stable and polynomial time computable. The stability of the cost-sharing rule will be proved in Subsection 3.1. Since \( f_o(x) \) is a linear function and its integral can be given explicitly, then for any packing \( \pi \), the time of computing all the cost shares is apparently polynomial in the instance’s input.

In the remaining part of the paper, when we mention selfish bin packing, if not otherwise specified, it is always assumed that the games are induced by the LSB cost-sharing scheme. As we mainly care about the NE packings and their performances in selfish bin packing, below we will see that the density function \( f(x) \) is not critical in the design of our LSB cost-sharing rule.

**Proposition 1.** Bin packing games with the same \( \Lambda \) but different cost density functions have exactly the same set of NE packings.
Proof. Note that in these games, a packing is an NE if and only if no item of this packing can reduce its cost by unilaterally moving into another feasible bin. Pick any item $i \in N$ in any packing $\pi$. With the LSB cost-sharing rule, if $i$ has incentive to move, then it must be the case that, either in the bin $B$ where $i$ stays, the total size of items that are bigger than $i$ is less than $\Lambda$ (otherwise $i$’s cost would be 0) and there exists another feasible bin (for $i$ to move in) with larger total size of bigger (than $i$) items, or $i$ is a bottom item of its bin $B$ with $s(B) < \Lambda$ and there exists another feasible bin (for $i$) with total load larger than $s(B) - s_i$. Thus we can see, once the constant $\Lambda$ is given, no matter what the exact density function $f(x)$ is, the NE packings are already uniquely determined.

With Proposition [1] at hand, if not otherwise specified in the following, we implicitly have $f(x) = f_o(x)$ and will only refer to bin packing games with certain constant $\Lambda$. As for games with different $\Lambda$s, we show that their NE sets are also highly correlated.

**Proposition 2.** For two different constants $\Lambda_1 > \Lambda_2$, the NE set of bin packing game with $\Lambda_1$ is a subset of the NE set of game with $\Lambda_2$.

**Proof.** Suppose packing $\pi$ is an NE of the bin packing game with $\Lambda_1$. In the game with $\Lambda_2$, if $\pi$ is not an NE, then there must exist an item $i$ in some bin $B$ such that it can reduce its cost by unilaterally moving into another feasible bin $B'$. This means that either the total size of items bigger than $i$ in $B$ is less than $\Lambda_2$ and $B'$ has a larger total size of bigger items, or $i$ is the bottom item of $B$ with $s(B) < \Lambda_2$ and $s(B') > s(B) - s_i$. Recall that $\Lambda_1 > \Lambda_2$. We can deduce that in the bin packing game with $\Lambda_1$, under $\pi$, item $i$ can also reduce its cost by unilaterally moving to $B'$, contradicting the definition of $\pi$. □

3 A Best LSB Cost-Sharing Rule

In this section, we will focus on the bin packing games with constant $\Lambda = \frac{3}{4}$ in our LSB cost-sharing rules, and show the LSB rule with $\Lambda = \frac{3}{4}$ is the best possible in reducing equilibria’s inefficiency.

3.1 Existence of Equilibria and $\text{PoS} = 1$

In the classical bin packing problems, an optimal solution is a packing with the minimum number of bins used. In the following, we will show that there always exists one optimal packing that being the NE of the game, which also implies its $\text{PoS} = 1$.

**Theorem 2.** The bin packing game always has an NE that uses the minimum number of bins among all packings.

To prove Theorem [2] we will seek optimal packings with some nice properties, among which we could figure out an NE. Given a packing $\pi$, for every item $i$ and the bin $B$ containing $i$, we refer to the total size of bigger items (than $i$) of $B$ as item $i$’s height level, denoted by $l_i(\pi)$. Apparently in this game, every item would like to stay in a bin with its own height level as high as possible. It would be relatively easy to ensure the existence of an optimal packing under which no item can unilaterally improve its height level. However, such packing is not necessarily an NE as some bottom item might still want to deviate when it stays in a bin with a total size less than $\Lambda$. This is because the bottom items do not only care about their height levels but also the total size of their bins. Thus the key challenge of proving Theorem [2] is how to guarantee the stability of bottom items in the corresponding optimal packings.
Now let $m^*$ be the minimum number of used bins that can pack all the items of $N$, and $\mathcal{P}^*$ be the set of the optimal packings, each of which uses $m^*$ bins. For each packing $\pi \in \mathcal{P}^*$, we use $\{b_1^\pi, b_2^\pi, \ldots, b_m^\pi\}$ with $b_1^\pi > \cdots > b_m^\pi$ to denote the set of the bottom items in the $m^*$ bins of $\pi$. For two packings $\pi, \pi' \in \mathcal{P}^*$, we say that $\pi$ lexicographically dominates $\pi'$ with respect to the bottom items if either $b_i^\pi > b_i^{\pi'}$, or for some $k > 1$ such that $b_j^\pi = b_j^{\pi'}$ for every $j < k$, and $b_k^\pi > b_k^{\pi'}$. Broadly speaking, packing $\pi$ dominates $\pi'$ if its bottom items are lexicographically “bigger” than $\pi'$s bottom items. Let $\mathcal{P}^*_b \subseteq \mathcal{P}^*$ be the set of packings that can not be dominated by other packings in $\mathcal{P}^*$. Then apparently, $\mathcal{P}^*_b \neq \emptyset$ and all packings in $\mathcal{P}^*_b$ have the same set of bottom items, which is denoted by $\{b_1^\pi, b_2^\pi, \ldots, b_m^\pi\}$.

Then assume that the items in $N \setminus \{b_1^\pi, b_2^\pi, \ldots, b_m^\pi\}$ are $a_1 > a_2 > \cdots > a_{n-m^*}$. For two packings $\pi, \pi' \in \mathcal{P}^*_b$, we say that $\pi$ is lexicographically higher than $\pi'$ with respect to $\{a_1, \ldots, a_{n-m^*}\}$ if for some $t \leq n-m^*$ the height levels satisfy: $l_{ah}(\pi) = l_{ah}(\pi')$ for every $h < t$ and $l_{at}(\pi) > l_{at}(\pi')$. In other words, for two packings $\pi, \pi' \in \mathcal{P}^*_b$, since they have the same set of bottom items, then $\pi$ is higher than $\pi'$ if the height levels of $\{a_1, a_2, \ldots, a_{n-m^*}\}$ under $\pi$ are lexicographically higher than their height levels under $\pi'$. Since $\mathcal{P}^*_b \neq \emptyset$, then there must exist a packing $\pi^\ast \in \mathcal{P}^*_b$ such that no other packing in $\mathcal{P}^*_b$ is lexicographically higher than it. Let $B_1, \ldots, B_{m^*}$ be the bins that contain the bottom items $b_1^\ast, \ldots, b_{m^*}^\ast$, respectively. To ease the following discussions, we additionally require $\pi^\ast$ to satisfy the following condition: if there is $h < m^*$ such that $s_{b_h^\ast} = s_{b_{m^*}^\ast}$, then the second biggest item of $B_{m^*}$ is not “bigger” than the second biggest item of bin $B_h$ (including the cases of the second item being empty). This additional requirement can be easily realized by directly swapping all the bottom items of the corresponding two bins from a given $\pi^\ast$.

