Busy-Time Scheduling on Heterogeneous Machines

Runtian Ren
School of Computer Science and Engineering
Nanyang Technological University, Singapore 639798
renr0002@ntu.edu.sg

Xueyan Tang
School of Computer Science and Engineering
Nanyang Technological University, Singapore 639798
asxytang@ntu.edu.sg

***Abstract*—We study a busy-time scheduling problem on heterogeneous machines (BSHM) which is motivated by server acquisition and task dispatching in cloud computing. The input of BSHM is a set of interval jobs, each specified by a size, an arrival time and a departure time. When a job arrives, it must be placed onto a machine immediately. The execution of a job cannot be interrupted until it departs. At any time, the total size of the jobs running on a machine cannot exceed the machine's capacity. m different types of machines are available and abundant machines are provided for each type. A type-i machine has a capacity g_i and is charged at a cost rate r_i when busy (running jobs). The target of BSHM is to schedule the given set of jobs onto machines with the minimum accumulated cost. Suppose the machine types are sorted by their capacities so that $g_1 \leq g_2 \leq \cdots \leq g_m$. We first consider two typical cases of BSHM. In BSHM-DEC, $\frac{r_i}{g_i} \geq \frac{r_{i+1}}{g_{i+1}}$ holds for each i. In BSHM-INC, $\frac{r_i}{g_i} \leq \frac{r_{i+1}}{g_{i+1}}$ holds for each i. For each case, we propose a $O(1)$-approximation algorithm in the offline setting and a $O(\mu)$-competitive algorithm in the non-clairvoyant online setting. Finally, we discuss how the scheduling strategies developed for these two cases can be combined to deal with the general BSHM problem.**

I. Introduction

We study the following busy-time scheduling problem on heterogeneous machines (BSHM). Each job is specified by a size, an arrival time and a departure time. Once a job arrives, it must be placed onto a machine and start running immediately. The execution of a job cannot be interrupted until it departs. At any time, the total size of the jobs running on a machine cannot exceed the machine's capacity. Assume that m different types of machines are available and abundant machines are provided for each type. A type-i machine has a capacity g_i and is charged at a cost rate r_i when being busy (running at least one job). The target of BSHM is to schedule all the jobs onto machines with the minimum accumulated cost.

Our BSHM problem is motivated by the server acquisition and job dispatching issues in cloud computing. In the past decade, renting computing resources on demand from the cloud has become a popular way for users to run their jobs. Due to the low maintenance overhead of cloud services and the widely-accepted "pay-as-you-go" billing mechanism, numerous companies have moved their business onto the clouds. The major providers of cloud computing services including Amazon EC2, Google Cloud and Microsoft Azure all prepare different types of predefined virtual machines for the customers to rent [1]–[3]. A natural issue faced by cloud users is to decide on the type and number of machines to rent in order to minimize the total renting cost for processing jobs.

In this paper, we focus on two typical and common cases of BSHM. Let the machine types be sorted in an increasing order of their capacities, i.e., $g_1 \leq g_2 \leq \cdots \leq g_m$. In the first case called BSHM-DEC, we assume $\frac{r_i}{g_i} \geq \frac{r_{i+1}}{g_{i+1}}$ for each i, which implies that the amortized cost rate per resource unit decreases with the machine capacity (following for example the convention that bulk purchase can normally receive a discount). In the second case called BSHM-INC, we assume $\frac{r_i}{g_i} \leq \frac{r_{i+1}}{g_{i+1}}$ for each i, which indicates that the amortized cost rate per resource unit increases with the machine capacity (due to reasons such as new architecture design and support for higher capacity). We also discuss how the scheduling strategies developed for these two cases can be integrated to address the general BSHM problem with an arbitrary sequence of amortized cost rates over the machine types.

A. Related Work

The interval scheduling problem with bounded parallelism can be seen as a special case of BSHM and has been studied extensively over the past decade. In this problem, all the interval jobs are of uniform size and only one type of machine is available for processing jobs. At any time, each machine can run at most g jobs concurrently. The objective is to minimize the total machine usage time for processing a set of jobs. Winker and Zhang [16] first defined this problem and proved its NP-hardness through a reduction from the Circular Arc Coloring problem. Alicherry and Bhatia [4] developed a 2-approximation algorithm in the offline setting through a network flow formulation. Kumar and Rudra [10] proposed another 2-approximation algorithm based on the 2-allocation technique introduced by Gergov [8]. Flammini *et al.* [7] introduced a greedy First-Fit algorithm which gives a 4-approximation. Recently, Chang *et al.* [6] proposed a 3-approximation algorithm called GreedyTracking. In addition, several special cases (the proper case, the clique case, etc.) of the problem were investigated by Flammini *et al.* [7] and Mertzios *et al.* [12]. Shalom *et al.* [15] established a tight bound g on the competitiveness of this problem in the online setting and also studied several special cases where better competitiveness can be achieved.

