On First Fit Bin Packing for Online Cloud Server Allocation

Xueyan Tang, Yusen Li, Runtian Ren, and Wentong Cai
School of Computer Engineering
Nanyang Technological University
Singapore 639798
Email: {asxytang, S080007, RENR0002, aswtcai}@ntu.edu.sg

***Abstract*—Cloud-based systems often face the problem of dispatching a stream of jobs to run on cloud servers in an online manner. Each job has a size that defines the resource demand for running the job. Each job is assigned to run on a cloud server upon its arrival and the job departs after it completes. The departure time of a job, however, is not known at the time of its arrival. Each cloud server has a fixed resource capacity and the total resource demand of all the jobs running on a server cannot exceed its capacity at all times. The objective of job dispatching is to minimize the total cost of the servers used, where the cost of renting each cloud server is proportional to its running hours by "pay-as-you-go" billing. The above job dispatching problem can be modeled as a variant of the Dynamic Bin Packing (DBP) problem known as MinUsageTime DBP. In this paper, we develop new approaches to the competitive analysis of the commonly used First Fit packing algorithm for the MinUsageTime DBP problem, and establish a new upper bound of $\mu + 4$ on the competitive ratio of First Fit packing, where μ is the ratio of the maximum job duration to the minimum job duration. Our result significantly reduces the gap between the upper and lower bounds for the MinUsageTime DBP problem to a constant value independent of μ, and shows that First Fit packing is near optimal for MinUsageTime DBP.**

I. Introduction

The standard Dynamic Bin Packing (DBP) problem [6] considers a set of items, each having an arrival time and a departure time. The items are to be packed into bins in an online manner such that the total size of the items in each bin does not exceed the bin capacity at all times. A bin is opened when it receives the first item and is closed when all items in the bin depart. The objective of DBP is to minimize the maximum number of concurrently open bins in the packing process.

In this paper, we consider a variant of the DBP problem. We focus on the duration of each bin's usage, i.e., the period from its opening to its closing. Our objective is to pack the items into bins to minimize the accumulated bin usage time. We refer to this variant of the DBP problem as *the MinUsageTime DBP problem* [16]. This problem is motivated by the online job dispatching problem arising from many cloud-based systems in which jobs may arrive at arbitrary times. Each job needs some amount of resources for execution and is assigned to run on a cloud server upon its arrival. The departure time of the job, however, is not known at the time of its arrival. The job is not reassigned to other servers during execution due to reasons such as high migration overheads and penalty. Each cloud server has a fixed resource capacity that restricts the total amount of resources needed by all the jobs running on the server at any time. The objective of job dispatching is to minimize the total cost of the servers used. The on-demand server instances (virtual machines) rented from public clouds such as Amazon EC2 are normally charged according to their running hours by "pay-as-you-go" billing [1]. Therefore, to minimize the total renting cost, it is equivalent to minimize the total running hours of the cloud servers. Such a job dispatching problem can be modeled exactly by the MinUsageTime DBP problem defined above, where the jobs and cloud servers correspond to the items and bins respectively.

A typical application of the preceding job dispatching problem is cloud gaming. In a cloud gaming system, games are run and rendered on cloud servers, while players interact with the games via networked thin clients [10], [14]. Running each game instance demands a certain amount of GPU resources. When a play request is received by the cloud gaming provider, it should be assigned to a cloud server that has enough GPU resources to run the requested game instance. Several game instances can share the same cloud server provided that the server's GPU resources are not saturated. Each game instance keeps running on the assigned server until the player stops the game. Migrating a game instance from one server to another during execution is usually not allowed due to interruption to game play. Cloud gaming providers such as GaiKai rent servers from public clouds to run game instances [19]. Then, a natural problem faced by the cloud gaming provider is how to dispatch the play requests to cloud servers to minimize the total renting cost of the servers used.

Previous work on MinUsageTime DBP. The MinUsageTime DBP problem was first proposed in our earlier work [15], [16]. Any online bin packing algorithm can be applied to the problem. In [15], [16], we analyzed the competitiveness of several classical bin packing algorithms, including Any Fit family of algorithms (which open a new bin only when no current open bin can accommodate an incoming item), First Fit and Best Fit (which are two particular Any Fit algorithms). We proved that the competitive ratio of any

Any Fit packing algorithm cannot be better than $\mu+1$, where μ is the ratio of the maximum item duration to the minimum item duration. The competitive ratio of Best Fit packing is not bounded for any given μ. If all the item sizes are smaller than $\frac{1}{\beta}$ of the bin capacity ($\beta > 1$ is a constant), the competitive ratio of First Fit packing has an upper bound of $\frac{\beta}{\beta-1} \cdot \mu + \frac{3\beta}{\beta-1} + 1$. For the general case, the competitive ratio of First Fit has an upper bound of $2\mu + 7$. We further proposed a Hybrid First Fit algorithm that classifies and packs items based on their sizes to achieve a competitive ratio no larger than $\frac{8}{7}\mu + \frac{55}{7}$ [16]. We also indicated a lower bound of μ on the competitive ratio of any online packing algorithm [16]. Kamali *et al.* [12] later presented a formal proof of this lower bound. They also showed that Next Fit packing has a competitive ratio bounded above by $2\mu + 1$.

Contributions of this paper. In this paper, we significantly tighten the gap between the upper and lower bounds for the MinUsageTime DBP problem, reducing the gap to a constant value independent of μ. We develop new approaches to improve the competitive analysis of First Fit packing, including dividing and consolidating bin usage periods based on item arrivals (Sections V and VI), and compensating for low utilization periods by exploiting high utilization periods to bound the amortized bin utilization (Section VII). Our analysis establishes a new upper bound of $\mu + 4$ on the competitive ratio of First Fit packing, regardless of the item sizes. This new bound is the current best upper bound for the MinUsageTime DBP problem.[1] It is the first known upper bound that has a multiplicative factor 1 for μ, whereas all the aforementioned upper bounds have multiplicative factors larger than 1 for μ. Our result indicates that First Fit packing is near optimal for the MinUsageTime DBP problem.

The rest of this paper is organized as follows. Section II reviews the related work. Section III provides some preliminaries. Sections IV to VII carry out the competitive analysis of First Fit packing. Section VIII compares the competitiveness of First Fit packing and Next Fit packing, and shows that the later is inherently worse. Finally, Section IX concludes the paper.

II. Related Work

The classical bin packing problem aims to pack a set of items into the minimum number of bins. It is well known that even the offline version of classical bin packing is NP-hard [8]. In the online version, each item must be placed in a bin without the knowledge of subsequent items. Once placed, the items are not allowed to move to other bins. The competitive ratios of various algorithms for classical online bin packing have been extensively studied [2], [18].

Dynamic bin packing (DBP) is a generalization of the classical bin packing problem [6]. In DBP, items may arrive and depart at arbitrary times. The objective is to minimize the maximum number of bins concurrently used in the packing. A large amount of research work has also been done to analyze the competitive ratios of various algorithms for DBP [4], [5], [6], [11]. However, standard DBP does not consider the duration of bin usage. In contrast, the MinUsageTime DBP problem we have defined aims to minimize the total amount of time the bins are used [16].

Interval scheduling [13] is another problem related to our MinUsageTime DBP problem. In the basic interval scheduling, each job is associated with one or several alternative time intervals for execution. The goal of scheduling is to maximize the number of jobs executed on a server that can process only a single job at any time [9], [20]. Recently, some works have studied interval scheduling with bounded parallelism, where each server can process multiple jobs simultaneously up to a fixed maximum number [7], [17]. A server is considered busy if at least one job is running on it. The objective is to minimize the total busy time of all servers to complete a given set of jobs. This target resembles the one we study in this paper. However, there is a crucial difference between interval scheduling and our MinUsageTime DBP problem. The ending times of jobs are known in interval scheduling, but the departure time of an item is not known at the time of its packing in our problem.

III. Preliminaries

A. Notations and Definitions

We first define some key notations used in this paper. For any time interval I, we use I^- and I^+ to denote the left and right endpoints of I respectively. For technical reasons, we shall view intervals as half-open, i.e., $I = [I^-, I^+)$. Let $l(I) = I^+ - I^-$ denote the length of the time interval I.

