Dynamic Bin Packing for On-Demand Cloud Resource Allocation

Yusen Li, Xueyan Tang, Wentong Cai

Abstract—Dynamic Bin Packing (DBP) is a variant of classical bin packing, which assumes that items may arrive and depart at arbitrary times. Existing works on DBP generally aim to minimize the maximum number of bins ever used in the packing. In this paper, we consider a new version of the DBP problem, namely, the MinTotal DBP problem which targets at minimizing the total cost of the bins used over time. It is motivated by the request dispatching problem arising from cloud gaming systems. We analyze the competitive ratios of the modified versions of the commonly used First Fit, Best Fit, and Any Fit packing (the family of packing algorithms that open a new bin only when no currently open bin can accommodate the item to be packed) algorithms for the MinTotal DBP problem. We show that the competitive ratio of Any Fit packing 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. For First Fit packing, if all the item sizes are smaller than $\frac{1}{\beta}$ of the bin capacity ($\beta > 1$ is a constant), the competitive ratio 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 packing has an upper bound of $2\mu + 7$. We also propose a Hybrid First Fit packing algorithm that can achieve a competitive ratio no larger than $\frac{5}{4}\mu + \frac{19}{4}$ when μ is not known and can achieve a competitive ratio no larger than $\mu + 5$ when μ is known.

Index Terms—Dynamic bin packing, online algorithms, competitive ratios, worst case bounds, theory.

1 INTRODUCTION

Bin packing is a classical combinatorial optimization problem which has been studied extensively [11], [13]. In the classical bin packing problem, given a set of items, the objective is to pack the items into a minimum number of bins such that the total size of the items in each bin does not exceed the bin capacity. Dynamic bin packing (DBP) is a generalization of the classical bin packing problem [12]. In the DBP problem, each item has a size, an arrival time and a departure time. The item stays in the system from its arrival to its departure. The objective is to pack the items into bins to minimize the maximum number of bins ever used over time. Dynamic bin packing has been used in [19] and [26] to model the resource consolidation problems in cloud computing.

In this paper, we consider a new version of the DBP problem, which is called the MinTotal DBP problem. In this problem, we assume that each bin used introduces a cost that is proportional to the duration of its usage, i.e., the period from its opening (when the first item is put into the bin) to its close (when all the items in the bin depart). The objective is to pack the items into bins to minimize the total cost of packing over time. We focus on the online version of the problem, where the items must be assigned to bins as they arrive without any knowledge of their departure times and future item arrivals. The arrival time and the size of an item are only known when the item arrives and the departure time is only known when the item departs. The items are not allowed to move from one bin to another once they have been assigned upon arrivals.

- *Yusen Li, Xueyan Tang and Wentong Cai are with the School of Computer Engineering, Nanyang Technological University, Nanyang Avenue 50, 639798, Singapore. E-mail: {S080007@e.ntu.edu.sg, ASXYTANG@ntu.edu.sg, ASWTCAI@ntu.edu.sg}*

The MinTotal DBP problem considered in this paper is primarily motivated by the request dispatching problem arising from cloud gaming systems. In a cloud gaming system, computer games run on powerful cloud servers, while players interact with the games via networked thin clients [17]. The cloud servers run the game instances, render the 3D graphics, encode them into 2D videos, and stream them to the clients. The clients then decode and display the video streams. This approach frees players from the overhead of setting up games, the hardware/software incompatibility problems, and the need for upgrading their computers regularly. Cloud gaming is a promising application of the rapidly expanding cloud computing infrastructure, and it has attracted a great deal of interests among entrepreneurs and researchers [23]. Several companies have offered cloud gaming services, such as GaiKai [2], OnLive [3], and StreamMyGame [4]. The cloud gaming market has been forecasted to reach 8 billion US dollars in 2017 [1].

Running each game instance demands a certain amount of GPU resources and the resource requirement can be different for running different games. In a cloud gaming system, when a playing request is received by the service provider, it needs to be dispatched to a game server that has enough GPU resources to run the game instance of this request. Several game instances can share the same game server as long as the server's GPU resources are not saturated. Each game instance keeps running in the system until the user stops playing the game. In general, the migration of game instances from one game server to another is not preferable due to large migration overheads and interruption to game play. In order to provide a good user experience, the gaming service provider needs to maintain a set of game servers with powerful GPUs for rendering the game instances. Constant workload fluctuation in cloud gaming makes the provisioning of game servers a challenging issue. The on-demand resource provisioning services in public clouds like Amazon EC2 provide an attractive solution. With these services, game service providers can rent virtual machines on demand to serve as game servers and pay for the resources according to their running hours. This frees game service providers from the complex process of planning, purchasing, and maintaining hardware. This approach has been adopted by many cloud gaming service providers like Gaikai and OnLive [27]. In the cloud gaming systems that use public clouds, one natural and important issue is how to dispatch the playing requests to game servers (i.e., virtual machines) so that the total cost of renting the game servers is minimized. The online MinTotal DBP problem we have defined exactly models this issue, where the game servers and playing requests correspond to the bins and items respectively.

For online bin packing, Any Fit packing algorithms have been extensively studied since they are simple and make decisions based on the current system state only. A packing algorithm is an Any Fit algorithm if it never packs an item in a new bin when there is a currently open bin with enough room for the item. First Fit and Best Fit are two commonly used Any Fit packing algorithms. First Fit puts a new item into the earliest open bin that can accommodate the item. Best Fit assigns a new item to the open bin with smallest residual capacity that can accommodate the item, ties broken in favor of the lowest-indexed open bin. In this paper, we analyze the performance of the modified versions of the First Fit, Best Fit and arbitrary Any Fit packing algorithms for the MinTotal DBP problem. We assume that an infinite number of bins are available for packing and all the bins have the same capacity and the same usage cost per unit time.

This paper extends a preliminary conference version [21] with significantly improved analysis, particularly on the competitive ratio of First Fit packing. The contributions of this paper are as follows. Let μ be the ratio of the maximum item duration to the minimum item duration. We prove that when $\mu > 1$, the competitive ratio of Any Fit packing cannot be better than $\mu+1$, and the competitive ratio of Best Fit packing is not bounded for any given μ. We show that for the case where 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, First Fit packing has a competitive ratio no larger than $2\mu+7$. In addition, we propose a Hybrid First Fit packing algorithm which classifies and assigns items according to their sizes. Hybrid First Fit packing can achieve a competitive ratio no larger than $\frac{5}{4}\mu + \frac{19}{4}$ when μ is not known and a competitive ratio no larger than $\mu + 5$ when μ is known.

The rest of this paper is structured as follows. The related work is summarized in Section 2. Section 3 introduces the system model, notations and packing algorithms. In Sections 4.1 to 4.3, the competitive ratios of First Fit, Best Fit, and arbitrary Any Fit packing algorithms for the MinTotal DBP problem are analyzed. Then, the Hybrid First Fit packing algorithm is proposed and its competitive ratio for the MinTotal DBP problem is analyzed in Section 4.4. Finally, conclusions are made and future work is discussed in Section 5.

2 Related Work

Cloud gaming systems have been implemented for both commercial use and research studies [4], [17], [3]. However, most of the existing work has focused on measuring the performance of cloud gaming systems [10], [25]. To the best of our knowledge, the resource management issues of cloud gaming have never been studied. The MinTotal DBP problem studied in this paper is related to a variety of research topics including the classical bin packing problem and its variations, as well as the interval scheduling problem.

The classical bin packing problem aims to put a set of items into the least number of bins. The problem and its variations have been studied extensively in both the offline and online versions [11], [15]. It is well known that the offline version of the classical bin packing problem is NP-hard [16]. For the online version, each item must be assigned to a bin without the knowledge of subsequent items. The items are not allowed to move from one bin to another. So far, the best upper bound on the competitive ratio for classical online bin packing is 1.58889, which is achieved by the HARMONIC++ algorithm proposed in [24]. The best known lower bound for any online packing algorithm is 1.54037 [5].

Dynamic bin packing is a variant of the classical bin packing problem [12]. It generalizes the problem by assuming that items may arrive and depart at arbitrary times. The objective is to minimize the maximum number of bins ever used in the packing. Coffman et al. [12] showed that the First Fit packing algorithm has a competitive ratio between 2.75 to 2.897 and no online algorithm can achieve a competitive ratio smaller than 2.5 against an optimal offline adversary that can repack everything at any time for free. Chan et al. [9] proved that the lower bound 2.5 on the competitive ratio also holds when the offline adversary does not repack. Ivkovic et al. [18] studied an even more general problem called the fully dynamic bin packing problem, where the online algorithm is also allowed to move already-packed items to different bins at any time for free. They proposed an online algorithm that achieves a competitive ratio of 1.25. Chan et al. [8] studied dynamic bin packing of unit fractions items (i.e., each item has a size $\frac{1}{w}$ for some integer $w \geq 1$). They showed that all Any Fit algorithms have competitive ratios 3.0 or better, that this bound is tight for Best Fit and Worst Fit, that First Fit has a competitive ratio between 2.45 and 2.4942, and that no online algorithm can have a competitive ratio better than 2.428. Classical dynamic bin packing does not consider bin usage costs and focuses simply on minimizing the maximum number of bins ever used. In contrast, the MinTotal DBP problem considered in this paper aims to minimize the total cost of the bins used in the packing.

The interval scheduling problem is also related to our problem [20]. The classical interval scheduling problem considers a set of jobs, each associated with a weight and an interval over which the job should be executed. Each machine can process only a single job at any time. Given a fixed number of machines, the objective is to schedule a feasible subset of jobs whose total weight is maximized [6]. Flammini et al. [14] have extended the classical model to a more general version, which is called interval scheduling with bounded parallelism. In this model, each machine can process $g > 1$ jobs simultaneously. If there is a job running on a machine, the machine is called busy. The objective is to assign the jobs to the machines such that the total busy time of the machines is minimized. It was proved that the problem to minimize the total busy time is NP-hard for $g \geq 2$ and a 4-competitive offline algorithm was proposed. Mertzios et al. [22] considered two special instances: clique instances (the intervals of all jobs share a common time point) and proper instances (the intervals of all jobs are not contained in one another), and provided constant factor approximation algorithms for these instances. However, the interval scheduling problem differs from our problem because the ending time of a job is known at the time of its assignment in interval scheduling, whereas in our MinTotal DBP model, the departure time of an item is not known at the time of its assignment. Furthermore, our MinTotal DBP problem does not assume that all the items have the same size, so the number of items that can be packed into a bin is not fixed.

3 Preliminaries

3.1 Notations and Definitions

Table 1 lists some notations used in this paper. Each item r to pack is associated with a 3-tuple $(a(r), d(r), s(r))$, where $a(r)$, $d(r)$ and $s(r)$ denote the arrival time, the departure time and the size of r respectively. Let $I(r)$ denote the time interval in which item r stays in the system (assume that $d(r) > a(r)$ is always true). We say that item r is *active* during this interval. The interval length of item r is represented by $len(I(r)) = d(r) - a(r)$. We extend the definition of len to unions of intervals by saying that $len(\cup_{r\in\mathcal{R}} I(r))$ is the length of time in which at least one item in $\mathcal{R}$ is active, and also refer to this as $span(\mathcal{R})$. Figure 1 shows an example of the span. Let $u(r) = s(r) \cdot len(I(r))$ denote the *resource demand* of item r. For any list of items $\mathcal{R}$, we define the total resource demand of $\mathcal{R}$ as $u(\mathcal{R}) = \sum_{r\in\mathcal{R}} u(r)$.