BSHM is also a generalization of the MinUsageTime Dynamic Bin Packing (DBP) problem, which generalizes the interval scheduling problem with bounded parallelism by allowing each interval job to have an arbitrary size. For the offline version of the MinUsageTime DBP problem, Khandekar *et al.* [9] first proposed a 5-approximation algorithm. In our earlier work [13], we introduced a 4-approximation Dual Coloring algorithm by extending the algorithm of Kumar and Rudra [10]. There are two settings for the online version

1530-2075/20/$31.00 ©2020 IEEE
DOI 10.1109/IPDPS47924.2020.00040

of the problem: a non-clairvoyant setting and a clairvoyant setting. In the non-clairvoyant setting, the departure time of a job is not known at its arrival and thus cannot be used for the scheduling purpose. Li *et al.* [11] established a lower bound of μ on the competitiveness in this setting, where μ represents the max/min job duration ratio among all the jobs to schedule. In our earlier work [14], we showed that the First Fit packing algorithm achieves a competitive ratio of $\mu+3$ which closely matches the lower bound. In the clairvoyant setting, the departure time of a job is revealed at its arrival and thus can be used for the scheduling purpose. Azar and Vainstein [5] established a tight bound $\Theta(\sqrt{\log \mu})$ on the competitiveness in this setting. However, none of the above work has studied busy-time scheduling on heterogeneous machines.

B. Contributions

For BSHM-DEC, we first propose a $O(1)$-approximation algorithm in the offline setting which places jobs onto machines in an iterative manner. Next, based on the First-Fit rule, we propose a $O(\mu)$-competitive online algorithm in the non-clairvoyant setting, where μ is the max/min job duration ratio. For BSHM-INC, we introduce a partitioning strategy, which can be used to design a $O(1)$-approximation algorithm and a $O(\mu)$-competitive algorithm in the offline and non-clairvoyant online settings respectively. Finally, we discuss how to deal with the general case of BSHM by combining these algorithms.

II. Preliminaries

We first introduce some key notations. Given 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 $\mathrm{len}(I)=I^+-I^-$ denote the length of interval I. For notational convenience, given a set $\mathcal{I}$ of intervals that are pairwise disjoint, we shall denote their total length by $\mathrm{len}(\mathcal{I})=\sum_{I\in\mathcal{I}}\mathrm{len}(I)$.

For each job J, we use $s(J)$ to denote J's size and use $I(J)$ to denote J's *active interval* which is the time interval from its arrival to departure. Then, J's arrival and departure times can be represented by $I(J)^-$ and $I(J)^+$ respectively. We say that J is *active* during $I(J)$. The length $\mathrm{len}(I(J))$ is known as J's *duration*. Given a set of jobs $\mathcal{J}$, let $s(\mathcal{J},t)$ denote the total size of the jobs active at time t, i.e., $s(\mathcal{J},t)=\sum_{J\in\mathcal{J}:t\in I(J)} s(J)$.

m different types of machines are available for processing jobs and abundant machines are provided for each machine type. A type-i machine has a capacity g_i and is charged at a cost rate r_i when it is busy (running at least one job). For notational convenience, we define $g_0=0$. Without loss of generality, suppose the machine types are sorted in an increasing order of their capacities, i.e., $g_1<g_2<\cdots<g_m$. Consequently, we must have $r_1<r_2<\cdots<r_m$.[1] For simplicity, we further assume that each r_i is a power of 2. This can be achieved by selecting a subset of machine types for processing jobs as follows. First, we normalize the cost rates by setting $r_i\leftarrow\frac{r_i}{r_1}$ for each machine type i. Consequently, r_1 is normalized to 1. Next, each r_i is rounded up to the nearest number which is a power of 2, i.e., if $2^{k_i-1}<r_i\le 2^{k_i}$, we set $r_i\leftarrow 2^{k_i}$. Then, if two successive machine types i and $i+1$ satisfy $r_i=r_{i+1}$, we delete the machine type i and never use any type-i machine for processing jobs. It is easy to infer that if a type-i machine is ever used in an optimal schedule, we can always replace it by a machine of a higher-indexed type with a cost rate no larger than twice that of a type-i machine. In this way, the assumption that each r_i is a power of 2 only causes us to lose at most a factor of 2 in deriving the approximation (competitive) ratio for any offline (online) algorithm.