For any item r, let $I(r)$ denote the time interval from r's arrival to its departure. We say that item r is *active* during the interval $I(r)$, and we refer to the length of $I(r)$ as the *item duration*. Let $s(r)$ denote the size of item r. For notational convenience, for a list of items $\mathcal{R}$, we also use $s(\mathcal{R})$ to denote the total size of all the items in $\mathcal{R}$, i.e., $s(\mathcal{R}) = \sum_{r \in \mathcal{R}} s(r)$. In addition, we refer to the time duration in which at least one item in $\mathcal{R}$ is active as the *span* of $\mathcal{R}$ and denote it by $span(\mathcal{R})$ (see Figure 1).

B. First Fit Packing Algorithm

In the bin packing process, a bin is *opened* when it receives the first item. When all the items in a bin depart, the bin is *closed*. At any time, the total size of all the active items in an open bin is referred to as the *bin level*.

We consider the following First Fit algorithm for online bin packing. Each time when a new item arrives, if there are one or more open bins that can accommodate the new item, First Fit places the item in the bin which was opened

[1] Hybrid First Fit and Next Fit packing algorithms that classify items based on their sizes can achieve competitive ratios of $\mu + 5$ [15] and $\mu + 2$ [12] respectively, but to do so, these algorithms require the max/min item duration ratio μ to be known a priori and thus are semi-online in nature.

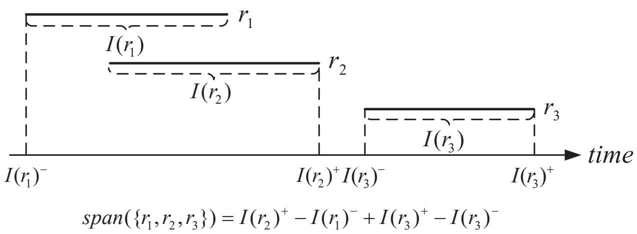

Figure 1. Span of an item list

earliest among these bins. Otherwise, if no open bin can accommodate the new item, then a new bin is opened to receive the item.

C. Competitive Ratio

The performance of an online algorithm is usually measured by its *competitive ratio*, i.e., the worst-case ratio between the solution constructed by the algorithm and an optimal solution [3].

Without loss of generality, we assume that the bins all have unit capacity. Given a list of items $\mathcal{R}$, let $OPT(\mathcal{R}, t)$ denote the minimum achievable number of bins into which all the active items at time t can be repacked. Then, the total bin usage time of an optimal offline adversary that can repack everything at any time is given by

$$OPT_{total}(\mathcal{R}) = \int_{\bigcup_{r\in\mathcal{R}} I(r)} OPT(\mathcal{R}, t)\ \mathrm{d}t,$$

where $\bigcup_{r\in\mathcal{R}} I(r)$ is the packing period, i.e., the time interval from the first item arrival to the last item departure in $\mathcal{R}$. As shown in our earlier work [15], [16], it is easy to obtain the following lower bounds on $OPT_{total}(\mathcal{R})$:

Proposition 1: $OPT_{total}(\mathcal{R}) \geq \sum_{r\in\mathcal{R}} \Big(s(r) \cdot l(I(r))\Big)$.

Proposition 2: $OPT_{total}(\mathcal{R}) \geq span(\mathcal{R})$.

The first bound is derived by assuming that no capacity of any bin is wasted at any time, where $s(r)\cdot l(I(r))$ is the *time-space demand* of an item r. The second bound is derived from the fact that at least one bin must be used at any time when at least one item is active.

Let $FF_{total}(\mathcal{R})$ denote the total bin usage time by applying First Fit packing to the list of items $\mathcal{R}$. The competitive ratio of First Fit packing is the maximum ratio of $FF_{total}(\mathcal{R})/OPT_{total}(\mathcal{R})$ over all instances of item lists $\mathcal{R}$. A standard approach to deriving bounds on the competitive ratio is to prove the following relation for all $\mathcal{R}$: $FF_{total}(\mathcal{R}) \leq \alpha \cdot OPT_{total}(\mathcal{R})$, where α is a constant [6]. Then, the competitive ratio of First Fit packing is bounded above by α.

IV. Bin Usage Periods

Now, we start the competitive analysis of First Fit packing for the MinUsageTime DBP problem. For any list of items $\mathcal{R}$, let $\mu = \frac{\max_{r\in\mathcal{R}} l(I(r))}{\min_{r\in\mathcal{R}} l(I(r))}$ denote the ratio of the maximum item duration to the minimum item duration among all the items in $\mathcal{R}$. Without loss of generality, we shall assume that the minimum item duration is 1, and the maximum item duration is μ ($\mu \geq 1$).

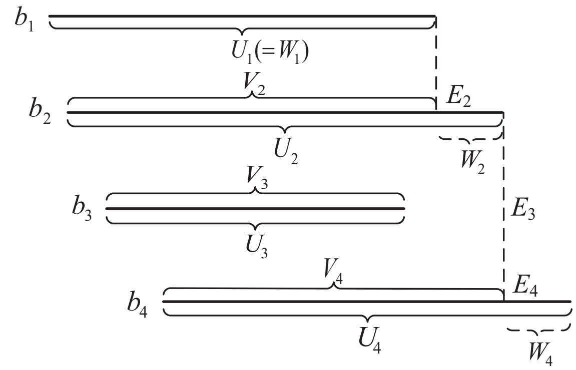

Figure 2. An example of usage periods

Suppose a total of m bins b_1,b_2,...,b_m are used by First Fit packing to pack a list of items $\mathcal{R}$. For each bin b_k, let $U_k = [U_k^-, U_k^+)$ denote the usage period of b_k, i.e., the period from the time when b_k is opened to the time when b_k is closed. Then, the total bin usage time of First Fit packing is given by the total length of the usage periods of all the bins used, i.e., $FF_{total}(\mathcal{R}) = \sum_{k=1}^{m} l(U_k)$.

Without loss of generality, assume that the bins are indexed in the temporal order of their openings, i.e., $U_1^- \leq U_2^- \leq \cdots \leq U_m^-$. For each bin b_k, let E_k be the latest closing time of all the bins that are opened before b_k, i.e., $E_k = \max\{U_i^+ | 1 \leq i < k\}$. For the first bin b_1, define $E_1 = U_1^-$. We divide the usage period U_k of each bin into two parts: V_k and W_k. V_k is the period $[U_k^-, \min\{U_k^+, E_k\})$. If $E_k \leq U_k^-$, define $V_k = \emptyset$. $W_k = U_k - V_k$ is the remaining period. Figure 2 shows an example of these definitions.

According to the definitions, we have $l(U_k) = l(V_k) + l(W_k)$. Apparently, for any two different bins b_{k_1} and b_{k_2}, $W_{k_1} \cap W_{k_2} = \emptyset$. It is also easy to see that $span(\mathcal{R}) = l(\bigcup_{k=1}^{m} W_k) = \sum_{k=1}^{m} l(W_k)$. Therefore,

$$\begin{aligned} FF_{total}(\mathcal{R}) &= \sum_{k=1}^{m} l(U_k) \\ &= \sum_{k=1}^{m} \Big(l(V_k) + l(W_k)\Big) = \left(\sum_{k=1}^{m} l(V_k)\right) + span(\mathcal{R}) \\ &\leq \left(\sum_{k=1}^{m} l(V_k)\right) + OPT_{total}(\mathcal{R}), \end{aligned} \tag{1}$$

where the last step follows from Proposition 2. In what follows, we shall analyze the amortized bin levels over the periods V_k to bound the total time-space demand of all items in terms of $\sum_{k=1}^{m} l(V_k)$ and then derive the competitive ratio of First Fit based on Proposition 1.

V. CREATION AND CONSOLIDATION OF SUBPERIODS

We classify items into small and large ones according to their sizes. Items of sizes less than $\frac{1}{\mu+3}$ are called *small items*, while those of sizes greater than or equal to $\frac{1}{\mu+3}$ are called *large items*, where μ is the max/min item duration ratio.

For each bin b_k, we choose a set of small items that are placed in b_k during period V_k. As illustrated in Figure 3, we start by selecting the first small item ever placed in bin b_k and then select subsequent items by repeating the following process. Given the current selected item r, if there are other small items placed in bin b_k within a duration μ (including μ) after r's arrival, the next item selected is the *last* item among these small items. Otherwise, if no other small item is placed in bin b_k within a duration μ after r's arrival, the next item selected is the *first* small item placed in b_k after r. The selection process terminates once (i) a small item arriving within a duration μ (including μ) before the end of period V_k is chosen, or (ii) the last small item arriving in period V_k is chosen.