Note that under any packing $\pi \in \mathcal{P}^*_b$, for every non-bottom item $i \in \{a_1, \ldots, a_{n-m^*}\}$, its cost share is $\int_{l_i(\pi)}^{l_i(\pi)+s_i} f(x) dx$. Recall that $f(x)$ is a non-increasing function. Thus under the selected packing $\pi^\ast$, by its definition, no item of $\{a_1, a_2, \ldots, a_{n-m^*}\}$ has the incentive to unilaterally deviate from its current bin since the single deviation cannot increase this item’s height level (thus decrease its cost). If $\pi^\ast$ is an NE packing, then we are done. Otherwise, there must exist one unsatisfied bottom item that wants to move into another feasible bin to decrease its cost. We first have the following claims.

Claim 3.1. If $\pi^\ast$ is not an NE packing, then the smallest bottom item $s_{b_{m^*}^\ast} \leq 1/2$.

Proof. Since $\pi^\ast$ is not an NE, then at least one bottom item of some bin can unilaterally move into another bin. This means that a bin can accommodate some two bottom-items of $\{b_1^\ast, b_2^\ast, \ldots, b_{m^*}^\ast\}$, which implies $s_{b_{m^*}^\ast} \leq 1/2$. \hfill \Box

Claim 3.2. If $\pi^\ast$ is not an NE packing, then the last bin $B_{m^*}$ contains only one item (i.e., $b_{m^*}^\ast$).

Proof. Now suppose on the contrary that there are at least two items in $B_{m^*}$. Let $x$ be the second biggest item in $B_{m^*}$. We also use $x$ to represent its size if there is no confusion. It is easy to check that item $x$ can not fit into any other bin. Because if $x$ can fit into another bin (say $B_j$), recalling $b_j^\ast > b_{m^*}^\ast$, and the additional requirement of $\pi^\ast$, it would imply that item $x$ would achieve a higher height level (than $l_x(\pi^\ast) = s_{b_{m^*}^\ast}$) after moving into $B_j$ and other bigger items’ height levels could not be affected, contradicting the selection of $\pi^\ast$.

Since $\pi^\ast$ is not an NE, let $b_k^\ast$ be the bottom item that is willing to move from its current bin $B_k$ to another bin $B_h$, where $s_{b_k^\ast} + s(B_h) \leq 1$. Now we claim it must be the case that bin $B_k$ is exactly $B_{m^*}$. Because otherwise from the definition of item $x$ and the fact that $b_{m^*}^\ast$ is the smallest bottom item, we know that item $x$ can also fit into $B_h$, thereby leading to a contradiction to the above $x$’s fitness. Therefore, the unsatisfied bottom item $b_k^\ast$ can only move from $B_k$ to $B_{m^*}$.
By the above definition of $b_k^*$ and the given LSB rule, we can deduce that $s(B_k) < 3/4$, and $s(B_k) - s_{b_k^*} < s(B_{m^*}) \leq 1 - s_{b_k^*}$. Since $x$ cannot fit into $B_k$, while $b_k^*$ fits into $B_{m^*}$, we have $x > 1 - s(B_k) > 1/4$ and $s(B_{m^*}) < s(B_k) < 3/4$. Therefore, we can deduce that $B_k$ contains at most two items that are bigger than item $x$.

Let $y$ be the second biggest item of $B_k$ (by slightly abusing notation, its size is also denoted by $y$). Then we have $b_{m^*} > y$, because otherwise, by moving $b_k^*$ from $B_k$ to $B_{m^*}$, we would obtain another optimal packing that dominates all the packings in $P_b^*$ w.r.t. the bottom items, contradicting the definition of $P_b^*$. Define $I'_k := B_k \setminus \{b_k^*, y\}$, $I'_{m^*} := B_{m^*} \setminus \{b_{m^*}\}$. Since $b_k^*$ can unilaterally move into $B_{m^*}$, thus

$$s_{b_k^*} + y + s(I'_{m^*}) \leq s_{b_k^*} + s_{b_{m^*}} + s(I'_{m^*}) = s_{b_k^*} + s(B_{m^*}) \leq 1.$$ 

Meanwhile,

$$s_{b_{m^*}} + s(I'_k) < s(B_k) < 3/4.$$ 

Therefore, by swapping $I'_k$ and $I'_{m^*}$, we will get a new optimal packing $\pi^\circ \in P_b^*$ from $\pi^*$, with the two bins $B_k, B_{m^*}$ of $\pi^*$ replaced by another two feasible bins $B'_k = \{b_k^*, y\} \cup I'_{m^*}, B'_{m^*} = \{b_{m^*}\} \cup I'_k$.

Recall that item $x$ is bigger than any item in $I'_k$. By noting $x \in B'_k$ in the new packing $\pi^\circ$, and combining with $b_k^* > b_{m^*}$ and the additional requirement of $\pi^*$, it is easy to see that packing $\pi^\circ$ is lexicographically higher than $\pi^*$, contradicting the definition of $\pi^*$. Now we can conclude that if $\pi^*$ is not an NE, then $b_{m^*}$ is the unique item in bin $B_{m^*}$.

It’s possible that $\pi^*$ is not an NE. So in the set $P_b^*$, we cannot guarantee the existence of an NE packing. Fortunately, in that case, we would have a nice estimation of the last bin’s size. Now we expand the set $P_b^*$ to the set $P_{b^*} \subseteq P^*$, which consist of all the optimal packings that only have $b_1^*, b_2^*, \ldots, b_{m^*-1}^*$ as the fixed bottom items (i.e., relax the possibility of the last bin’s bottom item). Apparently, $P_{b^*} \subseteq P_{b^*}^*$. We will show that $P_{b^*}^*$ must contain an NE packing. Let $\hat{\pi}^* := \hat{B}_1 \cup \hat{B}_2 \cup \cdots \cup \hat{B}_{m^*} \in P_{b^*}$ be the packing that has the minimum total size of the last bin $\hat{B}_{m^*}$, where $\hat{B}_k$ has the item $b_k^*$ as the bottom item for $k = 1, 2, \ldots, m^*-1$. Besides, we refine $\hat{\pi}^*$ such that all the non-bottom items of the first $m^*-1$ bins (i.e., the items in $\hat{B}_1 \cup \hat{B}_2 \cup \cdots \cup \hat{B}_{m^*-1} \setminus \{b_1^*, b_2^*, \ldots, b_{m^*-1}^*\}$) are in the lexicographically highest levels. That is, under $\hat{\pi}^*$, these non-bottom items have no incentive to unilaterally migrate to another bin of these $m^*-1$ bins. Now we are ready to complete the proof of Theorem 2.