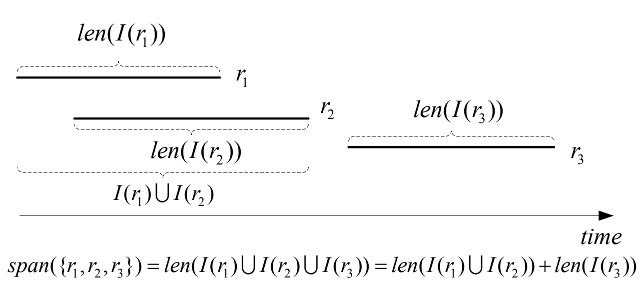


Fig. 1. Span of an item list

Without loss of generality, we assume in what follows that the bins all have unit capacity, and that the cost of using a bin for an interval of length L is simply L itself. At any time, the total size of all the items in an open bin is called the *level* of the bin. Let P be a packing of a list of items, and define $P(t)$ to be the number of bins containing items at time t in P. Then, the total cost of a packing P of a list of items $\mathcal{R}$ is

TABLE 1
Summary of Key Notations

Notation	Definition
$a(r)$	the arrival time of an item r
$d(r)$	the departure time of an item r
$s(r)$	the size of an item r
$I(r)$	the time interval in which an item r is active
$len(I(r))$	the length of $I(r)$
$u(r)$	the resource demand of an item r, $u(r) = s(r) \cdot len(I(r))$
$span(\mathcal{R})$	the span of an item list $\mathcal{R}$, $span(\mathcal{R}) = len(\bigcup_{r\in\mathcal{R}} I(r))$
$u(\mathcal{R})$	the total resource demand of an item list $\mathcal{R}$, $u(\mathcal{R}) = \sum_{r\in\mathcal{R}} u(r)$
$TotalCost(P_{A,\mathcal{R}})$	the total cost of a packing algorithm A applied to an item list $\mathcal{R}$
$OPT_{total}(\mathcal{R})$	the total cost of an optimal offline adversary that can repack everything at any time

$$TotalCost(P) = \int_{\min_{r\in\mathcal{R}} a(r)}^{\max_{r\in\mathcal{R}} d(r)} P(t)\mathrm{d}t,$$

where $\min_{r\in\mathcal{R}} a(r)$ is the time of the first item arrival and $\max_{r\in\mathcal{R}} d(r)$ is the time of the last item departure. Our goal is to find a packing P of $\mathcal{R}$ that minimizes the total cost.

3.2 Packing Algorithms

We consider versions of the standard bin packing algorithms which are modified as follows. When an empty bin first receives an item, we shall say that the bin is *opened*. When an open bin again becomes empty, we say it has been *closed*. Our modification is that, once a bin is closed, it is *permanently* closed, and we never place another item in it. Such a decision would be counterproductive in the Dynamic Bin Packing problem, where the goal is to minimize the number of bins used. It makes sense for our applications, however, where once a bin (server) becomes idle, we no longer pay for it, and it becomes indistinguishable from all the other idle servers. It also yields algorithms that are easier to reason about.

According to this set up, a Modified Any Fit (MAF) algorithm is any packing algorithm that never places an item in a new bin if it would fit in any of the currently open bins. We shall prove results that apply to all MAF algorithms, as well as results about the following two particular MAF algorithms.

- **Modified Best Fit**: The MAF algorithm which, if there is one or more open bins that can accommodate the current item, places the item in the one that accommodates the item with the least space left over, ties broken in favor of the lowest-indexed open bin.
- **Modified First Fit**: The MAF algorithm which, if there is one or more open bins that can accommodate the current item, places the item in the lowest-indexed open bin that has room for it.

For simplicity in what follows, we shall often refer to these algorithms simply as Best Fit (BF) and First Fit (FF), but readers should keep in mind that we are talking about the modified versions.

The performance of an online algorithm is normally measured by its competitive ratio, i.e., the worst-case ratio between the cost of the solution constructed by the algorithm and the cost of an optimal solution [7]. 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. Define

$$OPT_{total}(\mathcal{R}) = \int_{\min_{r\in\mathcal{R}} a(r)}^{\max_{r\in\mathcal{R}} d(r)} OPT(\mathcal{R}, t) \ \mathrm{d}t$$

It is easy to obtain the following two lower bounds on $OPT_{total}(\mathcal{R})$:

Bound (b.1): $OPT_{total}(\mathcal{R}) \geq u(\mathcal{R})$
Bound (b.2): $OPT_{total}(\mathcal{R}) \geq span(\mathcal{R})$

The first bound is derived by assuming that no bin capacity is wasted at any time. The second bound is derived from the fact that at least one bin must be use at any time when there is at least one active item.

For an algorithm A, let $P_{A,\mathcal{R}}$ denote the packing produced when A is applied to a list of items $\mathcal{R}$. Then, the total cost of algorithm A applied to $\mathcal{R}$ is $TotalCost(P_{A,\mathcal{R}})$. The competitive ratio for A is the maximum of $TotalCost(P_{A,\mathcal{R}})/OPT_{total}(\mathcal{R})$ over all lists of items $\mathcal{R}$. A standard approach to deriving bounds on the competitive ratio is to prove the following relation for all $\mathcal{R}$:

$$TotalCost(P_{A,\mathcal{R}}) \leq \alpha \cdot OPT_{total}(\mathcal{R})$$

where α is a constant [12]. Then, the competitive ratio for algorithm A is bounded above by α.

4 The Competitive Ratios

In this section, we analyze the competitive ratios of the packing algorithms for the MinTotal DBP problem. For any item list $\mathcal{R}$, let $\mu = \frac{\max_{r\in\mathcal{R}} len(I(r))}{\min_{r\in\mathcal{R}} len(I(r))}$ denote what we shall call the *max/min item interval length ratio*, where $\max_{r\in\mathcal{R}} len(I(r))$ is the maximum interval length among all the items $r \in \mathcal{R}$ and $\min_{r\in\mathcal{R}} len(I(r))$ is the minimum interval length among all the items $r \in \mathcal{R}$.

4.1 A Lower Bound for Any Fit Packing

First, we have the following result for Any Fit packing.

Theorem 4.1. *For any Modified Any Fit algorithm A, the MinTotal DBP competitive ratio of A is at least $\mu+1$, assuming $\mu > 1$.*

Proof: Let A be a Modified Any Fit algorithm. Let Δ be the minimum item interval length and $\mu\Delta$ be the maximum item interval length. Let k be an integer such that $\frac{1}{k} < (\mu-1)$. At time 0, let k^2 items of size $\frac{1}{k}$ arrive. A needs to open k bins to pack these items. Then, let one item depart from each open bin at time Δ. At time $\Delta + \frac{1}{2k}\Delta$, let k items of size $\frac{1}{k}$ arrive. It is easy to see that each of the k new items will be assigned to a different bin by A. After that, let all the "old" items (i.e., the items arrived at time 0) leave the system at time $\Delta + \frac{1}{k}\Delta$. At time $(\mu+1)\Delta + \frac{1}{2k}\Delta$, all the remaining items (i.e., the k items arrived at time $\Delta + \frac{1}{2k}\Delta$) leave the system.

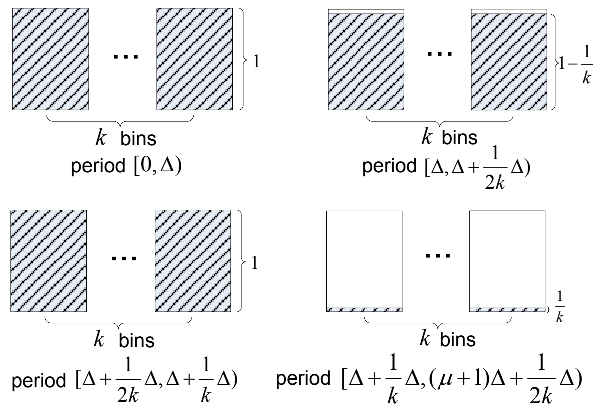


Fig. 2. Bin levels by a Modified Any Fit algorithm

As shown in Figure 2, there are always k open bins by algorithm A from time 0 to $(\mu+1)\Delta + \frac{1}{2k}\Delta$. Thus, the total cost of A is $TotalCost(P_{A,\mathcal{R}}) = k(\mu+1+\frac{1}{2k})\Delta$. From time $\Delta+\frac{1}{k}\Delta$ to $(\mu+1)\Delta+\frac{1}{2k}\Delta$, there are only k active items in the system which can in fact be packed into one bin. Therefore, $OPT_{total}(\mathcal{R}) \leq k(\Delta+\frac{1}{k}\Delta)+(\mu\Delta+\frac{1}{2k}\Delta-\frac{1}{k}\Delta) = \mu\Delta-\frac{1}{2k}\Delta$. It follows that

$$\begin{aligned}\frac{TotalCost(P_{A,\mathcal{R}})}{OPT_{total}(\mathcal{R})} &\geq \frac{k(\mu+1)\Delta}{k(\Delta+\frac{1}{k}\Delta)+(\mu\Delta-\frac{1}{2k}\Delta)} \\ &= \frac{k(\mu+1)}{k+1+\mu-\frac{1}{2k}} \geq \frac{k(\mu+1)}{k+1+\mu} \\ &= \frac{\mu+1}{1+\frac{1}{k}(\mu+1)} \qquad (1)\end{aligned}$$

It is obvious that expression (1) is monotonically increasing with k and $\lim_{k\to+\infty} \frac{\mu+1}{1+\frac{1}{k}(\mu+1)} = \mu+1$. So, given any small value $\varepsilon > 0$, we can always find an integer k such that $\frac{TotalCost(P_{A,\mathcal{R}})}{OPT_{total}(\mathcal{R})} > \mu+1-\varepsilon$. Therefore, the competitive ratio of algorithm A is at least $\mu+1$. □

4.2 Best Fit Packing

Next, we analyze the performance of Best Fit. Surprisingly, Best Fit is not competitive at all for the MinTotal DBP problem.

Theorem 4.2. *The MinTotal DBP competitive ratio of Best Fit is unbounded even when instances are restricted to those with $\mu \leq B$ for any constant $B > 1$.*

Proof: Let k be an integer. Let Δ be the minimum item interval length and $\mu\Delta$ be the maximum item interval length. Suppose that all the items have the same size ε, where ε is sufficiently small and $\frac{1}{\varepsilon}$ is an integer.

At time 0, let $\frac{k}{\varepsilon}$ items arrive. Best Fit needs to open k bins to pack all these items since their total size is k. Denote these k bins by $b_1, b_2, ..., b_k$. At time Δ, for each bin b_i, let some items depart to leave the level of b_i at $\frac{1}{k} - i \cdot \varepsilon$.

Then, let items arrive and depart according to the following iterative process. In the jth ($j \geq 1$) iteration, k groups of items arrive sequentially in the period $[j\mu\Delta - \delta, j\mu\Delta]$, where $\delta < (\mu - 1)\Delta$. The items in each group arrive at the same time and the ith group has $\frac{\frac{1}{k}-(j\cdot k+i)\cdot\varepsilon}{\varepsilon}$ items. By using Best Fit, the items in the first group (i.e., $i = 1$) will be assigned to b_1 since b_1 is the bin with the highest level in the system. After the items in the first group are packed, before the second group of items arrive, let all the "old" items in b_1 (the items arrived before time $j\mu\Delta - \delta$) depart. Then, the level of b_1 will become $\frac{1}{k} - (jk + 1)\cdot\varepsilon$. Next, the second group will be assigned to bin b_2, and so on so forth. In general, the items in the ith group will be packed in b_i since b_i is the bin with the highest level in the system when the ith group of items arrive. Before the $(i+1)$th group of items arrive, let the "old" items in b_i depart to leave the level of b_i at $\frac{1}{k} - (jk + i)\cdot\varepsilon$. Figure 3 shows the bin levels in the first few iterations.