As shown in Figure 3, the arrival times of the small items selected divide V_k into a set of periods x_0, x_1, x_2, x_3, $\ldots$, where x_0 is the period before the arrival of the first small item, and x_i $(i \geq 1)$ is the period between the arrival times of the i-th and $(i+1)$-th small items selected. Note that if no small item is ever placed in bin b_k during period V_k, we have $x_0 = V_k$. For each period x_i $(i \geq 1)$, if its length $l(x_i) > \mu$, we further split x_i into an l-subperiod $x_{l,i}$ and an h-subperiod $x_{h,i}$, where the length of the l-subperiod is $l(x_{l,i}) = \mu$ and the remaining period $x_{h,i} = x_i - x_{l,i}$ is the h-subperiod. We also rewrite x_0 as $x_{h,0}$ and refer to it as an h-subperiod. Recall that the maximum item duration is μ. Based on the above process of item selection, there cannot be any small item staying in bin b_k during the h-subperiods. This implies that the bin level must be at least $\frac{1}{\mu+3}$ high during the h-subperiods (where 'h' stands for high utilization). In contrast, the bin level is likely to be lower than $\frac{1}{\mu+3}$ during the l-subperiods (where 'l' stands for potentially low utilization). For notational convenience, if the length of period x_i does not exceed μ, we define $x_{l,i} = x_i$ and $x_{h,i} = \emptyset$. As a result, V_k is eventually partitioned into the following list of subperiods: $x_{h,0}$, $x_{l,1}$, $x_{h,1}$, $x_{l,2}$, $x_{h,2}$, $x_{l,3}$, $x_{h,3}$, $\ldots$, which we shall refer to as the subperiods produced from bin b_k. To facilitate presentation, we shall also refer to b_k as the bin of these subperiods.

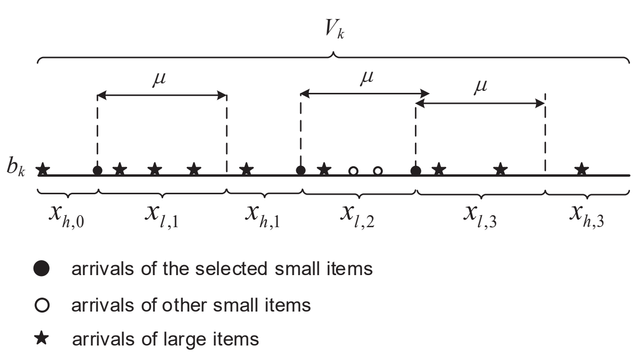

Figure 3. An example of item selection and period split

The above item selection and period split process implies the following properties:

Proposition 3: For each l-subperiod $x_{l,i}$, it holds that $l(x_{l,i}) \leq \mu$.

Proposition 4: At the left endpoint of each l-subperiod $x_{l,i}$, a new small item is placed in the bin of $x_{l,i}$.

Proposition 5: For any two consecutive l-subperiods $x_{l,i}$ and $x_{l,i+1}$, it holds that $l(x_{l,i}) + l(x_{l,i+1}) > \mu$.

Proof: Assume on the contrary that $l(x_{l,i})+l(x_{l,i+1}) \leq \mu$. This implies $l(x_{l,i}) < \mu$ and $l(x_{l,i+1}) < \mu$. It follows from the period split process that $x_{h,i} = \emptyset$ and $x_{h,i+1} = \emptyset$. Thus, we have $l(x_i) + l(x_{i+1}) = l(x_{l,i}) + l(x_{l,i+1}) \leq \mu$ and $l(x_i) < \mu$. Let r_i and r_{i+1} denote the selected small items arriving at the beginning of periods x_i and x_{i+1} respectively. If r_{i+1} is the last item selected, then the fact $l(x_i) + l(x_{i+1}) \leq \mu$ indicates that r_i is a small item arriving within a duration μ before the end of period V_k. According to termination condition (i), no item should be further selected after r_i, which contradict that r_{i+1} is a selected item. If r_{i+1} is not the last item selected, let r_{i+2} denote the next item selected after r_{i+1}. Since $l(x_i) < \mu$, r_{i+1} is a small item placed in bin b_k within a duration μ after r_i's arrival. The item selection process implies that r_{i+1} must be the last small item placed in bin b_k within a duration μ after r_i's arrival. Therefore, r_{i+2} must arrive beyond a duration μ after r_i's arrival. As a result, $l(x_i)+l(x_{i+1}) > \mu$, which again results in a contradiction. ■

Proposition 6: The bin level is at least $\frac{1}{\mu+3}$ high throughout the h-subperiods $x_{h,0}$, $x_{h,1}$, $x_{h,2}$, $x_{h,3}$, $\ldots$.

According to Proposition 4, a new small item is placed in bin b_k at the beginning of each l-subperiod produced from b_k. Then, by the definition of First Fit packing, the bins with indexes lower than k, if any, must have rather high bin levels at that time. Based on this observation, we attempt to find some periods in these bins to compensate for the potential low utilization of the l-subperiods.

First, for each l-subperiod $x_{l,i}$ produced from bin b_k, there must exist at least one open bin with an index lower than k at time $x_{l,i}^-$ (the beginning of $x_{l,i}$). Otherwise, b_k would be the open bin with the lowest index at time $x_{l,i}^-$. According to the previous definitions, $x_{l,i}^-$ would then belong to W_k, which contradicts that $x_{l,i}^-$ is in V_k. Among all the open bins with indexes lower than k at time $x_{l,i}^-$, we define the last opened bin (the bin with the highest index) as the *supplier bin* of the l-subperiod $x_{l,i}^-$. Figure 4 shows an example of supplier bins.

Next, we consolidate some l-subperiods produced from the same bin. Let $x_{l,1}$, $x_{l,2}$, $x_{l,3}$, $\ldots$ be the set of l-

subperiods produced from one bin. We define the *pair* relationship between two consecutive l-subperiods.

Definition 1: Two consecutive l-subperiods $x_{l,i}$ and $x_{l,i+1}$ are said to form a pair if they have the same supplier bin and $l(x_{l,i+1}) > \mu \cdot l(x_{l,i})$.

Proposition 7: If two consecutive l-subperiods $x_{l,i}$ and $x_{l,i+1}$ form a pair, then $x_{h,i} = \emptyset$.

Proof: By Proposition 3, we have $\mu \cdot l(x_{l,i}) < l(x_{l,i+1}) \le \mu$. It follows that $l(x_{l,i}) < 1 \le \mu$. This indicates $x_{h,i} = \emptyset$, since the h-subperiod $x_{h,i}$ is non-empty only if $l(x_{l,i}) = \mu$. ■

Note that it is possible for a sequence of three or more consecutive l-subperiods to form pairs in a concatenated manner. For example, suppose that $1 \le \mu < \frac{1+\sqrt{5}}{2}$. Let ϵ be a sufficiently small value such that $1 \le \mu < \mu + \epsilon < \frac{1+\sqrt{5}}{2}$. Then, we have $(\mu+\epsilon)^2 < \mu+\epsilon+1$ and thus $\frac{\mu}{\mu+\epsilon+1} < \frac{\mu}{(\mu+\epsilon)^2}$. Let z be a value satisfying $\frac{\mu}{\mu+\epsilon+1} < z < \frac{\mu}{(\mu+\epsilon)^2}$. Suppose the lengths of three consecutive l-subperiods are $l(x_{l,i}) = z$, $l(x_{l,i+1}) = (\mu+\epsilon)z$, and $l(x_{l,i+2}) = (\mu+\epsilon)^2 z$. Then, we have $l(x_{l,i}) < l(x_{l,i+1}) < l(x_{l,i+2}) < \mu$ (satisfying Proposition 3) and $l(x_{l,i+1}) + l(x_{l,i+2}) > l(x_{l,i}) + l(x_{l,i+1}) > \mu$ (satisfying Proposition 5). Since $l(x_{l,i+1}) > \mu \cdot l(x_{l,i})$ and $l(x_{l,i+2}) > \mu \cdot l(x_{l,i+1})$, the l-subperiod $x_{l,i+1}$ forms pairs with both its left and right neighboring l-subperiods.

If an l-subperiod does not form any pair with its neighboring l-subperiods, we call it a *single* l-subperiod. For the l-subperiods that form pairs, we combine each maximal sequence of consecutive l-subperiods into a *consolidated* l-subperiod.