Next, we establish a lower bounding scheme for any BSHM instance, which shall be used to analyze the algorithms for BSHM. The lower bounding scheme relaxes the requirement that each job must be processed by a single machine throughout its active interval. Specifically, we focus on seeking a minimum-cost machine configuration for each time point $t\in\bigcup_{J\in\mathcal{J}} I(J)$. Let $\mathcal{J}(t)=\{J\in\mathcal{J}: t\in I(J)\}$ denote all the jobs active at time t and $\mathcal{J}_{\ge i}(t)=\{J\in\mathcal{J}(t): s(J)>g_{i-1}\}$ denote those that must be placed onto machines of type at least i. Let $w(i,t)$ denote the number of type-i machines used at time t. In any feasible BSHM solution for scheduling jobs $\mathcal{J}$, $\sum_{j=i}^{m} w(j,t)\cdot g_j \ge s(\mathcal{J}_{\ge i}(t),t)$ must hold for each $i\in\{1,\ldots,m\}$. Thus, we aim to construct a machine configuration at time t satisfying these constraints. The target is to minimize the total cost rate $\sum_{i=1}^{m} w(i,t)\cdot r_i$ of all the machines used. We refer to the machine configuration with minimum cost rate as the *optimal machine configuration*. By using $w^*(i,t)$ ($i\in\{1,\ldots,m\}$) to denote the optimal machine configuration at time t, we can easily establish that:

$$\mathrm{OPT}_{\mathrm{BSHM}}(\mathcal{J}) \ge \int_{\bigcup_{J\in\mathcal{J}} I(J)} \Big(\sum_{i=1}^{m} w^*(i,t)\cdot r_i\Big)\,\mathrm{d}t, \quad (1)$$

where $\mathrm{OPT}_{\mathrm{BSHM}}(\mathcal{J})$ is the optimal (minimum) cost for an BSHM instance $\mathcal{J}$.

III. BSHM-DEC

In this section, we study a particular case BSHM-DEC in which $\frac{r_i}{g_i}\ge\frac{r_{i+1}}{g_{i+1}}$ holds for each $i\in\{1,\ldots,m-1\}$. In this case, it is generally desirable to assign all the jobs to the highest-indexed machine type m since it has the lowest amortized cost rate. However, if the total size of the active jobs is far smaller than the capacity of a type-m machine, then a lower-indexed machine type may be more cost-effective to use. Thus, job scheduling should strike a balance between the sizes of active jobs and the machine types to use. This is further challenged by the constraint that each job has to run on a single machine throughout its active interval. If a job is assigned to a high-indexed machine type at its arrival, it cannot be reassigned to a low-indexed machine type later even if the total size of the active jobs drops.

[1] Otherwise, if $g_i\le g_{i+1}$ and $r_i\ge r_{i+1}$, then no type-i machine is needed for processing jobs, since a type-i machine can always be replaced by a type-$(i+1)$ machine with no extra cost incurred.

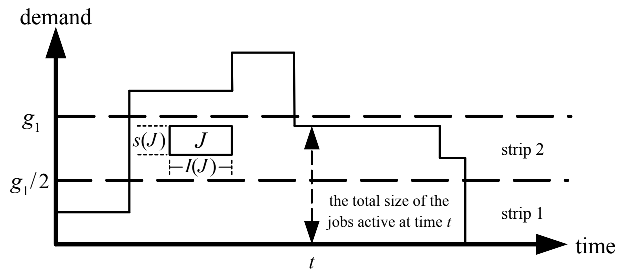

Fig. 1. Job placement in the demand chart

A. *Offline Setting*

In the offline setting, the information of all the jobs is known before the scheduling process. Inspired by the idea of the Dual Coloring algorithm designed for the MinUsageTime DBP problem [13], we propose a $O(1)$-approximation algorithm to place jobs onto machines in an iterative manner.

Our approximation algorithm works as follows. In the first iteration, we select all the jobs $\ddot{\mathcal{J}}_1 = \{J \in \mathcal{J} : s(J) \leq g_1\}$ which can be placed onto type-1 machines. A demand chart is constructed for these jobs, such that the height of the demand chart at any time t is the total size of the jobs active at this moment, i.e., $s(\ddot{\mathcal{J}}_1, t)$ (see Figure 1 for an illustration). Then, we apply the placement phase of the Dual Coloring algorithm to place all the jobs inside the demand chart with each job J represented by a rectangle spanning its active interval $I(J)$ in the time dimension and having a height of its size $s(J)$ in the demand dimension (please refer to [13] for details of the placement algorithm). The placement algorithm ensures that no three jobs overlap together in their placement. Next, the demand chart is partitioned into strips, each of height $\frac{g_1}{2}$. For each of the bottom $2 \cdot (\frac{r_2}{r_1} - 1)$ strips, we assign all the jobs placed fully inside it to a type-1 machine. Since each job placed fully inside a strip must have a size no larger than $\frac{g_1}{2}$ and no three jobs overlap together, the total size of the jobs fully inside a strip must be no larger than g_1 at any time, which makes the job assignment feasible. For the jobs crossing every pair of adjacent strips in the bottom $2 \cdot (\frac{r_2}{r_1} - 1)$ strips, we can use at most two type-1 machines for processing them. This is because each job has a size no more than g_1 and at most two jobs can overlap together at any point on the boundary between the two strips. As a result, at most $2 \cdot (\frac{r_2}{r_1} - 1) + 2 \cdot 2 \cdot (\frac{r_2}{r_1} - 1) = 6 \cdot (\frac{r_2}{r_1} - 1)$ type-1 machines are used for processing jobs at any time in the first iteration. Let $\check{\mathcal{J}}_1$ denote all the jobs assigned to these machines. If no jobs are left to be scheduled (i.e., $\check{\mathcal{J}}_1 = \ddot{\mathcal{J}}_1 = \mathcal{J}$), job scheduling completes. Otherwise, we proceed with the second iteration to schedule jobs onto type-2 machines.