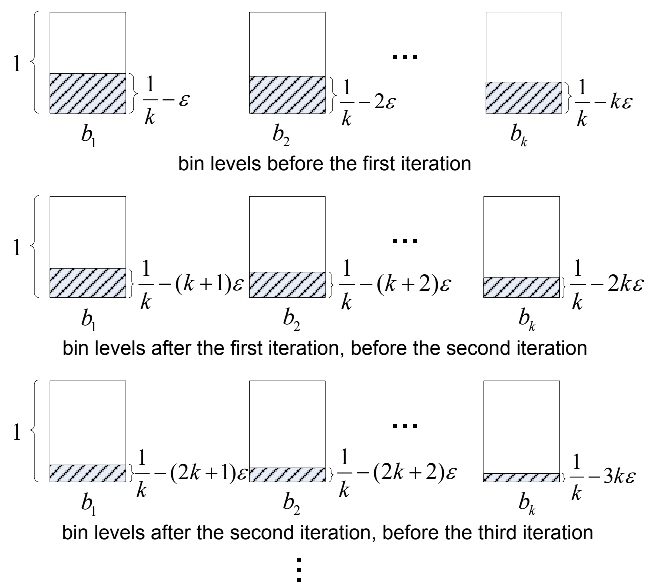


Fig. 3. Bin levels by Best Fit

Consider the time interval $[0, n\mu\Delta]$ in the above packing process, where n is an integer. Since there are always k open bins in the system, the total of Best Fit is $TotalCost(P_{BF,\mathcal{R}}) = kn\mu\Delta$. On the other hand, the total resource demand in the period $[0, \Delta]$ is $k\Delta$. After time Δ, except the periods $[j\mu\Delta - \delta, j\mu\Delta]$ (for each $1 \le j \le n$), all the active items in the system can be packed into one bin at any time. For the periods $[j\mu\Delta - \delta, j\mu\Delta]$, at most two bins are required to pack all the active items. Therefore,

$$\begin{aligned} OPT_{total}(\mathcal{R}) &\le k\Delta + (n\mu\Delta - \Delta - n\delta) + 2n\delta \\ &= k\Delta + (n\mu\Delta - \Delta) + n\delta \end{aligned}$$

It follows that

$$\frac{TotalCost(P_{BF,\mathcal{R}})}{OPT_{total}(\mathcal{R})} \ge \frac{kn\mu\Delta}{k\Delta + (n\mu\Delta - \Delta) + n\delta}$$

It can be proved that when $n \ge \frac{(k-1)\Delta}{\mu\Delta - \delta}$, we have

$$\frac{TotalCost(P_{BF,\mathcal{R}})}{OPT_{total}(\mathcal{R})} \ge \frac{k}{2} \tag{2}$$

Inequality (2) implies that the ratio $\frac{TotalCost(P_{BF,\mathcal{R}})}{OPT_{total}(\mathcal{R})}$ can be made arbitrarily large as k goes towards infinity. Therefore, the competitive ratio of Best Fit is unbounded even when μ is bounded by a constant $B > 1$. □

4.3 First Fit Packing

In this section, we study the competitive ratio of First Fit. We begin by examining the case of lists consisting of only "small" items, that is, items with sizes smaller than $\frac{1}{\beta}$ for some constant $\beta > 1$. Let Δ be the minimum item interval length and $\mu\Delta$ be the maximum item interval length.

4.3.1 Constructing Reference Periods

Suppose a total of m bins $b_1, b_2, ..., b_m$ are used by First Fit to pack $\mathcal{R}$. For each bin b_i, let I_i denote the usage period of b_i, that is, the interval from the time I_i^- when b_i was opened until the time I_i^+ when it was closed. For technical reasons, we shall view this interval as half-open, that is, as $[I_i^-, I_i^+)$. Denote the length $I_i^+ - I_i^-$ of I_i by $len(I_i)$. Note that, by definition of First Fit, we must have $I_1^- \le I_2^- \le \cdots \le I_m^-$.

Let $\mathcal{R}_i$ denote the set of items that are assigned to b_i by First Fit, then we have $I_i = \bigcup_{r\in\mathcal{R}_i} I(r)$. The total cost of First Fit is given by

$$TotalCost(P_{FF,\mathcal{R}}) = \sum_{i=1}^{m} len(I_i)$$

In what follows, we shall show how to construct lower bounds on $u(\mathcal{R}) \le OPT_{total}(\mathcal{R})$ in terms of $\sum_{i=1}^{m} len(I_i)$, which will thus yield bounds on the competitive ratio of First Fit.

For each bin b_i, let E_i be the latest closing time of all the bins that are opened before b_i, i.e., $E_i = \max\{I_j^+ | 1 \le j < i\}$, with $E_1 = I_1^-$. We divide period I_i into two parts, I_i^L and I_i^R. If $E_i \le I_i^-$, $I_i^L = \emptyset$. Otherwise, $I_i^L = [I_i^-, \min\{I_i^+, E_i\})$. In both cases, $I_i^R = I_i - I_i^L$. Note that if I_i^R is non-empty, it equals $[E_i, I_i^+)$. Figure 4 shows an example of these definitions. According to the definitions, we have $len(I_i) = len(I_i^L) + len(I_i^R)$ and it follows that

$$TotalCost(P_{FF,\mathcal{R}}) = \sum_{i=1}^{m} \left(len(I_i^L) + len(I_i^R)\right) \tag{3}$$

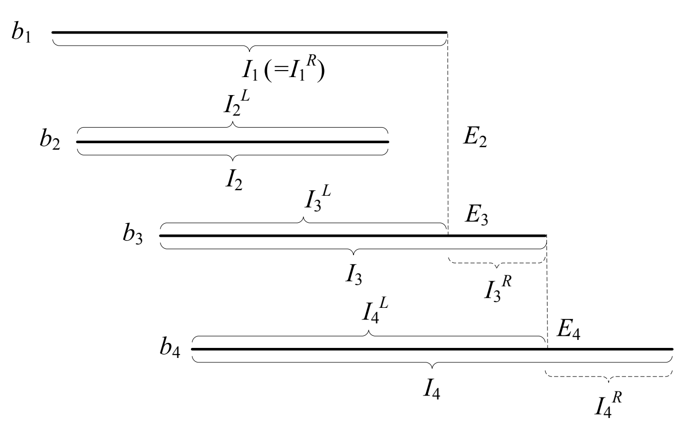


Fig. 4. An example of usage periods

Obviously, for any two different bins b_i and b_j, $I_i^R \cap I_j^R = \emptyset$. It is also easy to see that

$$span(\mathcal{R}) = \sum_{i=1}^{m} len(I_i^R) \tag{4}$$

According to (3) and (4), we have

$$TotalCost(P_{FF,\mathcal{R}}) = \sum_{i=1}^{m} len(I_i^L) + span(\mathcal{R}) \tag{5}$$

For each period I_i^L, if its length $len(I_i^L) > (\mu+2)\Delta$, we split I_i^L into $\lceil \frac{len(I_i^L)}{(\mu+2)\Delta} \rceil$ subperiods by inserting splitter points that are multiples of $(\mu+2)\Delta$ before the end of I_i^L, i.e., at times $\min\{I_i^+, E_i\} - k \cdot (\mu+2)\Delta$, for $k = 1, 2, \ldots, \lceil \frac{len(I_i^L)}{(\mu+2)\Delta} \rceil - 1$. After splitting, if the length of the first subperiod is shorter than 2Δ, we merge the first two subperiods into one. Then, we label all the subperiods in the temporal order by $I_{i,1}$, $I_{i,2}$, $I_{i,3}$, ..., i.e., their left endpoints satisfy $I_{i,1}^- < I_{i,2}^- < I_{i,3}^- < \cdots$. As before, the intervals are half-open, i.e., $[I_{i,j}^-, I_{i,j}^+)$. Figure 5 shows an example of the period split.

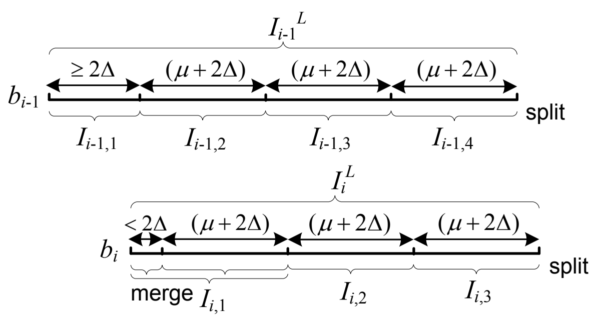


Fig. 5. An example of period split

Note that if the length of I_i^L does not exceed $(\mu+2)\Delta$, I_i^L is not split. In this case, we define $I_{i,1} = I_i^L$.

The above splitting and merging process implies the following features:

Feature (f.1): $len(I_{i,j}) < (\mu+4)\Delta$ for any i and j.

Feature (f.2): $len(I_{i,j}) = (\mu+2)\Delta$ for any $j \geq 2$ and any i.

Feature (f.3): For any i, if subperiod $I_{i,2}$ exists, then $len(I_{i,1}) \geq 2\Delta$.

For any set of subperiods $\mathbf{I}$, we denote the total length of the subperiods in $\mathbf{I}$ by $len(\mathbf{I}) = \sum_{I_{i,j} \in \mathbf{I}} len(I_{i,j})$. Let $\mathbf{I}_{\mathrm{L}}$ be the set of all the subperiods produced by the above splitting and merging process from all I_i^L's. It is obvious that

$$len(\mathbf{I}_{\mathrm{L}}) = \sum_{I_{i,j} \in \mathbf{I}_{\mathrm{L}}} len(I_{i,j}) = \sum_{i=1}^{m} len(I_i^L) \tag{6}$$

For each period $I_{i,j}$, it can be shown that at least one new item must be packed into bin b_i during $I_{i,j} = [I_{i,j}^-, I_{i,j}^+)$. In fact, if $len(I_{i,j}) \geq (\mu+2)\Delta$, there must be at least one new item packed into b_i during $[I_{i,j}^-, I_{i,j}^- + \mu\Delta)$. This is because all the items packed into b_i before $I_{i,j}^-$ would have departed by time $I_{i,j}^- + \mu\Delta$ since $\mu\Delta$ is the maximum item interval length. Thus, if no new item is packed into b_i during $[I_{i,j}^-, I_{i,j}^- + \mu\Delta)$, b_i would become empty and be closed by time $I_{i,j}^- + \mu\Delta$. On the other hand, if $len(I_{i,j}) < (\mu+2)\Delta$, according to Feature (f.2), $I_{i,j}$ must be the first subperiod in I_i^L, i.e., $j = 1$. Since b_i is opened at time $I_i^- = I_{i,1}^-$, at least one new item is packed into b_i at time $I_{i,1}^-$.

Let $t_{i,j}^\dagger$ denote the time when the first new item is packed into b_i during period $I_{i,j}$. We refer to $t_{i,j}^\dagger$ as the *reference point* of $I_{i,j}$. The above analysis implies that:

Feature (f.4): For each period $I_{i,1}$, it holds that $t_{i,1}^\dagger = I_{i,1}^-$.

Feature (f.5): For each period $I_{i,j}$, it holds that $I_{i,j}^- \leq t_{i,j}^\dagger \leq I_{i,j}^- + \mu\Delta$.

Lemma 4.3. *For each reference point $t_{i,j}^\dagger$ where $i > 1$, there must exist at least one bin b_h satisfying $h < i$ and $t_{i,j}^\dagger < I_h^+$.*

Proof: Assume on the contrary that all bins b_h with $h < i$ have $I_h^+ \leq t_{i,j}^\dagger$. Then, by definition, $E_i \leq t_{i,j}^\dagger$, and hence $t_{i,j}^\dagger \in I_i^R$, contradicting our assumption that $t_{i,j}^\dagger \in I_{i,j} \subseteq I_i^L$. □

Among all the bins b_h satisfying $h < i$ and $t_{i,j}^\dagger < I_h^+$, we define the last opened bin (the bin with the highest index) as the *reference bin* of $I_{i,j}$, and denote it by $b_{i,j}^\dagger$. We define the time interval $[t_{i,j}^\dagger - \Delta, t_{i,j}^\dagger + \Delta)$ associated with bin $b_{i,j}^\dagger$ as the *reference period* of $I_{i,j}$, and denote it by $p_{i,j}^\dagger$. Figure 6 shows an example of reference bins and reference periods.

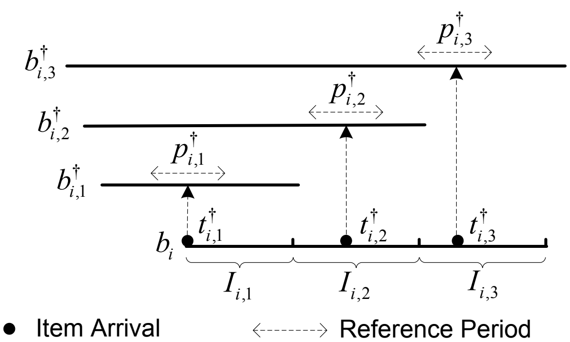


Fig. 6. An example of reference bins and periods

Since there is a new item packed into b_i at time $t_{i,j}^\dagger$ and the item size is smaller than $\frac{1}{\beta}$, the reference bin $b_{i,j}^\dagger$ must have a level higher than $1 - \frac{1}{\beta}$ at time $t_{i,j}^\dagger$ according to First Fit. That is, the total size of the items in $b_{i,j}^\dagger$ at time $t_{i,j}^\dagger$ is larger than $1 - \frac{1}{\beta}$. Recall that each of these items resides in the system for at least Δ time (the minimum item interval length). Thus, each of them must stay in bin $b_{i,j}^\dagger$ for at least Δ time during the reference period $p_{i,j}^\dagger = [t_{i,j}^\dagger - \Delta, t_{i,j}^\dagger + \Delta)$. Denote by $u(p_{i,j}^\dagger)$ the total resource demand of the items in bin $b_{i,j}^\dagger$ over period $p_{i,j}^\dagger$. It follows that

$$u(p_{i,j}^\dagger) \geq \left(1 - \frac{1}{\beta}\right) \cdot \Delta \tag{7}$$

In the following, we work towards calculating the total resource demand of all the reference periods. If two reference periods are associated with different bins, their total resource demand is simply the sum of their respective resource demands. However, if two reference periods are associated with the same bin, their total resource demand may be smaller than the sum of their respective demands due to possible temporal overlap between the reference periods. Two reference periods intersect if and only if they are associated with the same bin and their time intervals overlap. Note that the reference periods for two subperiods I_{i_1,j_1} and I_{i_2,j_2} intersect if and only if $b^\dagger_{i_1,j_1} = b^\dagger_{i_2,j_2}$ and $|t^\dagger_{i_1,j_1} - t^\dagger_{i_2,j_2}| < 2\Delta$.