Definition 2: A sequence of consecutive l-subperiods $\{x_{l,i}, x_{l,i+1}, \ldots, x_{l,j}\}$ $(j \ge i+1)$ is combined into a consolidated l-subperiod if and only if the following conditions are all satisfied:

(1) any two consecutive l-subperiods in the sequence form a pair;

(2) $x_{l,i}$ does not form a pair with $x_{l,i-1}$, or $i = 1$ ($x_{l,i}$ is the first l-subperiod of the bin);

(3) $x_{l,j}$ does not form a pair with $x_{l,j+1}$, or $x_{l,j}$ is the last l-subperiod of the bin.

Based on the above definitions, all the l-subperiods $x_{l,1}, x_{l,2}, x_{l,3}, \ldots$ produced from one bin are divided into single l-subperiods and consolidated l-subperiods. For each single l-subperiod $x_{l,i}$, we define the time interval $[x^-_{l,i} - \frac{l(x_{l,i})}{\mu+1}, x^-_{l,i} + \frac{l(x_{l,i})}{\mu+1})$ associated with $x_{l,i}$'s supplier bin as the *supplier period* of $x_{l,i}$ (see Figure 4). Note that the length of $x_{l,i}$'s supplier period is $\frac{2}{\mu+1} \cdot l(x_{l,i})$. The pair relationship (Definition 1) essentially gives the condition for the supplier periods of two consecutive l-subperiods to overlap if they were single l-subperiods. For each consolidated l-subperiod $\{x_{l,i}, x_{l,i+1}, \ldots, x_{l,j}\}$, by definition, all the l-subperiods in the sequence share a common supplier bin. We define the time interval $[x^-_{l,i+1} - \max\{\frac{l(x_{l,i})}{\mu+1} + l(x_{l,i}), \frac{l(x_{l,i})+l(x_{l,i+1})}{\mu+1}\}, x^-_{l,j} + \frac{l(x_{l,j})}{\mu+1})$ associated with their supplier bin as the *supplier period* of the consolidated l-subperiod (see Figure 4). As shall be shown later in Lemma 3, the supplier period of the consolidated l-subperiod includes the supplier periods of all the l-subperiods in it if they were single l-subperiods. We have the following lemma about the length of the supplier period.

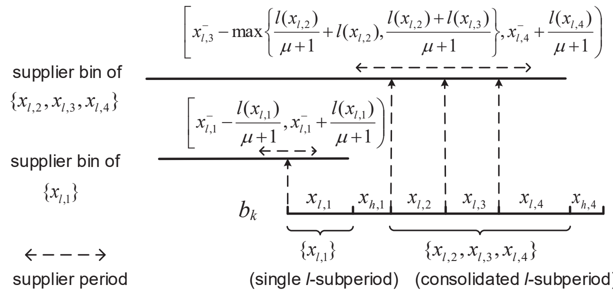

Figure 4. An example of supplier bins and periods

Lemma 1: The supplier period of a consolidated l-subperiod $\{x_{l,i}, x_{l,i+1}, \ldots, x_{l,j}\}$ has a length shorter than $\frac{2}{\mu+1} \cdot \sum_{i \le k \le j} l(x_{l,k})$.

Proof: Please refer to the extended version of this paper [21] for details. ■

VI. Intersection between Supplier Periods

To study the amortized bin level, we shall work towards calculating the total time-space demand for all the items in the l-subperiods and their supplier periods. To that end, we first check for possible intersections among all the supplier periods, where two supplier periods are defined to *intersect* if and only if they are associated with the *same* supplier bin and their time intervals overlap. In this section, we show that the supplier periods of all the single and consolidated l-subperiods do not intersect with each other.

A. *l-subperiods from the Same Bin*

We first examine the supplier periods of two successive single/consolidated l-subperiods produced from the same bin. Let $x_{l,1}, x_{l,2}, x_{l,3}, \ldots$ denote the set of l-subperiods produced from one bin.

If the two l-subperiods have different supplier bins, their supplier periods cannot intersect according to the definition. Thus, we only need to consider the situation where the two l-subperiods have the same supplier bin. Their relationship can be classified into the following two cases.

Case 1: An l-subperiod of any type followed by a single l-subperiod.

Suppose a single l-subperiod $x_{l,i}$ (or a consolidated l-subperiod $\{\ldots, x_{l,i-1}, x_{l,i}\}$) is followed by another single l-subperiod $x_{l,i+1}$. By definition, the supplier period of $x_{l,i}$ (or $\{\ldots, x_{l,i-1}, x_{l,i}\}$) has a right endpoint of $x^-_{l,i} + \frac{l(x_{l,i})}{\mu+1}$, and the supplier period of $x_{l,i+1}$ has a left endpoint of $x^-_{l,i+1} - \frac{l(x_{l,i+1})}{\mu+1}$. Since $x_{l,i}$ and $x_{l,i+1}$ do not form a pair, according to Definition 1, we have $l(x_{l,i+1}) \leq \mu \cdot l(x_{l,i})$. It follows that $l(x_{l,i}) + l(x_{l,i+1}) \leq (\mu+1) \cdot l(x_{l,i})$. So,

$$\frac{l(x_{l,i})}{\mu+1} + \frac{l(x_{l,i+1})}{\mu+1} \leq l(x_{l,i}) \\ \leq l(x_{l,i}) + l(x_{h,i}) = x^-_{l,i+1} - x^-_{l,i}, \qquad (2)$$

and hence,

$$x^-_{l,i} + \frac{l(x_{l,i})}{\mu+1} \leq x^-_{l,i+1} - \frac{l(x_{l,i+1})}{\mu+1}.$$

Therefore, the two supplier periods do not intersect.

Case 2: An l-subperiod of any type followed by a consolidated l-subperiod.

Suppose a single l-subperiod $x_{l,i}$ (or a consolidated l-subperiod $\{\ldots, x_{l,i-1}, x_{l,i}\}$) is followed by another consolidated l-subperiod $\{x_{l,i+1}, x_{l,i+2}, \ldots\}$. The supplier period of $x_{l,i}$ (or $\{\ldots, x_{l,i-1}, x_{l,i}\}$) has a right endpoint of $x^-_{l,i} + \frac{l(x_{l,i})}{\mu+1}$. The supplier period of $\{x_{l,i+1}, x_{l,i+2}, \ldots\}$ has a left endpoint of $x^-_{l,i+2} - \max\{\frac{l(x_{l,i+1})}{\mu+1} + l(x_{l,i+1}), \frac{l(x_{l,i+1})+l(x_{l,i+2})}{\mu+1}\}$. Since $x_{l,i}$ and $x_{l,i+1}$ do not form a pair, following (2) in the analysis of Case 1, we have

$$\frac{l(x_{l,i})}{\mu+1} + \frac{l(x_{l,i+1})}{\mu+1} + l(x_{l,i+1}) \leq l(x_{l,i}) + l(x_{l,i+1}). \qquad (3)$$

In addition, since $l(x_{l,i}) + l(x_{l,i+1}) > \mu \geq 1$ (Proposition 5), it follows that $\mu \cdot (l(x_{l,i}) + l(x_{l,i+1})) > \mu \geq l(x_{l,i+2})$. So, we have

$$(\mu+1) \cdot (l(x_{l,i}) + l(x_{l,i+1})) > l(x_{l,i}) + l(x_{l,i+1}) + l(x_{l,i+2}),$$

and hence,

$$l(x_{l,i}) + l(x_{l,i+1}) > \frac{l(x_{l,i})}{\mu+1} + \frac{l(x_{l,i+1}) + l(x_{l,i+2})}{\mu+1}. \qquad (4)$$

Combining (3) and (4), we obtain

$$\frac{l(x_{l,i})}{\mu+1} + \max\left\{\frac{l(x_{l,i+1})}{\mu+1} + l(x_{l,i+1}), \frac{l(x_{l,i+1}) + l(x_{l,i+2})}{\mu+1}\right\} \\ \leq l(x_{l,i}) + l(x_{l,i+1}) \\ \leq l(x_{l,i}) + l(x_{h,i}) + l(x_{l,i+1}) + l(x_{h,i+1}) \\ = x^-_{l,i+2} - x^-_{l,i}.$$

Therefore,

$$x^-_{l,i} + \frac{l(x_{l,i})}{\mu+1} \\ \leq x^-_{l,i+2} - \max\left\{\frac{l(x_{l,i+1})}{\mu+1} + l(x_{l,i+1}), \frac{l(x_{l,i+1}) + l(x_{l,i+2})}{\mu+1}\right\},$$

which implies that the two supplier periods do not intersect.