In general, in each iteration $i \in \{1, \ldots, m-1\}$, we consider all the jobs $\ddot{\mathcal{J}}_i = \{J \in \mathcal{J} : s(J) \leq g_i\} - \bigcup_{k=1}^{i-1} \check{\mathcal{J}}_k$ that have sizes no larger than g_i and are not scheduled by any of the previous $i-1$ iterations, where $\check{\mathcal{J}}_k$ denotes all the jobs scheduled in iteration k. A demand chart is constructed for $\ddot{\mathcal{J}}_i$ and all these jobs are placed inside the demand chart by applying the placement algorithm. Then, the demand chart is partitioned into strips of height $\frac{g_i}{2}$ each. Similar to the first iteration, we schedule all the jobs intersecting with the bottom $2 \cdot (\frac{r_{i+1}}{r_i} - 1)$ strips onto at most $6 \cdot (\frac{r_{i+1}}{r_i} - 1)$ type-i machines. Let $\check{\mathcal{J}}_i$ denote all the jobs scheduled in iteration i. If $\bigcup_{k=1}^{i} \check{\mathcal{J}}_k \neq \mathcal{J}$, we proceed with iteration $i+1$. In the final iteration m (if needed), we schedule all the remaining jobs onto type-m machines: after all the jobs $\ddot{\mathcal{J}}_m = \mathcal{J} - \bigcup_{k=1}^{m-1} \check{\mathcal{J}}_k$ are placed inside the demand chart constructed for them, the demand chart is partitioned into strips of height $\frac{g_m}{2}$ each. For each strip, we assign all the jobs fully inside it to a type-m machine. For every two adjacent strips, we use at most two type-m machines for scheduling all the jobs crossing the strips. We refer to the above iterative algorithm as DEC-OFFLINE.

Theorem 1: The DEC-OFFLINE algorithm achieves an approximation ratio of 14 for offline BSHM-DEC.

Proof: By the lower bounding scheme (1), we only need to show that at any time t, the total cost of the machines used by DEC-OFFLINE is bounded by $14 \cdot \sum_{i=1}^{m} w^*(i,t) \cdot r_i$. To prove this fact, we define $k(t)$ as the highest-indexed machine type onto which jobs active at time t are scheduled by DEC-OFFLINE. In other words, all the jobs active at time t are scheduled onto machines of types $1, \ldots, k(t)$.

First, we consider the case that at least one machine of type above $k(t)$ is used in an optimal machine configuration at time t, i.e., there exists an $l \in \{k(t)+1, \ldots, m\}$ with $w^*(l,t) \geq 1$. Note that by applying DEC-OFFLINE, in each iteration $i \in \{1, \ldots, k(t)\}$, at most $6 \cdot (\frac{r_{i+1}}{r_i} - 1)$ type-i machines are used at time t. Consequently, the total cost rate of all the machines used for processing jobs at time t is bounded by

$$6 \cdot \sum_{i=1}^{k(t)} (\frac{r_{i+1}}{r_i} - 1) \cdot r_i = 6 \cdot \sum_{i=1}^{k(t)} (r_{i+1} - r_i) < 6 \cdot r_{k(t)+1},$$

Since $w^*(l,t) \geq 1$, we can thus further bound $6 \cdot r_{k(t)+1}$ by $6 \cdot r_l \leq 6 \cdot w^*(l,t) \cdot r_l < 14 \cdot \sum_{i=1}^{m} w^*(i,t) \cdot r_i$.