Proof of Theorem 2

Proof. We claim that either $\pi^*$ or $\hat{\pi}^*$ is an NE packing. Now suppose $\pi^*$ is not an NE. Then from the definition of $\hat{\pi}^*$ and Claim 3.2, we know, $s(\hat{B}_{m^*}) < s_{b_{m^*}} \leq s_{b_{m^*-1}} \leq \cdots \leq s_{b_1^*}$. In this case, for all the non-bottom items in $\hat{B}_1 \cup \hat{B}_2 \cup \cdots \cup \hat{B}_{m^*-1}$, no one would like to unilaterally move into the last bin $\hat{B}_{m^*}$, i.e., such migration would not lower the item’s height level from at least $s_{b_{m^*-1}}$ to $s(\hat{B}_{m^*})$, thus increasing its cost share. Combining with the refinement of $\hat{\pi}^*$, we know that $\hat{\pi}^*$ is a stable packing for all the non-bottom items in $\hat{B}_1 \cup \hat{B}_2 \cup \cdots \cup \hat{B}_{m^*-1}$. For any item in the last bin $\hat{B}_{m^*}$, no one can unilaterally move into the first $m^*-1$ bins since otherwise, the resulting new packing will have a smaller total size of the last bin than $\hat{\pi}^*$. This also implies that among the first $m^* - 1$ bins, no bottom item of one bin can unilaterally fit into another bin. Besides, for any bottom item $b_k^*$ with $k \in \{1, 2, \ldots, m^*-1\}$, if it can fit into the last bin $\hat{B}_{m^*}$, then we must have $s(\hat{B}_k) - s_{b_k^*} \geq s(\hat{B}_{m^*})$ because otherwise, moving $b_k^*$ into $\hat{B}_{m^*}$ would result in a new packing with an even smaller size of the “last bin”. So under $\hat{\pi}^*$, every bottom item of $\{b_1^*, b_2^*, \ldots, b_{m^*-1}^*\}$ is stable. Therefore, from the above analysis we can conclude that if $\pi^*$ is not an NE, then $\hat{\pi}^*$ must be an NE packing.
Combining Proposition 3.2 and Theorem 2, we can also reach the following corollary for games with $\Lambda \leq \frac{3}{4}$.

**Corollary 1.** For every bin packing game with $\Lambda \leq \frac{3}{4}$, its cost-sharing rule is stable and $PoS = 1$.

**Remark 1.** From Proposition 2 we know, increasing the constant $\Lambda$ in the LSB cost-sharing rule will make the corresponding NE set shrink, thus is helpful to reduce the game’s $PoA$. It seems that we should set $\Lambda = 1$ instead of $\Lambda = \frac{3}{4}$. However, for the bin packing games with $\Lambda = 1$, we do not know whether an NE exists. In fact, for the bin packing games with $\Lambda > \frac{3}{4}$, the existence of NEs and their exact $PoS$ are still unknown and open. Fortunately, in the next subsection, we will show that the LSB cost-sharing rule with $\Lambda = \frac{3}{4}$ is already the optimal rule to attain the best possible $PoA = 4/3$.

### 3.2 Best possible $PoA$

In this section, we prove that the $PoA$ of the bin packing game with $\Lambda = \frac{3}{4}$ is at most $4/3$. From [13] we know, this bound is the best possible.

Given an NE packing $\pi$ of items in $N$. Suppose without loss of generality that $\pi$ is composed of $m$ nonempty bins $B_1, B_2, \ldots, B_m$ with $s(B_1) \geq s(B_2) \geq \cdots \geq s(B_m)$. For each bin $B_k$, we still use $b_k$ to denote its bottom item, meanwhile, let $t_k$ be the smallest item in it.

**Lemma 3.1.** For each bin $B_k$, if its smallest item pays (positive) cost, then the front bins are not feasible for any item of $B_k$ to move in, i.e., if $s(B_k) - s_{tk} < \frac{3}{4}$, then $\forall j < k$, we have $s(B_j) + s_{tk} > 1$.

**Proof.** Suppose on the contrary that there exists $j < k$ such that $s(B_j) + s_{tk} \leq 1$, which means that $t_k$ can fit into $B_j$. Let $B_j' \subseteq B_j$ be the subset of items of $B_j$ that are bigger than $t_k$, i.e., $B_j' := \{ i \in B_j : i > t_k \}$. Since $s(B_k) - s_{tk} < \frac{3}{4}$, by recalling the LSB cost-sharing rule and the fact that $\pi$ is an NE, then it must hold that $s(B_j') \leq s(B_k) - s_{tk} < \frac{3}{4}$, because otherwise $t_k$ would be able to reduce its cost by unilaterally moving into $B_j$. At this case, we have $B_j \setminus B_j' \neq \emptyset$ since $s(B_j') \leq s(B_k) - s_{tk} < s(B_k) \leq s(B_j)$. Let $i_j \in B_j \setminus B_j'$ be the biggest item of $B_j \setminus B_j'$. Then $i_j$’s cost share under $\pi$ is $\int_{s(B_j')}^{s(B_j')} f(x)dx$. Note that all items in $B_k$ are bigger than item $i_j$ since $t_k > i_j$ and $s(B_k) + s_{ij} \leq s(B_j) + s_{tk} \leq 1$. This means that under packing $\pi$, item $i_j$ can also unilaterally move into bin $B_k$ with a new cost share $\int_{s(B_k)}^{s(B_j)} f(x)dx$, which is smaller than $\int_{s(B_j')}^{s(B_j)} f(x)dx$ since $s(B_j') \leq s(B_k) - s_{tk} < \frac{3}{4}$. This contradicts the fact that $\pi$ is an NE packing.

It is clear that $s(B_{m-1}) + s(B_m) > 1$ either by $s(B_m) \geq 3/4$ or by Lemma 3.1 with $s(B_m) - s_{tm} < 3/4$. In the following, we will prove that the optimal packing uses at least $\lceil \frac{3m-3}{4} \rceil$ bins, which then implies $PoA \leq 4/3$.

**Theorem 3.** An optimal packing uses at least $\lceil \frac{3m-3}{4} \rceil$ bins, i.e., $PoA \leq \frac{4}{3}$.

To prove the theorem, first note that if $s(B_{m-2}) \geq \frac{3}{4}$, then the total size of the $n$ items is at least $\frac{3}{4} \cdot (m - 2) + s(B_{m-1}) + s(B_m) > \frac{3}{4}(m - 1)$. Since each bin can accommodate items with a total size at most 1, the conclusion of the theorem follows. Hence, in the discussion below, we will always assume that $s(B_{m-2}) < \frac{3}{4}$.

Let $L^* := s(B_{m-1})$. Obviously, $L^* \leq s(B_{m-2}) < \frac{3}{4}$. Then we use $L^*$ to divide all items into three types: for any item $i \in N$, if its size $s_i \geq L^*$, we call it a super item; if $L^* > s_i > 1 - L^*$, then we call it a regular item; otherwise $s_i \leq 1 - L^*$, we call it a tiny item.