B. l-subperiods from Different Bins

Next, we examine the supplier periods of two single/consolidated l-subperiods produced from different bins. Let $x_{l,1}, x_{l,2}, x_{l,3}, \ldots$ denote the set of l-subperiods produced from one bin, and let $y_{l,1}, y_{l,2}, y_{l,3}, \ldots$ denote the set of l-subperiods produced from another bin. Again, we only consider the situation where the two l-subperiods have the same supplier bin. Their relationship can also be classified into two cases.

Case 3: An l-subperiod of any type followed by a single l-subperiod.

Suppose a single l-subperiod $x_{l,i}$ (or a consolidated l-subperiod $\{\ldots, x_{l,i-1}, x_{l,i}\}$) is followed by another single l-subperiod $y_{l,j}$, i.e., their left endpoints satisfy $x^-_{l,i} \leq y^-_{l,j}$. Assume $x_{l,i}$ is produced from bin b_g, and $y_{l,j}$ is produced from bin b_k. If $k < g$, then the supplier bin of $x_{l,i}$ cannot have an index lower than k, since bin b_k is opened before bin b_g, and b_k is not closed at least until time $y^-_{l,j} \geq x^-_{l,i}$. On the other hand, by definition, the supplier bin of $y_{l,j}$ must have an index lower than k. As a result, $x_{l,i}$ and $y_{l,j}$ cannot have the same supplier bin. Therefore, if $x_{l,i}$ and $y_{l,j}$ have the same supplier bin, we must have $g < k$ and the supplier bin has an index lower than g (see Figure 5). Then, in order for $y_{l,j}$ to have a supplier bin with index lower than g, bin b_g must be closed by time $y^-_{l,j}$, which implies that $y^-_{l,j} \geq U^+_g$ where U^+_g is the ending time of b_g's usage period U_g. Since a new item is placed in bin b_g at time $x^-_{l,i}$ (Proposition 4) and the minimum item duration is 1, bin b_g must remain open for a duration at least 1 after $x^-_{l,i}$. Thus, we have

$$y^-_{l,j} \geq U^+_g \geq x^-_{l,i} + 1. \qquad (5)$$

Note that the supplier period of $x_{l,i}$ (or $\{\ldots, x_{l,i-1}, x_{l,i}\}$) has a right endpoint of $x^-_{l,i} + \frac{l(x_{l,i})}{\mu+1}$, and the supplier period of $y_{l,j}$ has a left endpoint of $y^-_{l,j} - \frac{l(y_{l,j})}{\mu+1}$. Since $l(y_{l,j}) \leq \mu$, it follows that

$$l(y_{l,j}) \leq \mu \leq \mu \cdot (y^-_{l,j} - x^-_{l,i}).$$

It is also obvious that $x^-_{l,i} + l(x_{l,i}) = x^+_{l,i} \leq U^+_g$, which indicates

$$l(x_{l,i}) \leq U^+_g - x^-_{l,i} \leq y^-_{l,j} - x^-_{l,i}. \qquad (6)$$

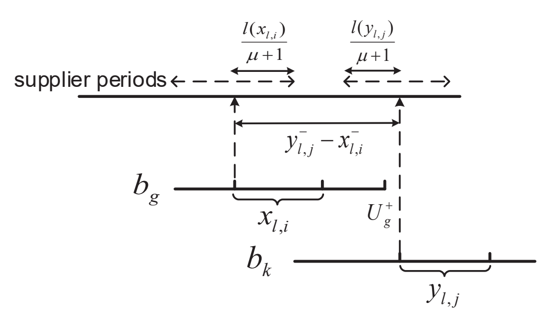

Figure 5. Case 3

Therefore,

$$\begin{aligned}&\frac{l(x_{l,i})}{\mu+1}+\frac{l(y_{l,j})}{\mu+1}\\&\le\frac{1}{\mu+1}\cdot(y^-_{l,j}-x^-_{l,i})+\frac{\mu}{\mu+1}\cdot(y^-_{l,j}-x^-_{l,i})\\&=y^-_{l,j}-x^-_{l,i}.\end{aligned}\tag{7}$$

Thus, we have

$$x^-_{l,i}+\frac{l(x_{l,i})}{\mu+1}\le y^-_{l,j}-\frac{l(y_{l,j})}{\mu+1},$$

so the two supplier periods do not intersect.

Case 4: An l-subperiod of any type followed by a consolidated l-subperiod.

Suppose a single l-subperiod $x_{l,i}$ (or a consolidated l-subperiod $\{\ldots,\,x_{l,i-1},\,x_{l,i}\}$) is followed by another consolidated l-subperiod $\{y_{l,j},\,y_{l,j+1},\,\ldots\}$ (see Figure 6). The supplier period of $x_{l,i}$ (or $\{\ldots,\,x_{l,i-1},\,x_{l,i}\}$) has a right endpoint of $x^-_{l,i}+\frac{l(x_{l,i})}{\mu+1}$. The supplier period of $\{y_{l,j},\,y_{l,j+1},\,\ldots\}$ has a left endpoint of $y^-_{l,j+1}-\max\{\frac{l(y_{l,j})}{\mu+1}+l(y_{l,j}),\,\frac{l(y_{l,j})+l(y_{l,j+1})}{\mu+1}\}$. Similar to Case 3, it can be proved that (7) holds, so

$$\begin{aligned}\frac{l(x_{l,i})}{\mu+1}+\frac{l(y_{l,j})}{\mu+1}&\le y^-_{l,j}-x^-_{l,i}\\&=y^-_{l,j+1}-l(y_{l,j})-l(y_{h,j})-x^-_{l,i}\\&\le y^-_{l,j+1}-l(y_{l,j})-x^-_{l,i}.\end{aligned}$$

Thus, we have

$$y^-_{l,j+1}-\Big(\frac{l(y_{l,j})}{\mu+1}+l(y_{l,j})\Big)\ge x^-_{l,i}+\frac{l(x_{l,i})}{\mu+1}.\tag{8}$$

Moreover, (5) and (6) in the analysis of Case 3 also hold, i.e., $y^-_{l,j}\ge x^-_{l,i}+1$ and $l(x_{l,i})\le y^-_{l,j}-x^-_{l,i}$. It follows from $y^-_{l,j}-x^-_{l,i}\ge 1$ that $\mu\cdot(y^-_{l,j}-x^-_{l,i})\ge\mu\ge l(y_{l,j+1})$. Thus,

$$\begin{aligned}&(\mu+1)\cdot(y^-_{l,j+1}-x^-_{l,i})\\&=(\mu+1)\cdot(y^-_{l,j}+l(y_{l,j})+l(y_{h,j})-x^-_{l,i})\\&=(\mu+1)\cdot(y^-_{l,j}+l(y_{l,j})-x^-_{l,i})\\&>l(y_{l,j})+\mu\cdot(y^-_{l,j}-x^-_{l,i})+(y^-_{l,j}-x^-_{l,i})\\&\ge l(y_{l,j})+l(y_{l,j+1})+l(x_{l,i}).\end{aligned}$$

Therefore,

$$y^-_{l,j+1}-\frac{l(y_{l,j})+l(y_{l,j+1})}{\mu+1}>x^-_{l,i}+\frac{l(x_{l,i})}{\mu+1}.\tag{9}$$

Combining (8) and (9), we have

$$\begin{aligned}&x^-_{l,i}+\frac{l(x_{l,i})}{\mu+1}\\&\le y^-_{l,j+1}-\max\Big\{\frac{l(y_{l,j})}{\mu+1}+l(y_{l,j}),\frac{l(y_{l,j})+l(y_{l,j+1})}{\mu+1}\Big\}.\end{aligned}$$

Thus, the two supplier periods do not intersect.

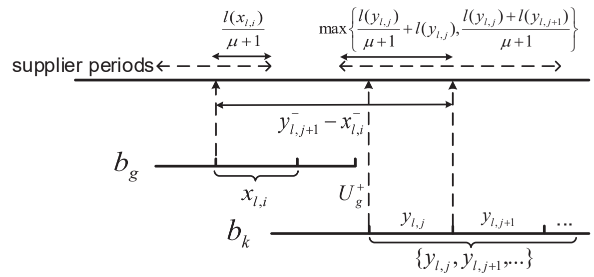

Figure 6. Case 4

Based on the above analysis, we have the following conclusion.

Lemma 2: The supplier periods of all the single and consolidated l-subperiods produced from all bins do not intersect with each other.