Now we consider the case that only machines of types $1, \ldots, k(t)$ are possibly used in an optimal machine configuration at time t, i.e., for each $l \in \{k(t)+1, \ldots, m\}$, $w^*(l,t) = 0$. Since $\frac{r_1}{g_1} \geq \frac{r_2}{g_2} \geq \cdots \geq \frac{r_{k(t)}}{g_{k(t)}}$, the cost rate per resource unit in the optimal machine configuration must be at least $\frac{r_{k(t)}}{g_{k(t)}}$. Thus, to serve all the workloads $s(\mathcal{J}, t)$ at time t, we have

$$\sum_{i=1}^{m} w^*(i,t) \cdot r_i \geq \frac{r_{k(t)}}{g_{k(t)}} \cdot s(\mathcal{J}, t). \tag{2}$$

Define $x = \left\lceil \frac{2 \cdot s(\ddot{\mathcal{J}}_{k(t)}, t)}{g_{k(t)}} \right\rceil$, where $s(\ddot{\mathcal{J}}_{k(t)}, t)$ is the height of the demand chart at time t in iteration $k(t)$ of DEC-OFFLINE. That is, $s(\ddot{\mathcal{J}}_{k(t)}, t)$ satisfies the following inequality:

$$(x-1) \cdot \frac{g_{k(t)}}{2} < s(\ddot{\mathcal{J}}_{k(t)}, t) \leq x \cdot \frac{g_{k(t)}}{2}. \tag{3}$$

Then, the demand chart is divided into x strips at time t. Thus, the number of type-$k(t)$ machines used at time t by DEC-OFFLINE is no more than $3x-2$, i.e., at most one type-$k(t)$ machine for processing the jobs fully inside each strip and at most two type-$k(t)$ machines for processing the jobs crossing every pair of adjacent strips (there are $x-1$ pairs). Besides, in each iteration $i \in \{1, \ldots, k(t)-1\}$, at most $6 \cdot (\frac{r_{i+1}}{r_i} - 1)$ type-i machines are used at time t for processing jobs. Since

$$6 \cdot \sum_{i=1}^{k(t)-1} (\frac{r_{i+1}}{r_i} - 1) \cdot r_i < 6 \cdot r_{k(t)}, \tag{4}$$

the total cost rate of all the machines used at time t is no more than $(3x-2+6) \cdot r_{k(t)} = (3x+4) \cdot r_{k(t)}$. In the following, we prove that $(3x+4) \cdot r_{k(t)} \leq 14 \cdot \sum_{i=1}^{m} w^*(i,t) \cdot r_i$ by considering three different cases.

In the case that $x \geq 3$, it follows from (2) and (3) that

$$\begin{aligned}
\sum_{i=1}^{m} w^*(i,t) \cdot r_i &\geq \frac{r_{k(t)}}{g_{k(t)}} \cdot s(\mathcal{J}, t) \\
&\geq \frac{r_{k(t)}}{g_{k(t)}} \cdot s(\ddot{\mathcal{J}}_{k(t)}, t) \\
&> \frac{r_{k(t)}}{g_{k(t)}} \cdot (x-1) \cdot \frac{g_{k(t)}}{2} \\
&= \frac{x-1}{2} \cdot r_{k(t)}.
\end{aligned}$$

Thus,

$$\begin{aligned}
(3x+4) \cdot r_{k(t)} &= (6 + \frac{14}{x-1}) \cdot \frac{x-1}{2} \cdot r_{k(t)} \\
&< (6 + \frac{14}{x-1}) \cdot \Big(\sum_{i=1}^{m} w^*(i,t) \cdot r_i \Big) \\
&\leq 13 \cdot \sum_{i=1}^{m} w^*(i,t) \cdot r_i.
\end{aligned}$$

In the case that $x = 2$, we need to further examine the optimal machine configuration at time t. If $w^*(k(t), t) \geq 1$, it is straightforward that $(3x+4) \cdot r_{k(t)} = 10 \cdot r_{k(t)} \leq 10 \cdot w^*(k(t), t) \cdot r_{k(t)} < 14 \cdot \sum_{i=1}^{m} w^*(i,t) \cdot r_i$. If $w^*(k(t), t) = 0$, then in the optimal machine configuration at time t, only machines of types $1, \ldots, k(t)-1$ are possibly used. This suggests that each job active at time t has a size no more than $g_{k(t)-1}$. Consequently, by the definition of DEC-OFFLINE, $\ddot{\mathcal{J}}_{k(t)} \subseteq \ddot{\mathcal{J}}_{k(t)-1}$. Besides, in iteration $k(t)-1$, each job $J \in \ddot{\mathcal{J}}_{k(t)}$ must be placed completely above the altitude $(\frac{r_{k(t)}}{r_{k(t)-1}} - 1) \cdot g_{k(t)-1}$ in the demand chart. Otherwise, the rectangle representing job J must overlap with at least one of the first $2 \cdot (\frac{r_{k(t)}}{r_{k(t)-1}} - 1)$ strips and thus J must have been scheduled in iteration $k(t)-1$, which leads to a contradiction. Since all the jobs in $\ddot{\mathcal{J}}_{k(t)}$ are placed above the altitude $(\frac{r_{k(t)}}{r_{k(t)-1}} - 1) \cdot g_{k(t)-1}$ and no three jobs overlap together in their placement, we have