The following two lemmas analyze the intersection between the reference periods.

Lemma 4.4. *Suppose $i \le j$. If subperiods $I_{i,g}$ and $I_{j,h}$ are distinct but have intersecting reference periods, then*
(a) $i < j$,
(b) $t^\dagger_{j,h} \ge I_i^+$,
(c) $g = 1$,
(d) $h = 1$, and
(e) $len(I_{i,1}) < 2\Delta$.

Proof: (a) If $i = j$ but the subperiods are distinct, we must have $g \ne h$. Without loss of generality, assume $g < h$. Thus, I_i has at least two subperiods, $len(I_{i,1}) \ge 2\Delta$ (Feature (f.3)), and $len(I_{i,h}) = (\mu + 2)\Delta$ (Feature (f.2)). If $g = 1$, then $t^\dagger_{i,g} = I_i^-$, and so $t^\dagger_{i,h} - t^\dagger_{i,g} \ge len(I_{i,1}) \ge 2\Delta$, and the corresponding reference periods cannot intersect. If $g > 1$, $t^\dagger_{i,g} \le I^-_{i,g} + \mu\Delta \le I^+_{i,g} - 2\Delta$, so, once again, $t^\dagger_{i,h} - t^\dagger_{i,g} \ge 2\Delta$, and the corresponding reference periods cannot intersect. Thus, we must have $i < j$.

(b) Since $j > i$, we must have $I_j^- \ge I_i^-$. If $t^\dagger_{j,h} < I_i^+$, we would thus have $t^\dagger_{j,h} \in I_i$ and so $b^\dagger_{j,h} \ge i$. But $b^\dagger_{i,g} < i$ by definition. So, the two reference periods could not intersect. Therefore, we must have $t^\dagger_{j,h} \ge I_i^+$.

(c) Now suppose $g > 1$. Then, $len(I_{i,g}) = (\mu + 2)\Delta$ (Feature (f.2)) and $t^\dagger_{i,g} \le I^-_{i,g} + \mu\Delta = I^+_{i,g} - 2\Delta \le I^+_i - 2\Delta \le t^\dagger_{j,h} - 2\Delta$, the last inequality following from (b). Once again the corresponding reference periods cannot intersect. So, we must have $g = 1$.

(d) Now suppose $g = 1$ and $h > 1$. Recall that $t^\dagger_{i,1} = I_i^-$ and that, since $i < j$, $I_i^- \le I_j^-$. Since $h > 1$, we also know that $len(I_{j,1}) \ge 2\Delta$ (Feature (f.3)). Therefore, $t^\dagger_{j,h} \ge I^-_{j,h} \ge I^+_{j,1} \ge I^-_{j,1} + len(I_{j,1}) \ge I_i^- + 2\Delta = t^\dagger_{i,1} + 2\Delta$, and once again the corresponding reference periods cannot intersect. So, we must have $h = 1$.

(e) By (b), we have $t^\dagger_{j,1} \ge I_i^+$, which, if $len(I_{i,1}) \ge 2\Delta$, is at least $t^\dagger_{i,1} + 2\Delta$, and once again the corresponding reference periods would not intersect. Thus, we must have $len(I_{i,1}) < 2\Delta$. □

Lemma 4.5. *For no i, $1 \le i \le m$, is there any point $t \in [I_i^-, I_i^+)$ that is contained in more than two reference periods.*

Proof: Suppose there are such an i and t. Then, there are indices $j > h > g > i$ and reference points $t^\dagger_{j,1}$, $t^\dagger_{h,1}$, and $t^\dagger_{g,1}$, for which bin b_i is the reference bin and such that t is in the reference period for each. (The second indices of all the reference points must be 1 by Lemma 4.4(c) and (d), since each is involved in an intersection).

By Lemma 4.4(b) and the fact that all items are of interval length at least Δ, we must have $t^\dagger_{h,1} \ge I_g^+ \ge I_g^- + \Delta$. Similarly, we must have $t^\dagger_{j,1} \ge I_h^- + \Delta$. But recall that by definition we have $I_h^- = t^\dagger_{h,1}$. Thus, we have $t^\dagger_{j,1} \ge I_g^- + 2\Delta$, and so the reference periods for $t^\dagger_{j,1}$ and $t^\dagger_{g,1}$ cannot intersect, and so t cannot belong to both, a contradiction. □

By Feature (f.1), we have $len(I_{i,j}) < (\mu + 4)\Delta$ for all subperiods. Thus, by (5) and Bound (b.2), there are at least

$$\begin{aligned}\frac{\sum_{i=1}^m len(I_i^L)}{(\mu+4)\Delta} &= \frac{TotalCost(P_{FF,\mathcal{R}}) - span(\mathcal{R})}{(\mu+4)\Delta} \\ &\ge \frac{TotalCost(P_{FF,\mathcal{R}}) - OPT_{total}(\mathcal{R})}{(\mu+4)\Delta}\end{aligned}$$

such periods. Now, by (7), we know that the reference period for each subperiod contains a total resource demand of at least $(1 - \frac{1}{\beta})\Delta$. By Lemma 4.5, no point is in more than two of these reference periods, so we have

$$\begin{aligned}&OPT_{total}(\mathcal{R}) = \sum_{r\in\mathcal{R}} u(r) \\ &\ge \frac{1}{2}\cdot\left(1 - \frac{1}{\beta}\right)\Delta\cdot\frac{TotalCost(P_{FF,\mathcal{R}}) - OPT_{total}(\mathcal{R})}{(\mu+4)\Delta} \\ &\ge \frac{1-\frac{1}{\beta}}{2\mu+8}\cdot(TotalCost(P_{FF,\mathcal{R}}) - OPT_{total}(\mathcal{R}))\end{aligned}$$

and hence

$$\begin{aligned}&OPT_{total}(\mathcal{R}) \\ &\ge \left(\frac{1-\frac{1}{\beta}}{2\mu+8}\right)\cdot\left(\frac{2\mu+8}{2\mu+8+1-\frac{1}{\beta}}\right)\cdot TotalCost(P_{FF,\mathcal{R}}) \\ &\ge \left(\frac{1-\frac{1}{\beta}}{2\mu+9}\right)\cdot TotalCost(P_{FF,\mathcal{R}})\end{aligned}$$

This implies an upper bound $\left(\frac{\beta}{\beta-1}\right)\cdot(2\mu+9)$ on the competitive ratio for our restricted item lists (with items of size smaller than $\frac{1}{\beta}$). The remainder of the section will show how to improve it by finding as-yet-uncounted quantities of resource demand to increase the lower bound on $OPT_{total}(\mathcal{R})$, and by providing a way of accounting for the items of size exceeding $\frac{1}{\beta}$ so that the result can be extended to arbitrary instances.

Let $\mathbf{I}_{\mathrm{L},1}$ be the set of all subperiods of the form $I_{i,1}$ and $\mathbf{I}_{\mathrm{L},2}$ be the set of all subperiods of the form $I_{i,j}$ where $j \ge 2$. Since all the periods in $\mathbf{I}_{\mathrm{L},2}$ have the same length (Feature (f.2)), we call them the *regular periods*.

Lemma 4.4 implies that only the periods in $\mathbf{I}_{\mathrm{L},1}$ may have intersecting reference periods. If the reference periods of $I_{i,1}$ and $I_{h,1}$ intersect and $i < h$ (i.e., $I^-_{i,1} \le I^-_{h,1}$), we call $I_{i,1}$ the *front-intersect period* of $I_{h,1}$ and call $I_{h,1}$ the *back-intersect period* of $I_{i,1}$. For any period $I_{i,1}$, if $I_{i,1}$ has two front-intersect or two back-intersect periods, it implies a common point in three reference periods, which contradicts Lemma 4.5. Thus, we have the following corollary.

Lemma 4.6. *For each period $I_{i,1}$, there is at most one front-intersect period and at most one back-intersect period of $I_{i,1}$.* □

Next, we construct pairs for the periods in $\mathbf{I}_{\mathrm{L},1}$ according to the following rule. Consider each period $I_{i,1} \in \mathbf{I}_{\mathrm{L},1}$ in the ascending order of i. If $I_{i,1}$ has not been added into any pair and $I_{i,1}$ has a "back-intersect" period (denoted by $I_{i',1}$), we construct a pair $(I_{i,1}, I_{i',1})$. We name the pair as a *joint period* and denote it by $J_{i,i'}$ (where $i < i'$). We define the reference period of the joint period $J_{i,i'}$ as the reference period of $I_{i,1}$, i.e., $p^{\dagger}_{i,1}$. Note that a period in $\mathbf{I}_{\mathrm{L},1}$ that has no "back-intersect" period might not be added into any pair. We name such period as a *single period*. An example of the pairing process is shown in Figure 7.

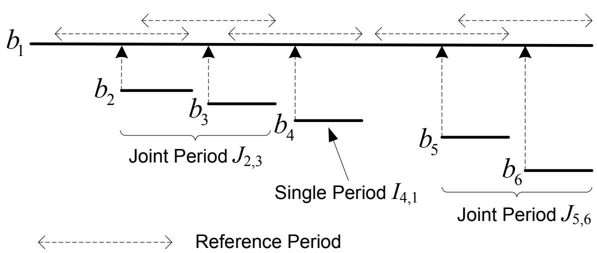


Fig. 7. An example of pairing

Lemma 4.7. *The reference periods of all the joint periods and single periods do not intersect with one another.*

Proof: Suppose we have two single/joint periods X_1 and X_2, whose reference periods intersect, with the latter having the rightmost reference point. For $j \in \{1, 2\}$, let $I_{i_j,1} = X_j$ if the latter is a single period, and be the first half of X_j if it is a joint period. The reference points for X_1 and X_2 are then $I^-_{i_1,1} \le I^-_{i_2,1}$. If X_1 is a single period, the fact that the reference periods for X_1 and X_2 intersect would thus imply that $I_{i_2,1}$ is the back-intersect period of $I_{i_1,1}$. But then we would have made a joint period out of $I_{i_1,1}$ and $I_{i_2,1}$, a contradiction of the fact that it is a single period. On the other hand, if X_1 is a joint period $J_{i_1,h}$, then the reference periods for $I_{i_1,1}$ and $I_{h,1}$ must both contain the rightmost point x in the reference period for $I_{i_1,1}$. But so would the reference period for $I^-_{i_2,1}$, by our hypothesis that the reference periods for X_1 and X_2 intersect, and that $I^-_{i_1,1} \le I^-_{i_2,1}$. Thus, x would be contained in three reference periods, violating Lemma 4.5. □

Let $\mathbf{J}$ denote the set of all the joint periods and $\mathbf{S}$ denote the set of all the single periods. Now, $\mathbf{I}_{\mathrm{L}}$ has been divided into three subsets:

$$\mathbf{I}_{\mathrm{L}} = \mathbf{I}_{\mathrm{L},2} \cup \mathbf{J} \cup \mathbf{S} \tag{8}$$

Let set $\mathbf{P}(\mathbf{I}_{\mathrm{L},2})$ denote the reference periods of all the regular periods in $\mathbf{I}_{\mathrm{L},2}$. Let set $\mathbf{P}(\mathbf{J})$ denote the reference periods of all the joint periods in $\mathbf{J}$, and let set $\mathbf{P}(\mathbf{S})$ denote the reference periods of all the single periods in $\mathbf{S}$. Lemma 4.7 together with Lemma 4.4 imply that the reference periods of all the joint periods, single periods and regular periods do not intersect.