VII. TIME-SPACE DEMAND AND COMPETITIVE RATIO

A. A Single l-subperiod and Its Supplier Period

First, we calculate the time-space demands of the items in a single l-subperiod and its supplier period. Consider a single l-subperiod $x_{l,i}$. By Proposition 4, let p_i denote the selected small item placed in the bin of $x_{l,i}$ at the left endpoint $x^-_{l,i}$. Recall that each item resides in the system for a duration at least 1 (the minimum item duration). Since $\frac{l(x_{l,i})}{\mu+1}\le\frac{\mu}{\mu+1}<1$ and $\frac{l(x_{l,i})}{\mu+1}<l(x_{l,i})$, the time-space demand of p_i over $x_{l,i}$ is at least $s(p_i)\cdot\frac{l(x_{l,i})}{\mu+1}$.

Let R_i denote the set of items in the supplier bin of $x_{l,i}$ at the left endpoint $x^-_{l,i}$. Let $u(x_{l,i})$ denote the supplier period of $x_{l,i}$, i.e., $u(x_{l,i})=[x^-_{l,i}-\frac{l(x_{l,i})}{\mu+1},x^-_{l,i}+\frac{l(x_{l,i})}{\mu+1})$. Since $\frac{l(x_{l,i})}{\mu+1}<1$, the items in R_i must stay in the system for a duration at least $\frac{l(x_{l,i})}{\mu+1}$ over the supplier period $u(x_{l,i})$. Thus, the total time-space demand of all the items in R_i over $u(x_{l,i})$ is at least $s(R_i)\cdot\frac{l(x_{l,i})}{\mu+1}$. By the definition of First Fit packing, since $x_{l,i}$'s supplier bin cannot accommodate item p_i, the total size of the items in R_i together with p_i must exceed 1, i.e., $s(R_i)+s(p_i)>1$.

Let $d\big(u(x_{l,i})\big)$ denote the total time-space demand of the items in $x_{l,i}$'s supplier bin over the supplier period $u(x_{l,i})$. Let $d(x_{l,i})$ denote the total time-space demand of the items in $x_{l,i}$'s bin over $x_{l,i}$. Then, we have $d\big(u(x_{l,i})\big)\ge s(R_i)\cdot\frac{l(x_{l,i})}{\mu+1}$ and $d(x_{l,i})\ge s(p_i)\cdot\frac{l(x_{l,i})}{\mu+1}$. Therefore,

$$\begin{aligned}&d\big(u(x_{l,i})\big)+d(x_{l,i})\ge\big(s(R_i)+s(p_i)\big)\cdot\frac{l(x_{l,i})}{\mu+1}\\&>\frac{l(x_{l,i})}{\mu+1}=\frac{1}{\mu+3}\cdot\Big(\frac{2}{\mu+1}\cdot l(x_{l,i})+l(x_{l,i})\Big)\\&=\frac{1}{\mu+3}\cdot\Big(l(u(x_{l,i}))+l(x_{l,i})\Big).\end{aligned}\tag{10}$$

This means the amortized bin level over a single l-subperiod and its supplier period is at least $\frac{1}{\mu+3}$.

B. A Consolidated l-subperiod and Its Supplier Period

Next, we calculate the time-space demands of the items in a consolidated l-subperiod and its supplier period. Consider a consolidated l-subperiod $\{x_{l,i},\, x_{l,i+1},\, \ldots,\, x_{l,j}\}$. For each $i \le k \le j$, let p_k denote the selected small item arriving at time $x_{l,k}^-$ (Proposition 4). Similar to the analysis for a single l-subperiod, the time-space demand of p_k over $x_{l,k}$ is at least $s(p_k) \cdot \frac{l(x_{l,k})}{\mu+1}$. Let $d(x_{l,i} \cup x_{l,i+1} \cup \cdots \cup x_{l,j})$ denote the total time-space demand of the items in $x_{l,i}, x_{l,i+1}, \ldots, x_{l,j}$'s bin over $x_{l,i} \cup x_{l,i+1} \cup \cdots \cup x_{l,j}$. Then, we have

$$d(x_{l,i} \cup x_{l,i+1} \cup \cdots \cup x_{l,j}) \ge \sum_{i \le k \le j} s(p_k) \cdot \frac{l(x_{l,k})}{\mu+1}. \quad (11)$$

For each $i \le k \le j$, let R_k denote the set of items in the supplier bin of the consolidated l-subperiod at time $x_{l,k}^-$. For any k_1 and k_2 such that $k_2 - k_1 \ge 2$, since $x_{l,k_2}^- - x_{l,k_1}^- \ge l(x_{l,k_1}) + l(x_{l,k_1+1}) > \mu$ (Proposition 5) and μ is the maximum item duration, it follows that $R_{k_1} \cap R_{k_2} = \emptyset$. Therefore, $R_i \cup R_{i+1} \cup \cdots \cup R_j$ can be divided into the following disjoint subsets $R_i - R_{i+1}$, $R_i \cap R_{i+1}$, $R_{i+1} - R_i - R_{i+2}$, $R_{i+1} \cap R_{i+2}$, $R_{i+2} - R_{i+1} - R_{i+3}, \ldots, R_{j-1} - R_{j-2} - R_j$, $R_{j-1} \cap R_j$, $R_j - R_{j-1}$. For notational convenience, we define $R_{i-1} = \emptyset$ and $R_{j+1} = \emptyset$, so that the first and last subsets in the above list can be rewritten as $R_i - R_{i-1} - R_{i+1}$ and $R_j - R_{j-1} - R_{j+1}$. Similar to the case of a single l-subperiod, for the items in each $R_k - R_{k-1} - R_{k+1}$ ($i \le k \le j$), they must stay in the supplier bin for a duration at least $\frac{l(x_{l,k})}{\mu+1}$ in the time interval $[x_{l,k}^- - \frac{l(x_{l,k})}{\mu+1}, x_{l,k}^- + \frac{l(x_{l,k})}{\mu+1})$. The following lemma shows that this time interval is fully contained in the supplier period of $\{x_{l,i},\, x_{l,i+1},\, \ldots,\, x_{l,j}\}$.

Lemma 3: For each k where $i \le k \le j$, the time interval $[x_{l,k}^- - \frac{l(x_{l,k})}{\mu+1}, x_{l,k}^- + \frac{l(x_{l,k})}{\mu+1})$ is fully contained in the supplier period of $\{x_{l,i},\, x_{l,i+1},\, \ldots,\, x_{l,j}\}$.

Proof: Please refer to the extended version of this paper [21] for details. ■

It follows from Lemma 3 that the aggregate time-space demand of the items in $R_k - R_{k-1} - R_{k+1}$ over $\{x_{l,i},\, x_{l,i+1},\, \ldots,\, x_{l,j}\}$'s supplier period is at least

$$s(R_k - R_{k-1} - R_{k+1}) \cdot \frac{l(x_{l,k})}{\mu+1}.$$

For the items in each $R_k \cap R_{k+1}$ ($i \le k < j$), by definition, they definitely stay in the supplier bin from time $x_{l,k}^-$ to $x_{l,k+1}^-$. Note that $l(x_{h,k}) = 0$ (Proposition 7). Thus, the length of the time interval from $x_{l,k}^-$ to $x_{l,k+1}^-$ is $l(x_{l,k})$. Since $x_{l,k}$ and $x_{l,k+1}$ form a pair, we have $l(x_{l,k}) < \frac{l(x_{l,k+1})}{\mu}$ (Definition 1), and it follows that

$$\frac{l(x_{l,k}) + l(x_{l,k+1})}{\mu+1} < \frac{\frac{l(x_{l,k+1})}{\mu} + l(x_{l,k+1})}{\mu+1} = \frac{l(x_{l,k+1})}{\mu} \le 1.$$

Therefore, the items in $R_k \cap R_{k+1}$ must stay in the supplier bin for a duration at least $\frac{l(x_{l,k})+l(x_{l,k+1})}{\mu+1}$ in the time interval $[x_{l,k}^- - (\frac{l(x_{l,k})+l(x_{l,k+1})}{\mu+1} - l(x_{l,k})), x_{l,k+1}^- + (\frac{l(x_{l,k})+l(x_{l,k+1})}{\mu+1} - l(x_{l,k}))) = [x_{l,k+1}^- - \frac{l(x_{l,k})+l(x_{l,k+1})}{\mu+1}, x_{l,k}^- + \frac{l(x_{l,k})+l(x_{l,k+1})}{\mu+1})$. The following lemma shows that this time interval is fully contained in the supplier period of $\{x_{l,i},\, x_{l,i+1},\, \ldots,\, x_{l,j}\}$.