$$s(\ddot{\mathcal{J}}_{k(t)}, t) \leq 2 \cdot \left(s(\ddot{\mathcal{J}}_{k(t)-1}, t) - (\frac{r_{k(t)}}{r_{k(t)-1}} - 1) \cdot g_{k(t)-1} \right)$$

and thus

$$\begin{aligned}
s(\ddot{\mathcal{J}}_{k(t)-1}, t) &\geq (\frac{r_{k(t)}}{r_{k(t)-1}} - 1) \cdot g_{k(t)-1} + \frac{1}{2} \cdot s(\ddot{\mathcal{J}}_{k(t)}, t) \\
&\geq \frac{1}{2} \cdot \frac{r_{k(t)}}{r_{k(t)-1}} \cdot g_{k(t)-1} + \frac{1}{2} \cdot s(\ddot{\mathcal{J}}_{k(t)}, t),
\end{aligned}$$

where the last inequality is due to $\frac{r_{k(t)}}{r_{k(t)-1}} \geq 2$. It follows from (3) and $x = 2$ that $s(\ddot{\mathcal{J}}_{k(t)}, t) > \frac{g_{k(t)}}{2}$. Therefore,

$$\begin{aligned}
s(\ddot{\mathcal{J}}_{k(t)-1}, t) &\geq \frac{1}{2} \cdot \frac{r_{k(t)}}{r_{k(t)-1}} \cdot g_{k(t)-1} + \frac{1}{2} \cdot \frac{g_{k(t)}}{2} \\
&\geq \frac{1}{2} \cdot \frac{r_{k(t)}}{r_{k(t)-1}} \cdot g_{k(t)-1} + \frac{1}{4} \cdot \frac{r_{k(t)}}{r_{k(t)-1}} \cdot g_{k(t)-1} \\
&= \frac{3}{4} \cdot \frac{r_{k(t)}}{r_{k(t)-1}} \cdot g_{k(t)-1}.
\end{aligned}$$

On one hand, since only machines of types $1, \ldots, k(t)-1$ are possibly used in the optimal machine configuration at time t, we have

$$\begin{aligned}
\sum_{i=1}^{m} w^*(i,t) \cdot r_i &\geq \frac{r_{k(t)-1}}{g_{k(t)-1}} \cdot s(\mathcal{J}, t) \\
&\geq \frac{r_{k(t)-1}}{g_{k(t)-1}} \cdot s(\ddot{\mathcal{J}}_{k(t)-1}, t) \\
&\geq \frac{r_{k(t)-1}}{g_{k(t)-1}} \cdot \frac{3}{4} \cdot \frac{r_{k(t)}}{r_{k(t)-1}} \cdot g_{k(t)-1} \\
&= \frac{3}{4} \cdot r_{k(t)}.
\end{aligned}$$

On the other hand, given $x = 2$, by applying DEC-OFFLINE, at most 3 type-$k(t)$ machines are used at time t in iteration $k(t)$: for each of the 2 strips, we assign all the jobs fully inside the strip onto a type-$k(t)$ machine; for all the jobs crossing the strips, at most one type-$k(t)$ machine is needed to schedule them, since each job has a size no more than $g_{k(t)-1} \leq \frac{r_{k(t)-1}}{r_{k(t)}} \cdot g_{k(t)} \leq \frac{1}{2} \cdot g_{k(t)}$. Further taking the machines of types below $k(t)$ into account (i.e., (4)), we can thus bound the total cost rate of all the machines used at time t by $6 \cdot r_{k(t)} + 3 \cdot r_{k(t)} = 9 \cdot r_{k(t)} = 12 \cdot \frac{3}{4} \cdot r_{k(t)} \leq 12 \cdot \sum_{i=1}^{m} w^*(i,t) \cdot r_i < 14 \cdot \sum_{i=1}^{m} w^*(i,t) \cdot r_i$.

In the case that $x = 1$, if $w^*(k(t), t) \geq 1$, we can easily have $(3x+4) \cdot r_{k(t)} = 7 \cdot r_{k(t)} \leq 7 \cdot w^*(k(t), t) \cdot r_{k(t)} < 14 \cdot \sum_{i=1}^{m} w^*(i,t) \cdot r_i$. If $w^*(k(t), t) = 0$, then in the optimal machine configuration at time t, only machines of types $1, \ldots, k(t)-1$ are possibly used. Thus, we have

$$\sum_{i=1}^{m} w^*(i,t) \cdot r_i \geq \frac{r_{k(t)-1}}{g_{k(t)-1}} \cdot s(\mathcal{J}, t).$$