**Lemma 3.2.** There are no tiny items in the last two bins $B_{m-1}$ and $B_m$. 

Proof. First we can see that the statement holds for $B_m$, because otherwise a tiny item can move into bin $B_{m-1}$, contradicting Lemma 3.1. Next, we show that $B_{m-1}$ does not contain any tiny item either. Note that if there is only one item in $B_{m-1}$, then this item must be a super item because of $s(B_{m-1}) = L^*$. If there are more than one item in $B_{m-1}$, then $s(B_{m-1}) < s(B_{m-2}) < \frac{3}{4}$ implies that $s_{t_{m-1}} > \frac{1}{4}$ (by Lemma 3.1). $B_{m-1}$ consists of only two items, and $\frac{1}{4} < s_{t_{m-1}} < \frac{1}{2}$. Suppose by way of contrary that there exists a tiny item in $B_{m-1}$. Then $t_{m-1}$ must be one such tiny item. By the definition of tiny item, we have $s_{t_{m-1}} \leq 1 - L^*$ and $s_{t_{m-1}} + s(B_m) \leq s_{t_{m-1}} + L^* \leq 1$. Recall that $\pi$ is an NE packing, which means that $t_{m-1}$ has no incentive to move into $B_m$. This, together with the fact that $B_m$ contains no tiny item, implies that $s_{b_{m-1}} \geq s(B_m)$. Recalling $s_{b_{m-1}} < \frac{1}{2}$, thus $s_{b_{m-1}} + s(B_m) < 1$, which means that the bottom item $b_{m-1}$ can fit into the last bin $B_m$. Since $\pi$ is an NE under the LSB cost-sharing rule with $\lambda = \frac{3}{4}$, it must be the case that $s_{t_{m-1}} \geq s(B_m) > 1 - L^*$, thereby leading to a contradiction. Now we can conclude that neither $B_{m-1}$ nor $B_m$ contains any tiny item.

Recall the classification of items. For any feasible packing, a bin can not accommodate one super item plus with a regular item. In addition, since a regular item is larger than $\frac{1}{4}$, a bin can accommodate at most 3 of them. Now we claim that under $\pi$, except for the last bin $B_m$, each bin either contains a super item or at least two regular items. This is because, otherwise the tiny items in a violating bin would unilaterally like to move into the bin $B_{m-1}$ which contains no tiny item, contradicting the NE assumption. By a similar argument, we arrive at the following lemma.

Lemma 3.3. Under the NE packing $\pi$, except the last bin $B_m$, the total size of all non-tiny items (i.e., the super items and regular items) in each bin is at least $L^*$.

Assume that among the first $(m-1)$ bins of $\pi$, there are $k_1$ bins with only one super item, $k_2$ bins with exactly two regular items, and $k_3$ bins with exactly three regular items, where $k_1 + k_2 + k_3 = m - 1$. So in the first $(m-1)$ bins, there are $k_1$ super items, and $2k_2 + 3k_3$ regular items. Recall that there is at least one regular or super item in $B_m$. In the following, we will finally show that even to pack all the non-tiny items, every valid packing (including the optimal one) uses at least $k_1 + \lceil\frac{3}{2}(k_2 + k_3)\rceil$ bins. To simplify the analysis, we will just assume that $B_m$ contains only one regular item. Thus we need to pack $k_1$ super items and $2k_2 + 3k_3 + 1$ regular items. Observe that under any valid packing, a super item can not be packed with any regular item into one bin. So each super item alone will occupy one bin. So we only need to show that the $2k_2 + 3k_3 + 1$ regular items will take at least $\lceil\frac{3}{2}(k_2 + k_3)\rceil$ bins to pack them. We rename all the regular items according to their size ranks such that $i_1 > i_2 > \cdots > i_p$, where $p = 2k_2 + 3k_3 + 1$. Let $N_r := \{i_1, i_2, \ldots, i_{k_2+1}\}$ be the biggest $k_2 + 1$ regular items.

Claim 3.3. No bin can accommodate two items in $N_r$ with another regular item.

Proof. Recall that under $\pi$, we have $k_2$ bins containing exactly two regular items (and perhaps some tiny items). Let $b_{j'}$ be the minimum bottom item of these $k_2$ bins, where its bin is denoted as $B_{j'}$. Then apparently, the size of $b_{j'}$ is no more than the size of regular item $i_{k_2}$. Therefore the total size of the two regular items in $B_{j'}$ is no more than the total size of $s(i_{k_2}) + s(i_{k_2+1})$. Then by Lemma 3.3, we know that $s(i_{k_2}) + s(i_{k_2+1}) \geq L^*$. It implies that any two regular items from $N_r$ sum up to at least $L^*$. Combining with the definition of regular items, we can conclude that the total size of two regular items of $N_r$ and any one more regular item will exceed the bin size 1.

Finally we can prove Theorem 3 by showing that every valid packing (including optimal ones) uses at least $\lceil\frac{3m-3}{4}\rceil$ bins, even for packing all the non-tiny items.

Proof of Theorem 3.8
Proof. First we show that the \( p = 2k_2 + 3k_3 + 1 \) regular items will take at least \( \lceil \frac{3}{4}(k_2 + k_3) \rceil \) bins to pack them. In any packing, if a bin contains 3 regular items, then among which at most one item of \( N_r \) is included. Note that there are \( p - (k_2 + 1) = k_2 + 3k_3 \) regular items outside \( N_r \). Based on whether all items in \( N_r \) are able to be packed into bins with 3 regular items (one item in \( N_r \) and two regular items outside \( N_r \)), the proof proceeds with the following two cases.

(i) \( \frac{k_2 + 3k_3}{2} \geq |N_r| = k_2 + 1 \). This condition is equivalent to \( k_2 \leq 3k_3 - 2 < 3k_3 \). Then under every valid packing, the number of bins that all the \( p = 2k_2 + 3k_3 + 1 \) regular items take is at least \( \lceil \frac{p}{3} \rceil \). We are done by noting that

\[
\frac{p/3}{(k_2 + k_3)} = \frac{2}{3} + \frac{k_3 + 1}{3(k_2 + k_3)} > \frac{2}{3} + \frac{1}{3(k_2^2 + 1)} > \frac{2}{3} + \frac{2}{12} = \frac{3}{4}.
\]

(ii) \( \frac{k_2 + 3k_3}{2} < |N_r| \). In this case, we can pack at most \( \frac{k_2 + 3k_3}{2} \) items of \( N_r \) into bins with 3 regular items. The most efficient way using the least number of bins to pack the regular items is to share one item from \( N_r \) with other two regular items out of \( N_r \), and then pack the remaining items in \( N_r \) in pairs. Suppose there are \( q \) items from \( N_r \) packed into bins of 3 regular items. It is easy to see that \( q \leq \frac{k_2 + 3k_3}{2} \). Except these \( q \) regular items, all the other \( k_2 + 1 - q \) items in \( N_r \) will occupy at least \( \frac{k_2 + 1 - q}{2} \) bins. Then the total number of bins needed to pack all the regular items is at least

\[
\frac{k_2 + 1 - q}{2} + \frac{q + 2k_2 - (k_2 + 1) + 3k_3 + 1}{3} = \frac{5k_2 + 6k_3 + 3}{6} - \frac{q}{6} \geq \frac{5k_2 + 6k_3 + 3}{6} - \frac{k_2 + 3k_3}{12} = \frac{3}{4}(k_2 + k_3) + \frac{1}{2},
\]

where the above inequality follows from \( q \leq \frac{k_2 + 3k_3}{2} \).