Lemma 4: For each k where $i \le k < j$, the time interval $[x_{l,k+1}^- - \frac{l(x_{l,k})+l(x_{l,k+1})}{\mu+1}, x_{l,k}^- + \frac{l(x_{l,k})+l(x_{l,k+1})}{\mu+1})$ is fully contained in the supplier period of $\{x_{l,i},\, x_{l,i+1},\, \ldots,\, x_{l,j}\}$.

Proof: Please refer to the extended version of this paper [21] for details. ■

It follows from Lemma 4 that the aggregate time-space demand of the items in $R_k \cap R_{k+1}$ over $\{x_{l,i},\, x_{l,i+1},\, \ldots,\, x_{l,j}\}$'s supplier period is at least

$$s(R_k \cap R_{k+1}) \cdot \frac{l(x_{l,k}) + l(x_{l,k+1})}{\mu+1}.$$

Let $u(\{x_{l,i}, x_{l,i+1}, \ldots, x_{l,j}\})$ denote the supplier period of $\{x_{l,i},\, x_{l,i+1},\, \ldots,\, x_{l,j}\}$. Let $d\big(u(\{x_{l,i}, x_{l,i+1}, \ldots, x_{l,j}\})\big)$ denote the total time-space demand of the items in $\{x_{l,i}, x_{l,i+1}, \ldots, x_{l,j}\}$'s supplier bin over the supplier period $u(\{x_{l,i}, x_{l,i+1}, \ldots, x_{l,j}\})$. Summarizing the above results, we have

$$\begin{aligned}
& d\big(u(x_{l,i}, x_{l,i+1}, \ldots, x_{l,j})\big) \\
& \ge \sum_{i \le k \le j} s(R_k - R_{j-1} - R_{k+1}) \cdot \frac{l(x_{l,k})}{\mu+1} \\
& \quad + \sum_{i \le k < j} s(R_k \cap R_{k+1}) \cdot \frac{l(x_{l,k}) + l(x_{l,k+1})}{\mu+1} \\
& = \sum_{i \le k \le j} \Big(s(R_{k-1} \cap R_k) + s(R_k - R_{k-1} - R_{k+1}) \\
& \quad + s(R_k \cap R_{k+1}) \Big) \cdot \frac{l(x_{l,k})}{\mu+1} \\
& = \sum_{i \le k \le j} s(R_k) \cdot \frac{l(x_{l,k})}{\mu+1}. \qquad (12)
\end{aligned}$$

By the definition of First Fit packing, for each $i \le k \le j$, we have $s(R_k) + s(p_k) > 1$. It follows from (11) and (12) that

$$\begin{aligned}
& d\big(u(\{x_{l,i}, x_{l,i+1}, \ldots, x_{l,j}\})\big) + d\big(x_{l,i} \cup x_{l,i+1} \cup \cdots \cup x_{l,j}\big) \\
& > \sum_{i \le k \le j} \frac{l(x_{l,k})}{\mu+1}.
\end{aligned}$$

By definition, $x_{l,i}, x_{l,i+1}, \ldots, x_{l,j}$ are all disjoint. Thus,

$$l(x_{l,i} \cup x_{l,i+1} \cup \cdots \cup x_{l,j}) = \sum_{i \le k \le j} l(x_{l,k}).$$

By Lemma 1,

$$l\big(u(\{x_{l,i}, x_{l,i+1}, \ldots, x_{l,j}\})\big) < \frac{2}{\mu+1} \cdot \sum_{i \le k \le j} l(x_{l,k}).$$

Therefore,

$$d\big(u(\{x_{l,i}, x_{l,i+1}, \ldots, x_{l,j}\})\big) + d(x_{l,i} \cup x_{l,i+1} \cup \cdots \cup x_{l,j})$$

$$> \frac{1}{\mu+3} \cdot \Big(\frac{2}{\mu+1} \cdot \sum_{i \le k \le j} l(x_{l,k}) + \sum_{i \le k \le j} l(x_{l,k}) \Big)$$

$$> \frac{1}{\mu+3} \cdot \Big(l\big(u(\{x_{l,i}, x_{l,i+1}, \dots, x_{l,j}\})\big) + l(x_{l,i} \cup x_{l,i+1} \cup \dots \cup x_{l,j}) \Big). \quad (13)$$

This shows the amortized bin level over a consolidated l-subperiod and its supplier period is also at least $\frac{1}{\mu+3}$.

C. All l-subperiods and Supplier Periods

Now, we put all the l-subperiods and their supplier periods together. Let $\mathcal{X}$ denote the set of all the single l-subperiods and consolidated l-subperiods produced from all bins in the packing process. Each $x \in \mathcal{X}$ is either a single or a consolidated l-subperiod. Let $u(x)$ denote the supplier period of x. In the previous subsections, we have proved that for each x, $d\big(u(x)\big) + d(x) > \frac{1}{\mu+3} \cdot \Big(l\big(u(x)\big) + l(x) \Big)$. We shall show in this subsection that

$$d\Big(\bigcup_{x \in \mathcal{X}} \big(u(x) \cup x\big) \Big) > \frac{1}{\mu+3} \cdot l\Big(\bigcup_{x \in \mathcal{X}} \big(u(x) \cup x\big) \Big). \quad (14)$$

Intuitively, the major barrier to establishing (14) is that the l-subperiods and their supplier periods may intersect. Recall from Lemma 2 that the supplier periods of all the single and consolidated l-subperiods do not intersect with each other. By definition, all the single and consolidated l-subperiods are also disjoint themselves. Thus, the only possibility of intersection is between a single/consolidated l-subperiod x_1 and the supplier period of another single/consolidated l-subperiod x_2. Fortunately, in deriving the amortized bin levels of (10) and (13), for the single/consolidated l-subperiod, we only count the time-space demand of the selected small items arriving at the left endpoints of the l-subperiods. Therefore, the time-space demands double-counted in the intersection parts are all due to these selected small items. Since small items have sizes less than $\frac{1}{\mu+3}$, the amortized bin level by aggregating all the l-subperiods and their supplier periods should remain at least $\frac{1}{\mu+3}$ when double counting is eliminated. A formal analysis leading to (14) is given in the extended version of this paper [21].

D. Competitive Ratio of First Fit Packing

Finally, we further add the h-subperiods produced from each bin into consideration. Let $\mathcal{Y}$ denote the set of all the h-subperiods produced from all bins. By definition, the h-subperiods do not intersect with the l-subperiods, but they may intersect with the supplier periods of l-subperiods. Since the h-subperiods have bin levels no less than $\frac{1}{\mu+3}$ (Proposition 6), we have

$$d\Big(\bigcup_{y \in \mathcal{Y}} y - \bigcup_{x \in \mathcal{X}} u(x) \Big) \ge \frac{1}{\mu+3} \cdot l\Big(\bigcup_{y \in \mathcal{Y}} y - \bigcup_{x \in \mathcal{X}} u(x) \Big). \quad (15)$$

Recall that all the l-subperiods and h-subperiods constitute the periods V_k. Thus, according to (1), (14) and (15), we have

$$\begin{aligned}
FF_{total}(\mathcal{R}) &\le \sum_{x \in \mathcal{X}} l(x) + \sum_{y \in \mathcal{Y}} l(y) + OPT_{total}(\mathcal{R}) \\
&= l\Big(\bigcup_{x \in \mathcal{X}} x \Big) + l\Big(\bigcup_{y \in \mathcal{Y}} y \Big) + OPT_{total}(\mathcal{R}) \\
&\le l\Big(\bigcup_{x \in \mathcal{X}} x \cup \bigcup_{x \in \mathcal{X}} u(x) \Big) + l\Big(\bigcup_{y \in \mathcal{Y}} y - \bigcup_{x \in \mathcal{X}} u(x) \Big) + OPT_{total}(\mathcal{R}) \\
&= l\Big(\bigcup_{x \in \mathcal{X}} (u(x) \cup x) \Big) + l\Big(\bigcup_{y \in \mathcal{Y}} y - \bigcup_{x \in \mathcal{X}} u(x) \Big) + OPT_{total}(\mathcal{R}) \\
&< (\mu+3) \cdot d\Big(\bigcup_{x \in \mathcal{X}} (u(x) \cup x) \Big) \\
&\quad + (\mu+3) \cdot d\Big(\bigcup_{y \in \mathcal{Y}} y - \bigcup_{x \in \mathcal{X}} u(x) \Big) + OPT_{total}(\mathcal{R}) \\
&= (\mu+3) \cdot d\Big(\bigcup_{x \in \mathcal{X}} x \cup \bigcup_{x \in \mathcal{X}} u(x) \cup \bigcup_{y \in \mathcal{Y}} y \Big) + OPT_{total}(\mathcal{R}).
\end{aligned}$$