By the definition of DEC-OFFLINE, in iteration $k(t)-1$, $2 \cdot (\frac{r_{k(t)}}{r_{k(t)-1}} - 1)$ strips cannot fully cover the demand chart created for $\ddot{\mathcal{J}}_{k(t)-1}$ at time t. This suggests that

$$\begin{aligned}
s(\ddot{\mathcal{J}}_{k(t)-1}, t) &> 2 \cdot (\frac{r_{k(t)}}{r_{k(t)-1}} - 1) \cdot \frac{g_{k(t)-1}}{2} \\
&= (\frac{r_{k(t)}}{r_{k(t)-1}} - 1) \cdot g_{k(t)-1} \\
&\geq \frac{1}{2} \cdot \frac{r_{k(t)}}{r_{k(t)-1}} \cdot g_{k(t)-1}.
\end{aligned}$$

Therefore,

$$\sum_{i=1}^{m} w^*(i,t) \cdot r_i \geq \frac{r_{k(t)-1}}{g_{k(t)-1}} \cdot s(\mathcal{J}, t)$$

$$
\begin{aligned}
&\geq \frac{r_{k(t)-1}}{g_{k(t)-1}} \cdot s(\ddot{\mathcal{J}}_{k(t)-1}, t) \\
&\geq \frac{r_{k(t)-1}}{g_{k(t)-1}} \cdot \frac{1}{2} \cdot \frac{r_{k(t)}}{r_{k(t)-1}} \cdot g_{k(t)-1} \\
&= \frac{1}{2} \cdot r_{k(t)}.
\end{aligned}
$$

As a result, $(3x+4) \cdot r_{k(t)} = 7 \cdot r_{k(t)} \leq 14 \cdot \sum_{i=1}^{m} w^*(i,t) \cdot r_i$. In summary, we have proved that at any time t, the total cost rate of all the machines used by DEC-OFFLINE is bounded by $14 \cdot \sum_{i=1}^{m} w^*(i,t) \cdot r_i$. Hence, the theorem is proven. □

B. Non-Clairvoyant Online Setting

In the non-clairvoyant online setting, each job must be scheduled onto a machine immediately when it arrives, without any information of the jobs arriving in the future. Besides, the departure time of a job is not known at its arrival and thus cannot be used for the scheduling purpose. Based on the First-Fit rule, we propose a $O(\mu)$-competitive online algorithm, where μ is the max/min job duration ratio among all the jobs.

Our online algorithm uses two groups of machines: a group A and a group B. In each group, at most $4 \cdot (\frac{r_{i+1}}{r_i} - 1)$ type-i ($i \in \{1, \ldots, m-1\}$) machines are allowed to be used concurrently at any time. However, the number of type-m machines used is not limited. In each group, all the machines of a given type $i \in \{1, \ldots, m\}$ are indexed. For group A, each type-i machine can only be used to accommodate jobs of size no larger than $\frac{g_i}{2}$; for group B, each type-i machine can only be used to accommodate jobs of size larger than $\frac{g_i}{2}$. Consequently, each type-i machine in group B can accommodate at most one job at any time.

The online algorithm schedules each job J onto a machine as follows. Note that $\frac{r_{i-1}}{g_{i-1}} \geq \frac{r_i}{g_i}$ and $\frac{r_i}{r_{i-1}} \geq 2$ indicate $g_{i-1} \leq \frac{g_i}{2} \leq g_i$. In the situation that J has a size $s(J) \in (\frac{g_i}{2}, g_i]$ for a particular machine type i, the algorithm checks whether there exists an empty type-i machine in group B at time $I(J)^-$. If so, J is placed onto the lowest-indexed empty type-i machine in group B. If not, J is placed onto a machine in group A according to the First-Fit rule. That is, if at least one type-$(i+1)$ machine in group A can accommodate job J at time $I(J)^-$, then J is placed onto the lowest-indexed type-$(i+1)$ machine that can accommodate it. Otherwise, the algorithm checks whether J can be placed onto a type-$(i+2)$ machine in group A and so on. In the situation that J has a size $s(J) \in (g_{i-1}, \frac{g_i}{2}]$ for a particular machine type i, J is directly placed onto a machine in group A according to the First-Fit rule. We refer to the above online algorithm as DEC-ONLINE.

Theorem 2: The DEC-ONLINE algorithm is $O(\mu)$-competitive for non-clairvoyant BSHM-DEC, where μ is the max/min job duration ratio among all the jobs to schedule.