Recall that \( k_1 + k_2 + k_3 = m - 1 \). We complete the proof by showing that

\[
k_1 + \left\lceil \frac{3}{4}(k_2 + k_3) \right\rceil = \left\lceil \frac{3}{4}(k_1 + k_2 + k_3) + \frac{1}{4}k_1 \right\rceil \geq \left\lceil \frac{3}{4}(m - 1) \right\rceil.
\]

\( \square \)

4 Construct Efficient Equilibria by BFD

In Section 3 we have shown that the LSB cost-sharing rule with \( \Lambda = \frac{3}{4} \) is already the best possible with \( PoS = 1 \) and \( PoA = \frac{4}{3} \) in the game. So one can not expect to reduce the inefficiency of equilibria in this sense. However, it would be favorable if a social planner can construct an NE packing in polynomial time, whose asymptotic approximation ratio is smaller than \( \frac{4}{3} \). In this case, the system can directly reach a stable and effective state under which no selfish item would like to move. In addition, it is desirable to compute an NE efficiently with an asymptotic approximation ratio comparable to the classic optimization algorithms for bin packing. To this end, we formally introduce a well-known bin packing algorithm BFD (Best Fit Decreasing), which works as follows.
Algorithm BFD. First reorder the items in non-increasing sizes such that item 1 $\succ$ item 2 $\succ$ \ldots $\succ$ item $n$. Then one by one in this order, the algorithm places the current item in a bin that it fits the tightest; if there is no existing bin with enough space for the item, then open a new bin and put it in.

In this section, we show that by setting $\Lambda = \frac{2}{3}$ in the bin packing games, even though its PoA is at least 1.5 as illustrated in the following example, the algorithm BFD always leads to NE packings whose PoA is $\frac{11}{9}$, as the asymptotic approximation ratio of BFD is $\frac{11}{9}$ [23].

**Example 1.** The item set contains $2k$ items with size $\frac{2}{3}$, $2k$ items with size $\frac{1}{3}$, and $3k$ items with size $\epsilon = \frac{1}{3k}$, where $k$ is a positive integer. In this instance, there exists a packing $\pi^*$ with $3k$ bins used as shown in Figure 2 and it is an NE. To see this, first note that the items with sizes $\frac{2}{3}$ and $\frac{1}{3}$ can not move; besides, the tiny items with size $\epsilon$ have no incentive to move since their costs are already zero. Meanwhile, the optimal packing only uses $m^* = 2k + 1$ bins as illustrated in Figure 3. Apparently,

$$\lim_{k \to \infty} \frac{m(\pi^*)}{m^*} = 1.5.$$  

Denote $\pi$ as the derived packing by the algorithm BFD. In the following, we show that $\pi$ is not just an NE (which in fact is easy to prove), but also a strong Nash equilibrium.

**Definition 3.** A packing is called a strong Nash equilibrium (strong NE) if no non-empty subset of items can change their bins simultaneously such that every one of this subset benefits from this changing.

**Theorem 4.** The packing $\pi$ produced by BFD is a strong NE under the LSB cost-sharing rule with $\Lambda = \frac{2}{3}$.

**Proof.** From the process of the BFD algorithm we know, that when every item $i \in N$ is considered, it is placed into a feasible bin with the highest height level. Given an arbitrary non-empty subset of items $S \subseteq N$. We show that no matter how the items in $S$ change their bins simultaneously, at least one item in $S$ cannot gain by doing this. Let $j \in S$ be the biggest item of $S$.

First consider the case that $j$ is not a bottom item of $\pi$. Since all other items which are bigger than $j$ don’t change their positions and $j$ is a non-bottom item, then $j$ has no incentive to move since it has already been placed in the best bin it fits and has the minimum possible cost-share. Now we turn to the case of $j$ being a bottom item. Suppose there are $m$ bins under packing $\pi$, and name them as $B_1, B_2, \ldots, B_m$ as their opening order under BFD algorithm, i.e., $B_1$ is the first opened bin, and $B_m$ is the last one. For $k = 1, 2, \ldots, m$, let $b_k$ be the bottom item of bin $B_k$. 

**Figure 2:** $m(\pi^*) = 3k$

**Figure 3:** $m^* = 2k + 1$
Then by the BFD algorithm, we have $b_1 > b_2 > \cdots > b_m$. Besides, we should note that when each bottom item $b_k$ ($k \leq m$) is considered by BFD, it can not be put into any existing bin $B_h$ with $h < k$. So we can see that after all items in $S$ change their bins, item $j$ is still a bottom item in the new resulted packing (denoted by $\pi'$). Now suppose under $\pi$, $j$ is the bottom of bin $B_k$, i.e., $j = b_k$; and under the new packing $\pi'$, $j \in B'$. If $j$ can decrease its cost by the simultaneous migration of $S$ (otherwise we are done), then combining with the cost-sharing rule with $\Lambda = \frac{2}{3}$, we know it must be the case that $s(B_k) < \frac{2}{3}$ (thus $B_k$ contains only one item with size at least $\frac{1}{3}$) and $s(B') > s(B_k)$. Denote by $i'$ the second biggest item in bin $B'$ of $\pi'$. We consider two possible scenarios.

(1) $s_\ell > \frac{1}{3}$. Under the original packing $\pi$, $i'$ must belong to some bin $B_h$ with $h < k$, because otherwise we should have $s(B_k) \geq \frac{2}{3}$ since $j$ and $i'$ can be packed into one same bin by BFD. Recalling that bottom item $j$ can not move into bin $B_h$, we have $i' \in S$. Then apparently, the cost-share of item $i'$ in $B'$ is no less than the previous cost-share in $B_h$. Thereby it has no incentive to move.