Note that $d\Big(\bigcup_{x \in \mathcal{X}} x \cup \bigcup_{x \in \mathcal{X}} u(x) \cup \bigcup_{y \in \mathcal{Y}} y \Big)$ is bounded by the total time-space demand of all items over the entire packing period. It follows from Proposition 1 that $d\Big(\bigcup_{x \in \mathcal{X}} x \cup \bigcup_{x \in \mathcal{X}} u(x) \cup \bigcup_{y \in \mathcal{Y}} y \Big) \le OPT_{total}(\mathcal{R})$. Therefore, we have the following result:

$$\begin{aligned}
FF_{total}(\mathcal{R}) &\le (\mu+3) \cdot OPT_{total}(\mathcal{R}) + OPT_{total}(\mathcal{R}) \\
&= (\mu+4) \cdot OPT_{total}(\mathcal{R}).
\end{aligned}$$

Theorem 1: For the MinUsageTime DBP problem, the competitive ratio of First Fit packing has an upper bound of $\mu + 4$.

It has been proved that for MinUsageTime DBP, the competitive ratio of any online packing algorithm cannot be better than μ [12], [16]. Thus, the result of Theorem 1 indicates that First Fit packing is near optimal for MinUsage-Time DBP.

VIII. Comparison between First Fit and Next Fit

The Next Fit packing algorithm keeps exactly one bin available for receiving new items at any time. If an incoming item does not fit in the available bin, the available bin is marked unavailable and a new bin is opened (and marked available) to receive the new item. Unavailable bins are never marked available again and are closed when all the items in the bin depart.

Kamali *et al.* [12] has shown that the competitive ratio of Next Fit packing has an upper bound of $2\mu + 1$ for the MinTotal DBP problem. In this section, we show that the competitive ratio of Next Fit has a lower bound of 2μ by constructing an example. This implies that the multiplicative factor 2 of μ is inevitable in the competitive ratio of Next Fit. Therefore, First Fit is the only known packing algorithm so far whose competitive ratio has a multiplicative factor 1 for μ.

Let n be an integer no less than 3. At time 0, let $2n$ pairs of items arrive in sequence. The first item of each pair has a size $\frac{1}{2}$ and the second item has a size $\frac{1}{2n}$. At time 1, let all the items of size $\frac{1}{2}$ depart. At time μ, let all the items of size $\frac{1}{2n}$ depart.

When Next Fit packing is applied, each pair of items are placed in a separate bin because the first item of the pair (of size $\frac{1}{2}$) cannot fit in the previous open bin which has a level $\frac{1}{2} + \frac{1}{2n}$. Thus, $2n$ bins are opened from time 0 to μ. Therefore, the total bin usage time of Next Fit packing is $2n\mu$. On the other hand, in the optimal packing, every two items of size $\frac{1}{2}$ can be packed into one bin so that only n bins are enough to store all the items of size $\frac{1}{2}$ from time 0 to 1. All the items of size $\frac{1}{2n}$ can be packed into only one bin. Therefore, the total bin usage time of the optimal packing is $n + \mu$. The ratio between the bin usage times of Next Fit packing and optimal packing is $\frac{2n\mu}{n+\mu}$, which can be made arbitrarily close to 2μ as n goes towards infinity. Thus, the competitive ratio of Next Fit packing has a lower bound of 2μ.

IX. Concluding Remarks

The MinUsageTime DBP problem models online job dispatching to cloud servers. In this paper, we have developed new approaches to analyze the competitiveness of the commonly used First Fit packing algorithm for the MinUsageTime DBP problem, and established a new upper bound of $\mu + 4$ on its competitive ratio, which is the current best upper bound for the MinUsageTime DBP problem. Our result significantly reduces the gap between the upper and lower bounds for the MinUsageTime DBP problem to a constant independent of μ, and indicates that First Fit packing is near optimal for MinUsageTime DBP. One direction for future work is to extend the MinUsageTime DBP problem to the multi-dimensional version to model multiple types of resources (e.g., CPU and memory) for online cloud server allocation.

Acknowledgment

This work is supported by Singapore Ministry of Education Academic Research Fund Tier 2 under Grant MOE2013-T2-2-067, and by the National Research Foundation Singapore under its Interactive Digital Media (IDM) Strategic Research Programme.

References

[1] Amazon EC2 pricing. http://aws.amazon.com/ec2/pricing/.

[2] J. Balogh, J. Békési, and G. Galambos. New lower bounds for certain classes of bin packing algorithms. *Approximation and Online Algorithms (Lecture Notes in Computer Science, Volume 6534)*, pages 25–36, 2011.

[3] A. Borodin and R. El-Yaniv. *Online computation and competitive analysis*, volume 53. Cambridge University Press Cambridge, 1998.

[4] J. W.-T. Chan, T.-W. Lam, and P. W. Wong. Dynamic bin packing of unit fractions items. *Theoretical Computer Science*, 409(3):521–529, 2008.

[5] J. W.-T. Chan, P. W. Wong, and F. C. Yung. On dynamic bin packing: An improved lower bound and resource augmentation analysis. *Computing and Combinatorics*, pages 309–319, 2006.

[6] E. G. Coffman, Jr, M. R. Garey, and D. S. Johnson. Dynamic bin packing. *SIAM Journal on Computing*, 12(2):227–258, 1983.

[7] M. Flammini, G. Monaco, L. Moscardelli, H. Shachnai, M. Shalom, T. Tamir, and S. Zaks. Minimizing total busy time in parallel scheduling with application to optical networks. In *Proc. IEEE IPDPS'09*, pages 1–12, 2009.

[8] M. R. Garey and D. S. Johnson. Computers and intractability: A guide to the theory of NP-completeness, 1979.

[9] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, 1980.

[10] C.-Y. Huang, K.-T. Chen, D.-Y. Chen, H.-J. Hsu, and C.-H. Hsu. GamingAnywhere: the first open source cloud gaming system. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 10(1s):10, 2014.

[11] Z. Ivkovic and E. L. Lloyd. Fully dynamic algorithms for bin packing: Being (mostly) myopic helps. *SIAM Journal on Computing*, 28(2):574–611, 1998.

[12] S. Kamali and A. López-Ortiz. Efficient online strategies for renting servers in the cloud. *SOFSEM 2015: Theory and Practice of Computer Science*, 8939:277–288, 2015.

[13] E. L. Lawler, J. K. Lenstra, A. H. Rinnooy Kan, and D. B. Shmoys. Sequencing and scheduling: Algorithms and complexity. *Handbooks in Operations Research and Management Science*, 4:445–522, 1993.

[14] Y. Li, Y. Deng, R. Seet, X. Tang, and W. Cai. MASTER: Multi-platform application streaming toolkits for elastic resources. In *Proc. ACM MM'15*, pages 805–806, 2015.

[15] Y. Li, X. Tang, and W. Cai. Dynamic bin packing for on-demand cloud resource allocation. *IEEE Transactions on Parallel and Distributed Systems*, 27(1):157–170, 2016.

[16] Y. Li, X. Tang, and W. Cai. On dynamic bin packing for resource allocation in the cloud. In *Proc. ACM SPAA'14*, pages 2–11, 2014.

[17] G. B. Mertzios, M. Shalom, A. Voloshin, P. W. Wong, and S. Zaks. Optimizing busy time on parallel machines. In *Proc. IEEE IPDPS'12*, pages 238–248, 2012.

[18] S. S. Seiden. On the online bin packing problem. *Journal of the ACM*, 49(5):640–671, 2002.

[19] R. Shea, J. Liu, E.-H. Ngai, and Y. Cui. Cloud gaming: Architecture and performance. *IEEE Network*, 27(4):16–21, 2013.

[20] F. C. R. Spieksma. On the approximability of an interval scheduling problem. *Journal of Scheduling*, 2:215–227, 1999.

[21] X. Tang, Y. Li, R. Ren, and W. Cai. On First Fit Bin Packing for Online Cloud Server Allocation (Extended Version). Available at http://www.ntu.edu.sg/home/asxytang/.