Lemma 4.8. *All the reference periods in $\mathbf{P}(\mathbf{I}_{\mathrm{L},2}) \cup \mathbf{P}(\mathbf{J}) \cup \mathbf{P}(\mathbf{S})$ do not intersect with one another.* □

It follows from Lemma 4.8 that the overall resource demand of the entire item list is at least the sum of the resource demands of all the reference periods in $\mathbf{P}(\mathbf{I}_{\mathrm{L},2}) \cup \mathbf{P}(\mathbf{J}) \cup \mathbf{P}(\mathbf{S})$. To approximate the overall resource demand more closely, in the next section, we further introduce some extra reference periods that do not intersect with those in $\mathbf{P}(\mathbf{I}_{\mathrm{L},2}) \cup \mathbf{P}(\mathbf{J}) \cup \mathbf{P}(\mathbf{S})$.

4.3.2 Adding Extra Reference Periods

We define the length of a joint period as the total length of the two periods in the pair, that is, $len(J_{i,i'}) = len(I_{i,1}) + len(I_{i',1})$. Note that by Lemma 4.4(b), $I^+_{i',1} \ge I^-_{i,1} + len(J_{i,i'})$.

According to Feature (f.1) and Lemma 4.4(e), the maximum possible length of a joint period is $2\Delta + (\mu+4)\Delta = (\mu+6)\Delta$. We further divide the joint periods in $\mathbf{J}$ into *long joint periods* and *short joint periods*. Long joint periods are those longer than $(\mu+3)\Delta$, and short joint periods are those shorter than or equal to $(\mu+3)\Delta$. Let $\mathbf{J}_{\mathrm{L}}$ and $\mathbf{J}_{\mathrm{S}}$ denote the sets of long joint periods and short joint periods respectively. Then, $\mathbf{J} = \mathbf{J}_{\mathrm{L}} \cup \mathbf{J}_{\mathrm{S}}$.

For each long joint period $J_{i,i'}$, we are going to introduce an *extra reference point* $t^{\sharp}_{i',1}$ in the interval $[I^-_{i',1}, I^+_{i',1} - 2\Delta)$. Note that this is a non-empty interval since, by Lemma 4.4(e) and the definition of long joint period, $len(I_{i',1}) > (\mu+3)\Delta - len(I_{i,1}) > (\mu+3)\Delta - 2\Delta = (\mu+1)\Delta > 2\Delta$. If $I^-_{i',1} \ge I^+_{i',1} - (\mu+2)\Delta$, then $t^{\sharp}_{i',1}$ is simply our old $t^{\dagger}_{i',1} = I^-_{i',1}$. Otherwise, let $t^{\sharp}_{i',1}$ be the time when the first new item was packed into bin $b_{i'}$ during the interval $[I^+_{i',1} - (\mu+2)\Delta, I^+_{i',1} - 2\Delta)$. Such a time must exist, since the interval is of length $\mu\Delta$, and if no items were packed in that interval, the bin would have become empty at or before time $I^+_{i',1} - 2\Delta$ contradicting the fact that it remained open until time $I^+_{i',1}$.

Similar to the reference point, Lemma 4.3 also applies to the extra reference point. So, there must exist at least one bin b_h satisfying $h < i'$ and $t^{\sharp}_{i',1} < I^+_h$. Among all these bins, we define the last opened bin (the bin with the highest index) as the *extra reference bin* of $J_{i,i'}$, and denote it by $b^{\sharp}_{i',1}$. If the length of the long joint period fulfils $len(J_{i,i'}) \ge (\mu+4)\Delta$, the *extra reference period* of $J_{i,i'}$ associated with bin $b^{\sharp}_{i',1}$, denoted by $p^{\sharp}_{i',1}$, is defined to be the time interval $[t^{\sharp}_{i',1} - \Delta, t^{\sharp}_{i',1} + \Delta)$. Otherwise, it is the time interval $[t^{\sharp}_{i',1} - \delta_{i,i'}, t^{\sharp}_{i',1} + \delta_{i,i'})$, where $\delta_{i,i'} = len(J_{i,i'}) - (\mu+3)\Delta < \Delta$. Some examples of the extra reference periods of long joint periods are shown in Figures 8(a), 8(b) and 8(c). It can be proven that the extra reference period does not intersect with the standard reference period for $J_{i,i'}$. For the case where $len(J_{i,i'}) \ge (\mu+4)\Delta$, since we have $t^{\sharp}_{i',1} \ge I^+_{i',1} - (\mu+2)\Delta \ge I^-_{i,1} + 2\Delta = t^{\dagger}_{i,1} + 2\Delta$, the two reference points are too far apart for their reference periods to intersect. For the case where $(\mu+3)\Delta < len(J_{i,i'}) < (\mu+4)\Delta$, we have $t^{\sharp}_{i',1} - (t^{\dagger}_{i,1} + \Delta) \ge I^+_{i',1} - (\mu+2)\Delta - t^{\dagger}_{i,1} - \Delta = I^+_{i',1} - I^-_{i,1} - (\mu+3)\Delta \ge len(J_{i,i'}) - (\mu+3)\Delta = \delta_{i,i'}$. Hence, the extra reference period and the standard reference period do not intersect either.

According to First Fit, the level of the extra reference bin $b^{\sharp}_{i',1}$ must be higher than $1 - \frac{1}{k}$ at the extra reference point $t^{\sharp}_{i',1}$. Denote by $u(p^{\sharp}_{i',1})$ the total resource demand of the items in bin $b^{\sharp}_{i',1}$ over the extra reference period $p^{\sharp}_{i',1}$. Since each item resides in the system for at least Δ time (the minimum item interval length), in the case of $len(J_{i,i'}) \ge (\mu+4)\Delta$, we have

$$u(p^{\sharp}_{i',1}) \ge \left(1 - \frac{1}{\beta}\right) \cdot \Delta \tag{9}$$

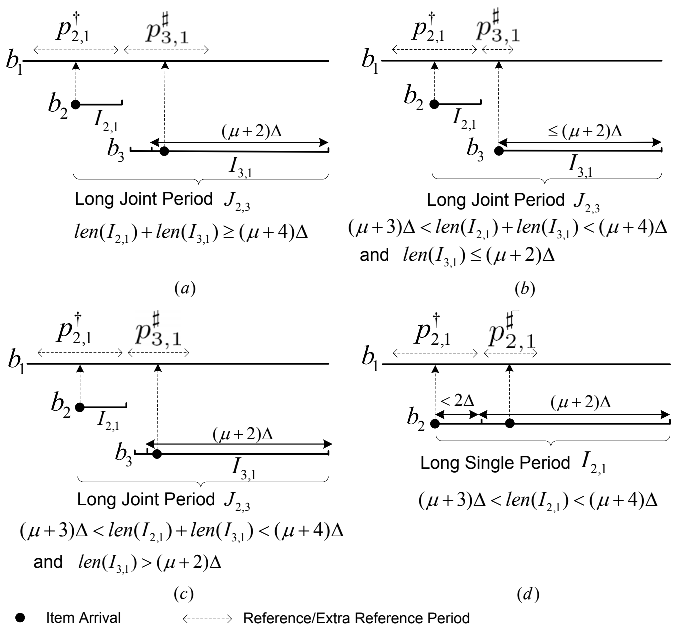


Fig. 8. Examples of extra reference periods

In the case of $(\mu+3)\Delta < len(J_{i,i'}) < (\mu+4)\Delta$, since the extra reference period $p^{\sharp}_{i',1}$ extends from $t^{\sharp}_{i',1}$ forwards and backwards each by $\delta_{i,i'} < \Delta$, the resource demand $u(p^{\sharp}_{i',1})$ must satisfy

$$
\begin{aligned}
u(p^{\sharp}_{i',1}) &\geq \left(1 - \frac{1}{\beta}\right) \cdot \delta_{i,i'} \\
&= \left(1 - \frac{1}{\beta}\right) \cdot \Big(len(J_{i,i'}) - (\mu+3)\Delta\Big) \qquad (10)
\end{aligned}
$$

According to Feature (f.1), the maximum possible length of a single period is $(\mu{+}4)\Delta$. Similar to the classification of joint periods, we also divide the single periods in $\mathbf{S}$ into *long single periods* and *short single periods*. Long single periods are those longer than $(\mu{+}3)\Delta$, and short single periods are those shorter than or equal to $(\mu+3)\Delta$. Let $\mathbf{S}_{\mathrm{L}}$ and $\mathbf{S}_{\mathrm{S}}$ denote the sets of long single periods and short single periods respectively. Then, $\mathbf{S} = \mathbf{S}_{\mathrm{L}} \cup \mathbf{S}_{\mathrm{S}}$.

For each long single period $I_{i,1}$, its reference point is $t^{\dagger}_{i,1} = I^{-}_{i,1}$ (Feature (f.4)). Since $len(I_{i,1}) = I^{+}_{i,1} - I^{-}_{i,1} > (\mu+3)\Delta$, we have $I^{+}_{i,1} - (\mu{+}2)\Delta > I^{-}_{i,1}$. Recall that $\mu\Delta$ is the maximum item interval length. Thus, at least one new item must be packed into bin b_i during the interval $[I^{+}_{i,1} - (\mu+2)\Delta, I^{+}_{i,1} - 2\Delta)$. Let $t^{\sharp}_{i,1}$ denote the time when the first new item is packed into b_i during $[I^{+}_{i,1} - (\mu+2)\Delta, I^{+}_{i,1} - 2\Delta)$. We refer to $t^{\sharp}_{i,1}$ as the *extra reference point* of the long single period $I_{i,1}$. Again, Lemma 4.3 applies to the extra reference point. So, there must exist at least one bin b_h satisfying $h < i$ and $t^{\sharp}_{i,1} < I^{+}_{h}$. Among all these bins, we define the last opened bin (the bin with the highest index) as the *extra reference bin* of $I_{i,1}$, and denote it by $b^{\sharp}_{i,1}$. The *extra reference period* of $I_{i,1}$ associated with bin $b^{\sharp}_{i,1}$, denoted by $p^{\sharp}_{i,1}$, is $[t^{\sharp}_{i,1} - \delta_i, t^{\sharp}_{i,1} + \delta_i)$, where $\delta_i = len(I_{i,1}) - (\mu+3)\Delta < \Delta$. An example of the extra reference period of a long single period is shown in Figure 8(d). Note that $t^{\sharp}_{i,1} - (t^{\dagger}_{i,1} + \Delta) \geq I^{+}_{i,1} - (\mu{+}2)\Delta - I^{-}_{i,1} - \Delta = len(I_{i,1}) - (\mu+3)\Delta = \delta_i$. Hence, the extra reference period does not intersect with the standard reference period for $I_{i,1}$. Since $\delta_i < \Delta$, similar to (10), the resource demand $u(p^{\sharp}_{i,1})$ for the extra reference period must satisfy

$$
\begin{aligned}
u(p^{\sharp}_{i,1}) &\geq \left(1 - \frac{1}{\beta}\right) \cdot \delta_i \\
&= \left(1 - \frac{1}{\beta}\right) \cdot (len(I_{i,1}) - (\mu+3)\Delta) \qquad (11)
\end{aligned}
$$

Lemma 4.4 is also applicable to the extra reference periods of long joint periods and long single periods. In the following, we analyze the intersection for the extra reference periods.

Lemma 4.9. *No extra reference period intersects with another valid reference period, either another extra one or one corresponding to a reference point for a period other than the second component for a joint period.*

Proof: Consider an extra reference period $p^{\sharp}$ in bin b_h. Suppose that this period is generated by an extra reference point $t^{\sharp}$ in bin b_i (in particular, $t^{\sharp}$ is in $I_{i,1}$) and suppose that the extra reference period is intersected by another reference period in b_h, generated by a reference point t (extra or standard) in bin b_j. Let t^{+} be the right endpoint of the joint/single period containing $t^{\sharp}$, and recall that by our choices of extra reference points, we must have $t^{\sharp} \leq t^{+} - 2\Delta$.

If $j > i$, we know that all reference points in bin b_j that map to bin b_h must be at least as large as I^{+}_{i}, and thus be at least 2Δ away from $t^{\sharp}$, and hence cannot yield intersecting reference periods.

If $j = i$, then t cannot come from the same period as $t^{\sharp}$, since we have already seen that the extra and standard reference periods for reference points in the same single period do not intersect. So, this means that t is in some period $I_{i,g}$ with $g > 1$, and hence is to the right of the interval $I_{i,1}$ containing $t^{\sharp}$. But, as already observed, $t^{\sharp}$ is at least 2Δ from the right endpoint of that interval, and hence at least that far from t, so the two reference periods cannot intersect.