(2) $s_\ell \leq \frac{1}{3}$. Then $s_\ell + s(B_k) \leq 1$. This means, that when $i'$ was considered by the BFD algorithm, it was placed in bin $B_k$ or a better bin, with a cost no more than $\int_{s_j}^{s_k} f(x)dx$. So if $i' \in S$, then $i'$ can not reduce its cost by moving into the bin $B'$ of $\pi'$, done. If $i' \notin S$, then this is to tell that $j$ moves into the bin containing $i'$ and $i'$ remains in the bin. Recall that $j$ is the biggest item of $S$ and it is still a bottom item after the deviation. We know that under $\pi$, item $i' \in B_\ell$ with $\ell > k$. Combining with the fact that $s(B_k) < \frac{2}{3}$ and $j = b_k > b_\ell$, we deduce that there are two items in $B_\ell$ with size larger than $1 - s_j \geq 1 - s(B_k) > \frac{1}{3}$, since otherwise one of the two biggest items in $B_\ell$ should be packed into $B_k$ by the BFD. Therefore under $\pi$, bin $B_\ell$’s size $s(B_\ell) > \frac{2}{3}$ and the cost-share of $b_\ell$ is exactly $\int_0^{b_\ell} f(x)dx$. Besides, since $j$ will move into the bin $B_\ell$ in the deviation of $S$ and $i'$ will be the second biggest item in the new bin, thus $b_\ell \in S$. Let $j' \in S$ be a bottom item in $S$ with cost-share being exactly $\int_0^{b_j} f(x)dx$ under $\pi$ and the biggest such one. Then it can be easily deduced that $s_{j'} \geq s_{b_\ell} > 1 - s_j$, and $j'$ is still a bottom item under the new packing $\pi'$, which implies that its new cost-share under $\pi'$ is at least $\int_0^{b_j} f(x)dx$. So $j'$ can not decrease its cost-share by the changes.

Overall, we can conclude that under packing $\pi$, no group of items can change the bins simultaneously to make every one of them better off. It implies that $\pi$ is a strong NE. 

\section{Extension}

In this section, we study a variant of the classical selfish scheduling game \cite{9, 22, 5, 24, 2}, and show that the key idea of our cost-sharing rule can be adapted to design a best coordination mechanism for this considered scheduling game. As far as we know, previously there was no such close connection between cost-sharing-rule design in a resource selection game and scheduling-policy design in a scheduling game.

In this variant scheduling game, given sufficiently many identical machines, there are $n$ jobs, each with a processing time $p_j$ for job $j \in N = \{1, 2, \ldots, n\}$. However, the machines have a fixed working time window $[d, d + T]$, where $T \geq \max_{j \in N} p_j$. To interpret this, one can imagine that the machines are controlled by crew members who have a fixed working period in a day. Each job is owned by an independent user whose goal is to minimize its own completion time by selecting a machine before the beginning time $d$. The scheduling policies on the machines thus become crucial.
As all jobs have selected their machines, we get an assignment, which is valid if the total processing time of jobs on every machine does not exceed $T$. It is important to emphasize that a scheduling policy only works if the assignment is valid, otherwise the overloaded machines will crash. The social objective of the game is to minimize the number of machines used. We remark here that the scheduling models with the objective of minimizing the number of used machines are also an important research topic, see literature [10, 7] for reference.

Apparently, if the social planner can control all the jobs, then from an optimization point of view, it is exactly the problem of bin packing optimization. However, from the game point of view, this selfish scheduling model (referred as selfish capacitated scheduling in this section) is quite different from the aforementioned selfish bin packing. In this game, each machine does not distribute a fixed cost among the jobs it processes but decides the starting (or completion) times for them. And the jobs only care about their own completion times. The existing cost-sharing rules in literature for selfish bin packing can not be applied to designing the scheduling policies.

Let us first look at what challenges we may face to guarantee efficient Nash equilibria. Note that one job may prefer to occupy a machine alone, rather than stay with other jobs. It results in a large efficiency loss. Namely, as an example, if every machine always processes jobs from the very beginning (time $d$), then the unique Nash equilibrium would be one machine for one job, which is the worst. In the following, we shall focus on coordination mechanisms under which no job would like to unilaterally move to an empty machine. Such coordination mechanisms are called reasonable. Then the general lower bound $4/3$ on $POA$ in [13] still holds for reasonable mechanisms.

**Proposition 3 ([13]).** For selfish capacitated scheduling, any reasonable coordination mechanism has a $PoA$ at least $4/3$.

In the following, we will propose a best possible scheduling policy for this game with $PoS = 1$ and $PoA \leq 4/3$. The policy will be denoted by $0^{\text{ShortestFirst}}$ and is adopted by every machine. The whole coordination mechanism first re-indexes the jobs in the non-decreasing order of their processing times (breaking ties arbitrarily) and fixes this order. Then for each machine $m$, suppose the valid job set it receives is $J_m$. The machine stays idle until time $s_m = d + (T - \sum_{j \in J_m} p_j)$. Then from the time $s_m$ on, the machine processes the jobs continuously in the non-decreasing order of jobs’ processing times (breaking the ties as predefined above). In fact, we can view the idle times of a machine as being processing “null jobs”. This scheduling policy can also be denoted as $0^{\text{LongestLast}}$.

It is obvious that the coordination mechanism with $0^{\text{ShortestFirst}}$ policy is reasonable since no job favors to be completed by the deadline $d + T$. Besides, in this coordination mechanism, every job is willing to select a machine with a larger total processing time of “longer” jobs. In this case, the job can be processed earlier. This is quite similar to the aforementioned selfish bin packing game with $\Lambda = 1$, where every job prefers to be packed into a bin with a higher height level. However, in the latter game, since each bin’s bottom item has to pay the remaining uncovered cost, the bottom item also prefers to stay in a bin with more content. Now we will try to modify the latter game to make the two games equivalent.

We consider relaxing the budget-balance constraint of $\sum_{i \in B} c(i) = 1$ to the constraint $\sum_{i \in B} c(i) \leq 1$ in the selfish bin packing game. Then it is not hard to find that by viewing jobs as items, the normalized processing times of jobs as items’ sizes, and machines as bins, then the games of selfish capacitated scheduling under $0^{\text{ShortestFirst}}$ policy, are essentially equivalent to the games of selfish bin packing under the following cost-sharing rule, with respect to their sets of Nash equilibria and NE’s inefficiency:

1. Let $f(x) = 2 - 2x$ with $x \in [0, 1]$.

15
2. Given a packing $\pi$, for each bin $B$, suppose w.l.o.g. that $B = \{i_1, i_2, \ldots, i_l\}$ and $i_1 > i_2 > \cdots > i_l$. Let $S^j_B := \sum_{h=1}^j s_{ih}$ for $j \leq l$. Then the cost-share of item $i_j$ for $j = 1, 2, \ldots, l$ is

$$c_{i_j}(\pi) = \begin{cases} \int_0^{S^j_B} f(x)dx, & \text{for } j = 1; \\ \int_{S^j_B}^{S^j_B+1} f(x)dx, & \text{for } j > 1. \end{cases}$$

This bin packing game is denoted by $\mathcal{G}_{\text{bin}}$. To avoid introducing more definitions and simplify the analysis, in the following we will focus on studying the bin packing game $\mathcal{G}_{\text{bin}}$ and follow the notations defined previously. The results derived below can be adapted to the selfish capacitated scheduling under the $^0\text{ShortestFirst}$ policy.

**Observation 5.1.** The above cost-sharing rule of game $\mathcal{G}_{\text{bin}}$ is local, monotone, fair, and polynomial-time computable, but not budget-balanced.

Now we show that $\mathcal{G}_{\text{bin}}$ always has an NE being an optimal packing, and its PoA is the best possible.

**Theorem 5.** For game $\mathcal{G}_{\text{bin}}$, it’s PoS = 1.

**Proof.** Assume that the items in $N$ are $a_1 > a_2 > \cdots > a_n$. Let $P^*$ be the set of the optimal packings which use the minimum number of bins. For two packings $\pi, \pi' \in P^*$, we say that $\pi$ is lexicographically higher than $\pi'$ if the height levels satisfy that, for some $\ell \leq n$ we have, $l_{a_h}(\pi) = l_{a_h}(\pi')$ for every $h < \ell$ and $l_{a_\ell}(\pi) > l_{a_\ell}(\pi')$. Since $P^* \neq \emptyset$, then there must exist a packing in $P^*$ that is lexicographically highest, which is denoted by $\pi^*$. Note that under $\pi^*$, no item in $N$ can increase its height level (thus reduce its cost) by unilaterally moving into another feasible bin. This is because otherwise, it contradicts the definition of $\pi^*$ as the deviation will not change the height levels of all the bigger items. Therefore $\pi^*$ is exactly an NE.

**Theorem 6.** For game $\mathcal{G}_{\text{bin}}$, it’s PoA = 4/3.

**Proof sketch of Theorem 6.** The key idea of proving Theorem 6 is same as the analysis in Section 3.2. However, since the cost-sharing rule of game $\mathcal{G}_{\text{bin}}$ is different from the rule in selfish bin packing with $\Lambda = \frac{3}{4}$, its proof still has some critical differences, which will be listed below.

Given an NE packing $\pi$ of game $\mathcal{G}_{\text{bin}}$. Suppose without loss of generality that $\pi$ is composed of $m$ nonempty bins $B_1, B_2, \ldots, B_m$ with $s(B_1) \geq s(B_2) \geq \cdots \geq s(B_m)$. Again, for each bin $B_k$, we use $b_k$ to denote its bottom item, and $t_k$ to denote its minimum item. First, as a result of $f(x)$ being a decreasing function on the whole interval $[0, 1]$, we have the following stronger conclusion (compared with Lemma 3.1) about the infeasibility of the smallest item of one bin moving into another bin.

**Lemma 5.1.** For every two bins $B_j$ and $B_k$ with $j < k$, we have $s(B_j) + s_{t_k} > 1$.

Consequently, the following corollary is easy to get.

**Corollary 2.** Except for the last bin, every other bin $B_j$’s size $s(B_j) > 1/2$, where $j \leq m - 1$.

Next, we still try to prove that the optimal packing uses at least $\left\lfloor \frac{3m}{4} \right\rfloor$ bins, which implies $PoA \leq 4/3$. However, since we can not guarantee the correctness of Lemma 3.2 in game $\mathcal{G}_{\text{bin}}$, we have a slightly different assumption base and then further arguments. Note that if $s(B_{m-3}) \geq \frac{3}{4}$, then the total size of the $n$ items is at least $\frac{3}{4} \cdot (m-3) + s(B_{m-2}) + s(B_{m-1}) + s(B_m) > \frac{3}{4} \cdot (m-
3) \( s(B_{m-2}) + 1 > \frac{3m}{4} - \frac{2}{3} \), which still implies the optimal packing using at least \( \left\lceil \frac{3m}{4} \right\rceil \) bins. So in the following, we will assume that \( s(B_{m-3}) < \frac{3}{4} \).

Different from the definition in Section 3.2, we denote \( L^* := s(B_{m-2}) \leq s(B_{m-3}) < \frac{3}{4} \). For any item \( i \in N \), if its size \( s_i \geq L^* \), we call it a super item; if \( L^* > s_i > 1 - L^* \), we call it a regular item; otherwise \( s_i \leq 1 - L^* \), we call it a tiny item.

**Lemma 5.2.** There are no tiny items in the last three bins \( B_{m-2}, B_{m-1} \) and \( B_m \).

**Proof.** The statement’s correctness for the last two bins \( B_{m-1}, B_m \) is apparent by Lemma 5.1. For bin \( B_{m-2} \), as a result of \( s(B_{m-3}) < \frac{3}{4} \) and Lemma 5.1, we have \( s_t_{m-2} > 1/4 \). Thus there are at most two items in \( B_{m-2} \). If \( s_t_{m-2} \) is not a tiny item, then we are done. Otherwise, \( s_t_{m-2} < s_t_{m-1} \) and \( s_t_{m-2} + s(B_{m-1}) \leq s_t_{m-2} + L^* \leq 1 \). Recall that \( \pi \) is an NE. So item \( t_{m-2} \) has no incentive to move into \( B_{m-1} \). This means that the bottom item of \( B_{m-2} \) satisfy \( s_{b_{m-2}} \geq s(B_{m-1}) > 1/2 \). Then \( S(B_{m-2}) = s_{b_{m-2}} + s_t_{m-2} > 1/2 + 1/4 \), contradicting the fact \( s(B_{m-2}) < 3/4 \).

In the following, by focusing on all the non-tiny items in the first \((m - 2)\) bins, and using a similar analysis as the counterpart’s in Section 3.2, we can show that packing all the non-tiny items in the first \((m - 2)\) bins and the items in \( B_{m-1} \) and \( B_m \) needs at least \( \left\lceil \frac{3m}{4} - \frac{2}{3} \right\rceil \geq \left\lceil \frac{3m}{4} \right\rceil - \frac{2}{3} \) bins. \( \square \)

We have shown in Section 3 that, for the game of selfish bin packing, by setting constant \( \Lambda = \frac{3}{4} \) in the LSB cost-sharing rule, we can guarantee its PoS = 1 and PoA = \( \frac{4}{3} \). However, if we want to compute an NE in polynomial time with PoA smaller than \( \frac{3}{4} \), we have to reset \( \Lambda = \frac{3}{2} \) in the game, but with the PoA \( \geq \frac{3}{2} \), as illustrated in Section 4. But for game \( \mathcal{G}_{bin} \), we can achieve all the desired results simultaneously.

**Theorem 7.** For game \( \mathcal{G}_{bin} \), the packing \( \pi \) produced by the algorithm BFD is a strong NE.

**Proof.** Given an arbitrary subset of items \( S \subseteq N \). Let \( i \in S \) be the biggest item of \( S \). We show that no matter how the items in \( S \) change their bins simultaneously, item \( i \) can not reduce its cost by these deviations. From the process of BFD algorithm we know, every item has been placed into the best possible bin if all other bigger items don’t change their positions. By recalling the cost-sharing rule of game \( \mathcal{G}_{bin} \), no matter whether \( i \) is a bottom item or not, its cost share can not be reduced after the deviations of \( S \). \( \square \)

**Remark 2.** Even though the game \( \mathcal{G}_{bin} \) is introduced as a tool to study the game of selfish capacitated scheduling in this paper, it has an independent value as a bin packing game. Note that in classical selfish bin packing, the social planner’s aim is to minimize the number of bins used, not to earn revenue. So the budget-balance constraint needs not to be strict. Instead, one can replace the constraint \( \sum_{i \in B} c(i) = 1 \) with \( \sum_{i \in B} c(i) \leq 1 \) in the game. If \( \sum_{i \in B} c(i) < 1 \) for a bin \( B \) in some instance, the social planner can pay the remaining cost of \( 1 - \sum_{i \in B} c(i) \).