Finally, assume $j < i$, in which case we must have $t < I^{+}_{j} \leq t^{\sharp}$. If t comes from an interval $I_{j,g}$ with $g > 1$, we know that the interval has length $(\mu+2)\Delta$ and t was the first item to enter the bin during the interval. Hence, it must have arrived by time $t^{-}_{j,g} + \mu\Delta \leq I^{+}_{j,g} - 2\Delta \leq I^{+}_{j} - 2\Delta$, and once again the two points are too far apart to yield intersecting reference periods. So, t must come from the interval $I_{j,1}$. If it is an extra reference point, then it is at least 2Δ to the left of I^{+}_{j}, and once again is too far from $t^{\sharp}$. If t is a standard reference point, then it must be $I^{-}_{j,1}$. But now consider the subperiod $I_{i,1}$ of bin b_i that contains $t^{\sharp}$. Since $j < i$, we must have $t = I^{-}_{j,1} \leq I^{-}_{i,1} = t^{\dagger}_{i,1}$, and if the reference period for $t^{\sharp}$ intersects that for t, then so does that for $t^{\dagger}_{i,1}$. Consequently, $I_{j,1}$ is the front-intersect period for $I_{i,1}$. Since t is a standard reference point, interval $I_{j,1}$ cannot be the second half of a joint period. But this means that our joint period construction routine must have made a joint period out of $I_{j,1}$ and $I_{i,1}$, and we have already observed that the standard and extra reference periods for a joint period do not intersect.

Thus, all possibilities lead to a contradiction, and the Lemma holds. ∎

Let set $\mathbf{P}^\sharp(\mathbf{J}_\mathrm{L})$ denote the extra reference periods of all the long joint periods in $\mathbf{J}_\mathrm{L}$, and let set $\mathbf{P}^\sharp(\mathbf{S}_\mathrm{L})$ denote the extra reference periods of all the long single periods in $\mathbf{S}_\mathrm{L}$. Lemmas 4.8 and 4.9 together have shown that all the extra reference periods in $\mathbf{P}^\sharp(\mathbf{J}_\mathrm{L}) \cup \mathbf{P}^\sharp(\mathbf{S}_\mathrm{L})$ and all the standard reference periods in $\mathbf{P}(\mathbf{I}_{\mathrm{L},2}) \cup \mathbf{P}(\mathbf{J}) \cup \mathbf{P}(\mathbf{S})$ do not intersect with one another. Thus, their total resource demand is given by the sum of their individual resource demands.

4.3.3 Calculating Total Resource Demand

Next, we calculate the ratio of the total resource demand to the total length of the reference and extra reference periods. Recall that $\mathbf{J} = \mathbf{J}_\mathrm{L} \cup \mathbf{J}_\mathrm{S}$ and $\mathbf{S} = \mathbf{S}_\mathrm{L} \cup \mathbf{S}_\mathrm{S}$. It follows from (8) that $\mathbf{I}_\mathrm{L} = \mathbf{I}_{\mathrm{L},2} \cup \mathbf{J}_\mathrm{L} \cup \mathbf{J}_\mathrm{S} \cup \mathbf{S}_\mathrm{L} \cup \mathbf{S}_\mathrm{S}$. For convenience, we also divide $\mathbf{P}(\mathbf{J})$ into $\mathbf{P}(\mathbf{J}_\mathrm{L})$ and $\mathbf{P}(\mathbf{J}_\mathrm{S})$, which are the reference periods of the long joint periods and the short joint periods respectively. Similarly, we divide $\mathbf{P}(\mathbf{S})$ into $\mathbf{P}(\mathbf{S}_\mathrm{L})$ and $\mathbf{P}(\mathbf{S}_\mathrm{S})$, which are the reference periods of the long single periods and the short single periods respectively.

For each regular period $I_{i,j} \in \mathbf{I}_{\mathrm{L},2}$, according to (7), its reference period $p^\dagger_{i,j}$ has the resource demand $u(p^\dagger_{i,j}) \geq (1 - \frac{1}{\beta})\Delta$. Since $len(I_{i,j}) = (\mu+2)\Delta$ (Feature (f.2)), it follows that

$$u(p^\dagger_{i,j}) \geq \frac{(1-\frac{1}{\beta})\Delta}{(\mu+2)\Delta} \cdot len(I_{i,j}) > \frac{1-\frac{1}{\beta}}{\mu+3} \cdot len(I_{i,j})$$

For convenience, we extend the notation $u(\cdot)$ to a set of reference periods. For example, $u(\mathbf{P}(\mathbf{I}_{\mathrm{L},2}))$ represents the total resource demand of the reference periods in $\mathbf{P}(\mathbf{I}_{\mathrm{L},2})$. Then, we have

$$\begin{aligned} u(\mathbf{P}(\mathbf{I}_{\mathrm{L},2})) &= \sum_{p^\dagger_{i,j} \in \mathbf{P}(\mathbf{I}_{\mathrm{L},2})} u(p^\dagger_{i,j}) \\ &> \frac{1-\frac{1}{\beta}}{\mu+3} \cdot \sum_{I_{i,j} \in \mathbf{I}_{\mathrm{L},2}} len(I_{i,j}) \\ &= \frac{1-\frac{1}{\beta}}{\mu+3} \cdot len(\mathbf{I}_{\mathrm{L},2}) \end{aligned} \tag{12}$$

where $len(\mathbf{I}_{\mathrm{L},2})$ is the total length of the periods in $\mathbf{I}_{\mathrm{L},2}$.

For each long joint period $J_{i,i'} \in \mathbf{J}_\mathrm{L}$, if $len(J_{i,i'}) \geq (\mu+4)\Delta$, according to (7) and (9), its reference period $p^\dagger_{i,1}$ has the resource demand $u(p^\dagger_{i,1}) \geq (1-\frac{1}{\beta})\Delta$, and its extra reference period $p^\sharp_{i',1}$ has the resource demand $u(p^\sharp_{i',1}) \geq (1-\frac{1}{\beta})\Delta$. Based on Lemma 4.4(e) and Feature (f.1), we have $len(J_{i,i'}) < 2\Delta + (\mu+4)\Delta = (\mu+6)\Delta$. Thus,

$$\begin{aligned} u(p^\dagger_{i,1}) + u(p^\sharp_{i',1}) &\geq 2 \cdot \Big(1 - \frac{1}{\beta}\Big)\Delta \\ &> \frac{2 \cdot (1-\frac{1}{\beta})\Delta}{(\mu+6)\Delta} \cdot len(J_{i,i'}) \\ &> \frac{1-\frac{1}{\beta}}{\mu+3} \cdot \Big(len(I_{i,1}) + len(I_{i',1})\Big) \end{aligned}$$

If $(\mu+3)\Delta < len(J_{i,i'}) < (\mu+4)\Delta$, according to (10), the extra reference period $p^\sharp_{i',1}$ has the resource demand $u(p^\sharp_{i',1}) \geq (1-\frac{1}{\beta}) \cdot (len(J_{i,i'}) - (\mu+3)\Delta)$. Thus,

$$\begin{aligned} &u(p^\dagger_{i,1}) + u(p^\sharp_{i',1}) \\ &\geq \Big(1-\frac{1}{\beta}\Big) \cdot \Big(\Delta + len(J_{i,i'}) - (\mu+3)\Delta\Big) \\ &= \Big(1-\frac{1}{\beta}\Big) \cdot \Big(len(J_{i,i'}) - (\mu+2)\Delta\Big) \end{aligned}$$

Since $len(J_{i,i'}) > (\mu+3)\Delta$, we have

$$\left(\frac{\mu+2}{\mu+3}\right) \cdot len(J_{i,i'}) > (\mu+2)\Delta$$

and hence

$$\begin{aligned} len(J_{i,i'}) - (\mu+2)\Delta &> \left(1 - \frac{\mu+2}{\mu+3}\right) \cdot len(J_{i,i'}) \\ &= \frac{len(J_{i,i'})}{\mu+3} \end{aligned}$$

As a result, it again holds that

$$\begin{aligned} u(p^\dagger_{i,1}) + u(p^\sharp_{i',1}) &> \frac{1-\frac{1}{\beta}}{\mu+3} \cdot len(J_{i,i'}) \\ &= \frac{1-\frac{1}{\beta}}{\mu+3} \cdot \Big(len(I_{i,1}) + len(I_{i',1})\Big) \end{aligned}$$

Therefore,

$$\begin{aligned} &u(\mathbf{P}(\mathbf{J}_\mathrm{L}) \cup \mathbf{P}^\sharp(\mathbf{J}_\mathrm{L})) \\ &= \sum_{p^\dagger_{i,1} \in \mathbf{P}(\mathbf{J}_\mathrm{L})} u(p^\dagger_{i,1}) + \sum_{p^\sharp_{i',1} \in \mathbf{P}^\sharp(\mathbf{J}_\mathrm{L})} u(p^\sharp_{i',1}) \\ &> \frac{1-\frac{1}{\beta}}{\mu+3} \cdot \sum_{J_{i,i'} \in \mathbf{J}_\mathrm{L}} \Big(len(I_{i,1}) + len(I_{i',1})\Big) \\ &= \frac{1-\frac{1}{\beta}}{\mu+3} \cdot len(\mathbf{J}_\mathrm{L}) \end{aligned} \tag{13}$$

For each short joint period $J_{i,i'} \in \mathbf{J}_\mathrm{S}$, according to (7), its reference period $p^\dagger_{i,1}$ has the resource demand $u(p^\dagger_{i,1}) \geq (1-\frac{1}{\beta})\Delta$. Since $len(J_{i,i'}) \leq (\mu+3)\Delta$, it follows that

$$\begin{aligned} u(p^\dagger_{i,1}) &\geq \frac{(1-\frac{1}{\beta})\Delta}{(\mu+3)\Delta} \cdot len(J_{i,i'}) \\ &= \frac{1-\frac{1}{\beta}}{\mu+3} \cdot \Big(len(I_{i,1}) + len(I_{i',1})\Big) \end{aligned}$$

Thus,

$$\begin{aligned} u(\mathbf{P}(\mathbf{J}_\mathrm{S})) &= \sum_{p^\dagger_{i,1} \in \mathbf{P}(\mathbf{J}_\mathrm{S})} u(p^\dagger_{i,1}) \\ &\geq \frac{1-\frac{1}{\beta}}{\mu+3} \cdot \sum_{J_{i,i'} \in \mathbf{J}_\mathrm{S}} \Big(len(I_{i,1}) + len(I_{i',1})\Big) \\ &= \frac{1-\frac{1}{\beta}}{\mu+3} \cdot len(\mathbf{J}_\mathrm{S}) \end{aligned} \tag{14}$$

For each long single period $I_{i,1} \in \mathbf{S}_\mathrm{L}$, remember that $(\mu+3)\Delta < len(I_{i,1}) < (\mu+4)\Delta$. According to (7) and (11), its reference period $p^\dagger_{i,1}$ has the resource demand $u(p^\dagger_{i,1}) \geq$

$(1-\frac{1}{\beta})\Delta$, and its extra-reference period $p^{\sharp}_{i,1}$ has the resource demand $u(p^{\sharp}_{i,1}) \geq (1-\frac{1}{\beta}) \cdot (len(I_{i,1}) - (\mu+3)\Delta)$. Thus,

$$\begin{aligned} u(p^{\dagger}_{i,1}) + u(p^{\sharp}_{i,1}) &\geq \left(1-\frac{1}{\beta}\right) \cdot \Big(\Delta + len(I_{i,1}) - (\mu+3)\Delta\Big) \\ &= \left(1-\frac{1}{\beta}\right) \cdot \Big(len(I_{i,1}) - (\mu+2)\Delta\Big) \end{aligned}$$

Since $len(I_{i,1}) > (\mu+3)\Delta$, we have

$$\left(\frac{\mu+2}{\mu+3}\right) \cdot len(I_{i,1}) > (\mu+2)\Delta$$

and hence

$$\begin{aligned} len(I_{i,1}) - (\mu+2)\Delta &> \left(1-\frac{\mu+2}{\mu+3}\right) \cdot len(I_{i,1}) \\ &= \frac{len(I_{i,1})}{\mu+3} \end{aligned}$$

As a result,

$$u(p^{\dagger}_{i,1}) + u(p^{\sharp}_{i,1}) > \frac{1-\frac{1}{\beta}}{\mu+3} \cdot len(I_{i,1})$$

Therefore,

$$\begin{aligned} u(\mathbf{P}(\mathbf{S}_{\mathrm{L}}) \cup \mathbf{P}^{\sharp}(\mathbf{S}_{\mathrm{L}})) &= \sum_{p^{\dagger}_{i,1} \in \mathbf{P}(\mathbf{S}_{\mathrm{L}})} u(p^{\dagger}_{i,1}) + \sum_{p^{\sharp}_{i,1} \in \mathbf{P}^{\sharp}(\mathbf{S}_{\mathrm{L}})} u(p^{\sharp}_{i,1}) \\ &> \frac{1-\frac{1}{\beta}}{\mu+3} \cdot \sum_{I_{i,1} \in \mathbf{S}_{\mathrm{L}}} len(I_{i,1}) \\ &= \frac{1-\frac{1}{\beta}}{\mu+3} \cdot len(\mathbf{S}_{\mathrm{L}}) \end{aligned} \tag{15}$$

For each short single period $I_{i,1} \in \mathbf{S}_{\mathrm{S}}$, according to (7), its reference period $p^{\dagger}_{i,1}$ has the resource demand $u(p^{\dagger}_{i,1}) \geq (1-\frac{1}{\beta})\Delta$. Since $len(I_{i,1}) \leq (\mu+3)\Delta$, it follows that

$$u(p^{\dagger}_{i,1}) \geq \frac{(1-\frac{1}{\beta})\Delta}{(\mu+3)\Delta} \cdot len(I_{i,1}) = \frac{1-\frac{1}{\beta}}{\mu+3} \cdot len(I_{i,1})$$

Thus,

$$\begin{aligned} u(\mathbf{P}(\mathbf{S}_{\mathrm{S}})) &= \sum_{p^{\dagger}_{i,1} \in \mathbf{P}(\mathbf{S}_{\mathrm{S}})} u(p^{\dagger}_{i,1}) \\ &\geq \frac{1-\frac{1}{\beta}}{\mu+3} \cdot \sum_{I_{i,1} \in \mathbf{S}_{\mathrm{S}}} len(I_{i,1}) \\ &= \frac{1-\frac{1}{\beta}}{\mu+3} \cdot len(\mathbf{S}_{\mathrm{S}}) \end{aligned} \tag{16}$$

Combining (12), (13), (14), (15), and (16), we have

$$\begin{aligned} & u(\mathbf{P}(\mathbf{I}_{\mathrm{L},2}) \cup \mathbf{P}(\mathbf{J}) \cup \mathbf{P}(\mathbf{S}) \cup \mathbf{P}^{\sharp}(\mathbf{J}_{\mathrm{L}}) \cup \mathbf{P}^{\sharp}(\mathbf{S}_{\mathrm{L}})) \\ &= u(\mathbf{P}(\mathbf{I}_{\mathrm{L},2}) \cup \mathbf{P}(\mathbf{J}_{\mathrm{L}}) \cup \mathbf{P}(\mathbf{J}_{\mathrm{S}}) \cup \mathbf{P}(\mathbf{S}_{\mathrm{L}}) \cup \mathbf{P}(\mathbf{S}_{\mathrm{S}}) \\ &\qquad \cup \mathbf{P}^{\sharp}(\mathbf{J}_{\mathrm{L}}) \cup \mathbf{P}^{\sharp}(\mathbf{S}_{\mathrm{L}})) \\ &= u(\mathbf{P}(\mathbf{I}_{\mathrm{L},2})) + u(\mathbf{P}(\mathbf{J}_{\mathrm{L}}) \cup \mathbf{P}^{\sharp}(\mathbf{J}_{\mathrm{L}})) + u(\mathbf{P}(\mathbf{J}_{\mathrm{S}})) \\ &\qquad + u(\mathbf{P}(\mathbf{S}_{\mathrm{L}}) \cup \mathbf{P}^{\sharp}(\mathbf{S}_{\mathrm{L}})) + u(\mathbf{P}(\mathbf{S}_{\mathrm{S}})) \\ &\geq \frac{1-\frac{1}{\beta}}{\mu+3} \cdot (len(\mathbf{I}_{\mathrm{L},2}) + len(\mathbf{J}_{\mathrm{L}}) + len(\mathbf{J}_{\mathrm{S}}) \\ &\qquad + len(\mathbf{S}_{\mathrm{L}}) + len(\mathbf{S}_{\mathrm{S}})) \\ &= \frac{1-\frac{1}{\beta}}{\mu+3} \cdot len(\mathbf{I}_{\mathrm{L}}) \end{aligned} \tag{17}$$

The overall resource demand $u(\mathcal{R})$ of the entire item list is at least $u(\mathbf{P}(\mathbf{I}_{\mathrm{L},2}) \cup \mathbf{P}(\mathbf{J}) \cup \mathbf{P}(\mathbf{S}) \cup \mathbf{P}^{\sharp}(\mathbf{J}_{\mathrm{L}}) \cup \mathbf{P}^{\sharp}(\mathbf{S}_{\mathrm{L}}))$. Thus, it follows that

$$u(\mathcal{R}) \geq \frac{1-\frac{1}{\beta}}{\mu+3} \cdot len(\mathbf{I}_{\mathrm{L}}) \tag{18}$$

According to (5), (6) and (18), we have

$$\begin{aligned} TotalCost(P_{FF,\mathcal{R}}) &= len(\mathbf{I}_{\mathrm{L}}) + span(\mathcal{R}) \\ &\leq \frac{\mu+3}{1-\frac{1}{\beta}} \cdot u(\mathcal{R}) + span(\mathcal{R}) \end{aligned} \tag{19}$$

It follows from Bounds (b.1) and (b.2) that

$$TotalCost(P_{FF,\mathcal{R}}) \leq \left(\frac{\beta}{\beta-1} \cdot \mu + \frac{3\beta}{\beta-1} + 1\right) \cdot OPT_{total}(\mathcal{R})$$

Therefore, we have the following result.

Theorem 4.10. *For the MinTotal DBP problem, for any item list $\mathcal{R}$, if the item size $s(r) < \frac{1}{\beta}$ ($\beta > 1$ is a constant) for all the items $r \in \mathcal{R}$, the total cost of First Fit is at most $(\frac{\beta}{\beta-1} \cdot \mu + \frac{3\beta}{\beta-1} + 1) \cdot OPT_{total}(\mathcal{R})$.* □

4.3.4 The General Case

Now, we consider the general case for First Fit. We follow the above analysis for Theorem 4.10.

Consider a period $I_{i,j}$ in $\mathbf{I}_{\mathrm{L},2} \cup \mathbf{S}_{\mathrm{L}} \cup \mathbf{S}_{\mathrm{S}}$ or a joint period $J_{i,i'}$ in $\mathbf{J}_{\mathrm{L}} \cup \mathbf{J}_{\mathrm{S}}$. Recall that its reference period $p^{\dagger}_{i,j}$ is the time interval $[t^{\dagger}_{i,j} - \Delta, t^{\dagger}_{i,j} + \Delta)$ associated with the reference bin $b^{\dagger}_{i,j}$. Let $p^{\ddagger}_{i,j}$ denote the same time interval $[t^{\dagger}_{i,j} - \Delta, t^{\dagger}_{i,j} + \Delta)$ associated with bin b_i. We refer to $p^{\ddagger}_{i,j}$ as the *auxiliary period* of $I_{i,j}$ or $J_{i,i'}$. Figure 9 shows an example of auxiliary periods.

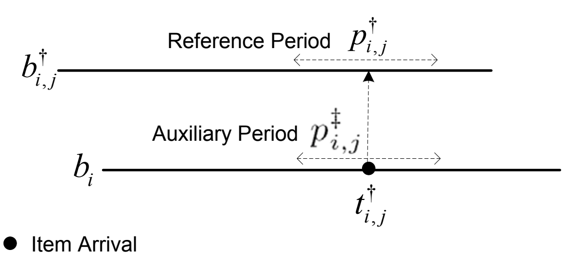


Fig. 9. An example of auxiliary periods

In the above analysis for Theorem 4.10, we have shown that for each reference period $p^{\dagger}_{i,j}$, a new item was packed into b_i at time $t^{\dagger}_{i,j}$. According to First Fit, after this item is packed, the total level of bins b_i and $b^{\dagger}_{i,j}$ should exceed 1. Otherwise, the new item would have been packed into $b^{\dagger}_{i,j}$ instead. Moreover, since Δ is the minimum item interval length, all the items in bin b_i at time $t^{\dagger}_{i,j}$ must reside in the system for at least Δ time during the auxiliary period $p^{\ddagger}_{i,j}$, and all the items in bin $b^{\dagger}_{i,j}$ at time $t^{\dagger}_{i,j}$ must reside in the system for at least Δ time during the reference period $p^{\dagger}_{i,j}$. It follows that the total resource demand of the items in bin b_i over $p^{\ddagger}_{i,j}$ and the items in bin $b^{\dagger}_{i,j}$ over $p^{\dagger}_{i,j}$ satisfies

$$u(p^{\dagger}_{i,j}) + u(p^{\ddagger}_{i,j}) \geq \Delta \tag{20}$$

Similarly, for each long joint period $J_{i,i'}$ (and each long single period $I_{i',1}$), we define its extra auxiliary period $p^{\S}_{i',1}$ as the same time interval as its extra reference period but associated with bin $b_{i'}$. For the case of a long joint period $J_{i,i'}$ with length between $(\mu+3)\Delta$ and $(\mu+4)\Delta$, the total resource demand of the items in bin $b_{i'}$ over the extra auxiliary period $p^{\S}_{i',1}$ and the items in bin $b^{\sharp}_{i',1}$ over the extra reference period $p^{\sharp}_{i',1}$ satisfies

$$u(p^{\sharp}_{i',1}) + u(p^{\S}_{i',1}) \geq len(I_{i,1}) + len(I_{i',1}) - (\mu+3)\Delta \quad (21)$$

For the case of a long joint period $J_{i,i'}$ with length above $(\mu+4)\Delta$, the total resource demand satisfies

$$u(p^{\sharp}_{i',1}) + u(p^{\S}_{i',1}) \geq \Delta \quad (22)$$

For the case of a long single period $I_{i',1}$, the total resource demand satisfies

$$u(p^{\sharp}_{i',1}) + u(p^{\S}_{i',1}) \geq len(I_{i',1}) - (\mu+3)\Delta \quad (23)$$

According to the previous analysis, all the reference/extra reference periods in $\mathbf{P}(\mathbf{I}_{\mathrm{L},2}) \cup \mathbf{P}(\mathbf{J}) \cup \mathbf{P}(\mathbf{S}) \cup \mathbf{P}^{\sharp}(\mathbf{J}_{\mathrm{L}}) \cup \mathbf{P}^{\sharp}(\mathbf{S}_{\mathrm{L}})$ do not intersect with each other. Next, we examine the intersection among the auxiliary/extra auxiliary periods.

Lemma 4.11. *All the auxiliary/extra auxiliary periods do not intersect with one another.*

Proof: First, it is apparent that any two extra auxiliary periods $p^{\S}_{i'_1,1}$ and $p^{\S}_{i'_2,1}$ do not intersect because there is at most one per bin.

Next, we show that any two auxiliary periods $p^{\ddagger}_{i_1,j_1}$ and $p^{\ddagger}_{i_2,j_2}$ do not intersect. If $i_1 \neq i_2$, $p^{\ddagger}_{i_1,j_1}$ and $p^{\ddagger}_{i_2,j_2}$ do not intersect since they are associated with different bins. If $i_1 = i_2$, without loss of generality, suppose $j_1 < j_2$. Since there are at least two subperiods in the bin, this means that $len(I_{i_1,j_1}) \geq 2\Delta$ and so, by now familiar arguments, the reference point of I_{i_1,j_1} is at least 2Δ to the left of the right endpoint of I_{i_1,j_1}, and hence the auxiliary reference period of I_{i_1,j_1} cannot intersect with any subsequent ones. Therefore, $p^{\ddagger}_{i_1,j_1}$ and $p^{\ddagger}_{i_2,j_2}$ do not intersect.