**Theorem 8.** For selfish capacitated scheduling game with ShortestFirst policy, its PoS = 1, PoA = \( \frac{4}{3} \), and the algorithm BFD leads to a strong NE.

6 Conclusion

This paper proposes a best cost-sharing rule for selfish bin packing, assuming that no items in a bin are willing to afford the whole cost by moving to an empty bin. To break the lower bound of 4/3, one has to redefine the game, allowing negative payments or some other selfish behaviors. One open question this work still leaves is the NE existence (and also the exact PoS) for bin packing.
games with $\Gamma > \frac{3}{2}$. We believe that the framework of our cost-sharing rule may shed light on many other variants of selfish bin packing and help design good mechanisms/protocols for more general resource selection games.

References

[1] Elliot Anshelevich, Anirban Dasgupta, Jon Kleinberg, Éva Tardos, Tom Wexler, and Tim Roughgarden. The price of stability for network design with fair cost allocation. *SIAM Journal on Computing*, 38(4):1602–1623, 2008.

[2] Yossi Azar, Lisa Fleischer, Kamal Jain, Vahab Mirrokni, and Zoya Svitkina. Optimal coordination mechanisms for unrelated machine scheduling. *Operations Research*, 63(3):489–500, 2015.

[3] Vittorio Biló. On the packing of selfish items. In *Proceedings 20th IEEE International Parallel & Distributed Processing Symposium (IPDPS)*, pages 9–pp. IEEE, 2006.

[4] Zhigang Cao and Xiaoguang Yang. Selfish bin covering. *Theoretical Computer Science*, 412(50):7049–7058, 2011.

[5] Ioannis Caragiannis. Efficient coordination mechanisms for unrelated machine scheduling. *Algorithmica*, 66(3):512–540, 2013.

[6] Ho-Lin Chen, Tim Roughgarden, and Gregory Valiant. Designing network protocols for good equilibria. *SIAM Journal on Computing*, 39(5):1799–1832, 2010.

[7] Lin Chen, Nicole Megow, and Kevin Schewior. An $O(\log m)$-competitive algorithm for online machine minimization. *SIAM Journal on Computing*, 47(6):2057–2077, 2018.

[8] Xin Chen, Qingqin Nong, and Qizhi Fang. An improved mechanism for selfish bin packing. *Journal of Combinatorial Optimization*, 42(3):636–656, 2021.

[9] George Christodoulou, Elias Koutsoupias, and Akash Nanavati. Coordination mechanisms. *Theoretical Computer Science*, 410(36):3327–3336, 2009.

[10] Julia Chuzhoy, Sudipto Guha, Sanjeev Khanna, and Joseph Seffi Naor. Machine minimization for scheduling jobs with interval constraints. In *Proceedings of the 45th annual IEEE symposium on foundations of computer science (FOCS)*, pages 81–90. IEEE, 2004.

[11] Richard Cole, José R Correa, Vasilis Gkatzelis, Vahab Mirrokni, and Neil Olver. Decentralized utilitarian mechanisms for scheduling games. *Games and Economic Behavior*, 92:306–326, 2015.

[12] György Dósa and Leah Epstein. Quality of equilibria for selfish bin packing with cost sharing variants. *Discrete Optimization*, 38:100556, 2020.

[13] György Dósa, Hans Kellerer, and Zsolt Tuza. Using weight decision for decreasing the price of anarchy in selfish bin packing games. *European Journal of Operational Research*, 278(1):160–169, 2019.

[14] Leah Epstein and Elena Kleiman. Selfish bin packing. *Algorithmica*, 60(2):368–394, 2011.

[15] Leah Epstein and Elena Kleiman. Selfish vector packing. *Algorithmica*, 83(9):2952–2988, 2021.

[16] Leah Epstein, Elena Kleiman, and Julián Mestre. Parametric packing of selfish items and the subset sum algorithm. *Algorithmica*, 74(1):177–207, 2016.

[17] W. Vega Fernandez de la and G. S. Lueker. Bin packing can be solved within $1 + \varepsilon$ in linear time. *Combinatorica*, 1(4):349–355, 1981.

[18] Vasilis Gkatzelis, Konstantinos Kollias, and Tim Roughgarden. Optimal cost-sharing in general resource selection games. *Operations Research*, 64(6):1230–1238, 2016.
[19] Tobias Harks, Martin Hoefer, Anja Schedel, and Manuel Surek. Efficient black-box reductions for separable cost sharing. Mathematics of Operations Research, 46(1):134–158, 2021.

[20] Tobias Harks and Philipp von Falkenhausen. Optimal cost sharing for capacitated facility location games. European Journal of Operational Research, 239(1):187–198, 2014.

[21] Rebecca Hoberg and Thomas Rothvoss. A logarithmic additive integrality gap for bin packing. In Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 2616–2625. SIAM, 2017.

[22] Nicole Immorlica, Li Ernan Li, Vahab S Mirrokni, and Andreas S Schulz. Coordination mechanisms for selfish scheduling. Theoretical Computer Science, 410(17):1589–1598, 2009.

[23] D. S. Johnson, A. Demers, J. D. Ullman, M. R. Garey, and R. L. Graham. Worst-case performance bounds for simple one-dimensional packing algorithms. SIAM Journal on Computing, 3(4):299–325, 1974.

[24] Konstantinos Kollias. Nonpreemptive coordination mechanisms for identical machines. Theory of Computing Systems, 53(3):424–440, 2013.

[25] Elias Koutsoupias and Christos Papadimitriou. Worst-case equilibria. In Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science (STACS), pages 404–413. Springer, 1999.

[26] Ruixin Ma, György Dósa, Xin Han, Hing-Fung Ting, Deshi Ye, and Yong Zhang. A note on a selfish bin packing problem. Journal of Global Optimization, 56(4):1457–1462, 2013.

[27] Q.Q. Nong, T. Sun, T.C.E. Cheng, and Q.Z. Fang. Bin packing game with a price of anarchy of 3/2. Journal of Combinatorial Optimization, 35(2):632–640, 2018.

[28] Tim Roughgarden and Éva Tardos. How bad is selfish routing? Journal of the ACM, 49(2):236–259, 2002.

[29] Philipp von Falkenhausen and Tobias Harks. Optimal cost sharing for resource selection games. Mathematics of Operations Research, 38(1):184–208, 2013.

[30] Guosong Yu and Guochuan Zhang. Bin packing of selfish items. In Proceedings of the 4th International Workshop on Internet and Network Economics (WINE), pages 446–453. Springer, 2008.

[31] Chenhao Zhang and Guochuan Zhang. From packing rules to cost-sharing mechanisms. Journal of Combinatorial Optimization, pages 1–16, 2020.