Finally, we show that an extra auxiliary period $p^{\S}_{i',1}$ does not intersect with an auxiliary period $p^{\ddagger}_{j,h}$. If $i' \neq j$, $p^{\S}_{i',1}$ and $p^{\ddagger}_{j,h}$ do not intersect because they are associated with different bins. If $i' = j$ and $h \geq 2$, we are in the same situation as in the previous case and there can be no intersection. So, assume that $h = 1$ and hence both the auxiliary reference period and the extra auxiliary reference period come from the same subperiod $I_{j,1}$. We already argued in the proof of Lemma 4.9 that the reference and extra reference periods for this subperiod do not intersect even if they are associated with the same bin. Thus, neither do the corresponding auxiliary and extra auxiliary periods. □

Since all the reference/extra reference periods of those in $\mathbf{P}(\mathbf{I}_{\mathrm{L},2}) \cup \mathbf{P}(\mathbf{J}) \cup \mathbf{P}(\mathbf{S}) \cup \mathbf{P}^{\sharp}(\mathbf{J}_{\mathrm{L}}) \cup \mathbf{P}^{\sharp}(\mathbf{S}_{\mathrm{L}})$ do not intersect and all the auxiliary/extra auxiliary periods do not intersect either, any time point associated with each bin can be shared by at most one reference/extra reference period and one auxiliary/extra auxiliary period. Therefore, similar to the derivation of (18), it follows from (20), (21), (22) and (23) that

$$u(\mathcal{R}) \geq \frac{1}{2} \cdot \frac{1}{\mu+3} \cdot len(\mathbf{I}_{\mathrm{L}}) \quad (24)$$

According to (5), (6) and (24), we have

$$\begin{aligned} TotalCost(P_{FF,\mathcal{R}}) &\leq 2(\mu+3) \cdot u(\mathcal{R}) + span(\mathcal{R}) \\ &\leq (2\mu+6) \cdot OPT_{total}(\mathcal{R}) + OPT_{total}(\mathcal{R}) \\ &\leq (2\mu+7) \cdot OPT_{total}(\mathcal{R}) \end{aligned}$$

Therefore, we have the following result.

Theorem 4.12. *The MinTotal DBP competitive ratio of First Fit has an upper bound of* $2\mu+7$.

4.4 A Hybrid First Fit Packing Algorithm

Theorem 4.10 shows that the total cost of First Fit is much related to the item sizes. Inspired by Theorem 4.10, we propose a new Hybrid First Fit algorithm that can achieve improved competitive ratios.

- **Hybrid First Fit (HFF)**: Define a variable $\beta > 1$. The items with sizes equal to or larger than $\frac{1}{\beta}$ are classified as large items. The items with sizes smaller than $\frac{1}{\beta}$ are classified as small items. Hybrid First Fit uses the Modified First Fit algorithm defined in Section 3.2 to pack the large items and the small items separately.

Theorem 4.13. *Hybrid First Fit can achieve a MinTotal DBP competitive ratio no larger than* $\frac{5}{4}\mu + \frac{19}{4}$ *when* μ *is not known and a competitive ratio no larger than* $\mu+5$ *when* μ *is known.*

Proof: Given an item list $\mathcal{R}$, let $\mathcal{R}_{\mathrm{L}}$ denote the set of all the large items and $\mathcal{R}_{\mathrm{S}}$ denote the set of all the small items. Then, $s(r) \geq \frac{1}{\beta}$ for all $r \in \mathcal{R}_{\mathrm{L}}$, and $s(r) < \frac{1}{\beta}$ for all $r \in \mathcal{R}_{\mathrm{S}}$.

The total cost of any packing algorithm is bounded by that of assigning each item to a new bin, i.e., $\sum_{r\in\mathcal{R}} len(I(r))$. Thus, for the large items, we have

$$\begin{aligned} TotalCost(P_{HFF,\mathcal{R}_{\mathrm{L}}}) &\leq \sum_{r\in\mathcal{R}_{\mathrm{L}}} len(I(r)) \\ &= \sum_{r\in\mathcal{R}_{\mathrm{L}}} \frac{u(r)}{s(r)} \leq \frac{\sum_{r\in\mathcal{R}_{\mathrm{L}}} u(r)}{\frac{1}{\beta}} \\ &= \beta \cdot u(\mathcal{R}_{\mathrm{L}}) \end{aligned} \quad (25)$$

For the small items, according to (19) in the analysis of Theorem 4.10, we have

$$TotalCost(P_{HFF,\mathcal{R}_{\mathrm{S}}}) \leq \frac{\mu+3}{1-\frac{1}{\beta}} \cdot u(\mathcal{R}_{\mathrm{S}}) + span(\mathcal{R}_{\mathrm{S}})$$

Note that $u(\mathcal{R}_{\mathrm{L}}) \leq u(\mathcal{R})$, $u(\mathcal{R}_{\mathrm{S}}) \leq u(\mathcal{R})$, and $span(\mathcal{R}_{\mathrm{S}}) \leq span(\mathcal{R})$. Thus, it follows that

$$\begin{aligned} &TotalCost(P_{HFF,\mathcal{R}}) \\ &= TotalCost(P_{HFF,\mathcal{R}_{\mathrm{L}}}) + TotalCost(P_{HFF,\mathcal{R}_{\mathrm{S}}}) \\ &\leq \beta \cdot u(\mathcal{R}_{\mathrm{L}}) + \frac{\mu+3}{1-\frac{1}{\beta}} \cdot u(\mathcal{R}_{\mathrm{S}}) + span(\mathcal{R}_{\mathrm{S}}) \\ &\leq \max\left\{\beta, \frac{\mu+3}{1-\frac{1}{\beta}}\right\} \cdot u(\mathcal{R}) + span(\mathcal{R}) \end{aligned}$$

If the max/min item interval length ratio μ is not known, we can set $\beta = 5$ in Hybrid First Fit. In this case,

$$\max\left\{\beta, \frac{\mu+3}{1-\frac{1}{\beta}}\right\} = \max\left\{5, \frac{5}{4}\mu + \frac{15}{4}\right\}$$

Since $\mu \geq 1$, we have $\frac{5}{4}\mu + \frac{15}{4} \geq 5$. Therefore,

$$\begin{aligned} TotalCost(P_{HFF,\mathcal{R}}) &\leq \left(\frac{5}{4}\mu + \frac{15}{4}\right) \cdot u(\mathcal{R}) + span(\mathcal{R}) \\ &\leq \left(\frac{5}{4}\mu + \frac{19}{4}\right) \cdot OPT_{total}(\mathcal{R}) \end{aligned}$$

Thus, when μ is not known, Hybrid First Fit can achieve a competitive ratio no larger than $\frac{5}{4}\mu + \frac{19}{4}$.

If μ is known, it can be derived that when $\beta = \mu + 4$, $\max\left\{\beta, \frac{\mu+3}{1-\frac{1}{\beta}}\right\}$ achieves the smallest value which is given by $\mu + 4$. Therefore, we have

$$\begin{aligned} TotalCost(P_{HFF,\mathcal{R}}) &\leq (\mu+4) \cdot u(\mathcal{R}) + span(\mathcal{R}) \\ &\leq (\mu+5) \cdot OPT_{total}(\mathcal{R}) \end{aligned}$$

Thus, when μ is known,[1] Hybrid First Fit can achieve a competitive ratio no larger than $\mu + 5$. □

5 Conclusions

In this paper, we have studied the MinTotal Dynamic Bin Packing problem that aims to minimize the total cost of the bins used over time. We have analyzed the competitive ratios of appropriately modified versions of the classic Any Fit algorithms for ordinary bin packing, including bounds on the competitive ratios of arbitrary Any Fit algorithms and more specific bounds for the modified versions of Best Fit and First Fit. We also introduced a new online algorithm, Hybrid First Fit, with a better competitive ratio than we were able to prove for First Fit. There remains an open question of tightening the gap between the current upper and lower bounds on the competitive ratios of First Fit and Hybrid First Fit. Another direction for future work is to further investigate the constrained Dynamic Bin Packing problem in which each item is allowed to be assigned to only a subset of bins to cater for the interactivity constraints of dispatching playing requests among distributed clouds in cloud gaming.

Acknowledgments

This research is supported by Multi-plAtform Game Innovation Centre (MAGIC), funded by the Singapore National Research Foundation under its IDM Futures Funding Initiative and administered by the Interactive & Digital Media Programme Office, Media Development Authority. The authors would like to thank anonymous reviewers for their helpful suggestions to improve the presentation of this paper.

1. Since μ is known, Hybrid First Fit in this case is a semi-online algorithm. In certain applications such as cloud gaming, it is possible to estimate the max/min item interval length ratio μ according to the statistics of historical playing data.

References

[1] Distribution and monetization strategies to increase revenues from cloud gaming. http://www.cgconfusa.com/report/documents/Content-5minCloudGamingReportHighlights.pdf.

[2] Gaikai. http://www.gaikai.com/.

[3] Onlive. http://www.onlive.com/.

[4] Streammygame. http://www.streammygame.com/smg/index.php.

[5] 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.

[6] A. Bar-Noy, R. Bar-Yehuda, A. Freund, J. S. Naor, and B. Schieber. A unified approach to approximating resource allocation and scheduling. *Journal of the ACM*, 48(5):735–744, Sept. 2001.

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

[8] 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.

[9] 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.

[10] K.-T. Chen, Y.-C. Chang, P.-H. Tseng, C.-Y. Huang, and C.-L. Lei. Measuring the latency of cloud gaming systems. In *Proc. ACM MM'11*, pages 1269–1272. ACM, 2011.

[11] E. G. Coffman, J. Csirik, G. Galambos, S. Martello, and D. Vigo. Bin packing approximation algorithms: Survey and classification. *Handbook of Combinatorial Optimization (second ed.)*, pages 455–531, 2013.

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

[13] E. G. Coffman, Jr., M. R. Garey, and D. S. Johnson. Approximation algorithms for bin packing: A survey. *Approximation Algorithms for NP-hard Problems*, pages 46–93, 1997.

[14] 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. IEEE, 2009.

[15] G. Galambos and G. J. Woeginger. On-line bin packingła restricted survey. *Zeitschrift für Operations Research*, 42(1):25–45, 1995.

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

[17] C.-Y. Huang, C.-H. Hsu, Y.-C. Chang, and K.-T. Chen. Gaminganywhere: an open cloud gaming system. In *Proc. ACM MMSys'13*, pages 36–47. ACM, 2013.

[18] 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.

[19] J. W. Jiang, T. Lan, S. Ha, M. Chen, and M. Chiang. Joint vm placement and routing for data center traffic engineering. In *Proc. IEEE INFOCOM'12*, pages 2876–2880. IEEE, 2012.

[20] 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.

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

[22] 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. IEEE, 2012.

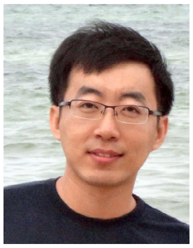
[23] P. E. Ross. Cloud computing's killer app: Gaming. *IEEE Spectrum*, 46(3):14–14, 2009.

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

[25] C. Sharon, W. Bernard, G. Simon, C. Rosenberg, et al. The brewing storm in cloud gaming: A measurement study on cloud to end-user latency. In *Proc. ACM NetGames'12*, 2012.

[26] A. Stolyar. An infinite server system with general packing constraints. *arXiv preprint arXiv:1205.4271*, 2012.

[27] C. Zhang, Z. Qi, J. Yao, M. Yu, and H. Guan. vgasa: Adaptive scheduling algorithm of virtualized gpu resource in cloud gaming. *IEEE Transactions on Parallel and Distributed Systems*, accepted to appear, 2014.



Yusen Li is currently a Research Fellow in the School of Computer Engineering at Nanyang Technological University. His research interests include scheduling, resource management issues in distributed systems and cloud computing.

Xueyan Tang received the BEng degree in computer science and engineering from Shanghai Jiao Tong University in 1998, and the PhD degree in computer science from the Hong Kong University of Science and Technology in 2003. He is currently an associate professor in the School of Computer Engineering at Nanyang Technological University, Singapore. He has served as an associate editor of IEEE Transactions on Parallel and Distributed Systems. His research interests include distributed systems, mobile and pervasive computing, and wireless sensor networks. He is a senior member of the IEEE.

Wentong Cai is a Professor in the School of Computer Engineering at Nanyang Technological University, Singapore. His expertise is mainly in the areas of Modeling and Simulation and Parallel and Distributed Computing. He is an associate editor of the ACM Transactions on Modeling and Computer Simulation (TOMACS) and an editor of the Future Generation Computer Systems (FGCS). He has chaired a number of international conferences. Most recent ones include CloudCom 2014, SIMUTools 2013, ICPADS 2012, and MACOTS 2011.