The main idea to prove Theorem 2 is as follows. Based on the given set of jobs $\mathcal{J}$ and the lower bounding scheme (1), we determine a set of intervals $\mathcal{I}'_{i,j}$ (for each $i \in \{1, \ldots, m\}$ and $j \geq 1$) such that $\sum_{i=1}^{m} \sum_{j \geq 1} \mathrm{len}(\mathcal{I}'_{i,j}) \leq 4 \cdot (\mu + 1) \cdot \mathrm{OPT}_{\mathrm{BSHM}}(\mathcal{J})$. Then, for each job J placed onto a type-i machine M of index $\in \{4j-3, 4j-2, 4j-1, 4j\}$ (no matter whether M is in Group A or Group B), we show that $I(J)$ is fully contained in $\mathcal{I}'_{i,j}$, which implies that the DEC-ONLINE algorithm achieves a competitive ratio of $32 \cdot (\mu + 1)$.

Given any set of jobs $\mathcal{J}$, we first construct a machine configuration $\mathcal{M}(t)$ for each time point $t \in \bigcup_{J \in \mathcal{J}} I(J)$ such that the total cost rate of the machines in $\mathcal{M}(t)$ is bounded by 4 times that of an optimal machine configuration at time t, i.e., $4 \cdot \sum_{i=1}^{m} w^*(i,t) \cdot r_i$.

To construct $\mathcal{M}(t)$ at time t, we define two parameters $p_1(t)$ and $p_2(t)$ for this moment. For the first parameter $p_1(t)$, we select the job J_t of the largest size from all the active jobs at time t, i.e., $s(J_t) = \max_{J \in \mathcal{J}(t)} s(J)$. If J_t has to be placed onto a machine of type at least i, i.e., $s(J_t) \in (g_{i-1}, g_i]$, we define $p_1(t) = i$. For the second parameter $p_2(t)$, we check the total size $s(\mathcal{J}, t)$ of all the active jobs at time t. If $s(\mathcal{J}, t) > (\frac{r_m}{r_{m-1}} - 1) \cdot g_{m-1}$, we define $p_2(t) = m$. Otherwise, we must have $s(\mathcal{J}, t) \in \big((\frac{r_i}{r_{i-1}} - 1) \cdot g_{i-1}, (\frac{r_{i+1}}{r_i} - 1) \cdot g_i\big]$ for a particular machine type $i \leq m-1$. In this case, we define $p_2(t) = i$. Given the two parameters, if $p_1(t) > p_2(t)$, we let $\mathcal{M}(t)$ contain $\frac{r_{i+1}}{r_i} - 1$ type-i machines for each $i \in \{1, \ldots, p_1(t)-1\}$ and contain only one machine of type $p_1(t)$. Otherwise, if $p_1(t) \leq p_2(t)$, we let $\mathcal{M}(t)$ contain $\frac{r_{i+1}}{r_i} - 1$ type-i machines for each $i \in \{1, \ldots, p_2(t)-1\}$ and contain $\lceil \frac{s(\mathcal{J},t)}{g_{p_2(t)}} \rceil$ machine(s) of type $p_2(t)$.

Lemma 1: The total cost rate of the machines in $\mathcal{M}(t)$ is bounded by $4 \cdot \sum_{i=1}^{m} w^*(i,t) \cdot r_i$.

Proof: For the case $p_1(t) > p_2(t)$, the total cost rate of the machines in $\mathcal{M}(t)$ is bounded by

$$
\sum_{i=1}^{p_1(t)-1} (\frac{r_{i+1}}{r_i} - 1) \cdot r_i + r_{p_1(t)} = \sum_{i=1}^{p_1(t)-1} (r_{i+1} - r_i) + r_{p_1(t)} < 2 \cdot r_{p_1(t)}.
$$

Since the job J_t has to be placed onto a machine of type at least $p_1(t)$, we can infer that $\sum_{i=1}^{m} w^*(i,t) \cdot r_i \geq r_{p_1(t)}$. Consequently, the total cost rate of the machines in $\mathcal{M}(t)$ is bounded by $2 \cdot \sum_{i=1}^{m} w^*(i,t) \cdot r_i$.

For the case $p_1(t) \leq p_2(t)$, the total cost rate of the machines in $\mathcal{M}(t)$ is bounded by

$$
\begin{aligned}
&\sum_{i=1}^{p_2(t)-1} (\frac{r_{i+1}}{r_i} - 1) \cdot r_i + \Big\lceil \frac{s(\mathcal{J},t)}{g_{p_2(t)}} \Big\rceil \cdot r_{p_2(t)} \\
&= \sum_{i=1}^{p_2(t)-1} (r_{i+1} - r_i) + \Big\lceil \frac{s(\mathcal{J},t)}{g_{p_2(t)}} \Big\rceil \cdot r_{p_2(t)} \\
&< \Big(\Big\lceil \frac{s(\mathcal{J},t)}{g_{p_2(t)}} \Big\rceil + 1\Big) \cdot r_{p_2(t)}.
\end{aligned}
$$

If $\lceil \frac{s(\mathcal{J},t)}{g_{p_2(t)}} \rceil = 1$, the total cost rate of the machines in $\mathcal{M}(t)$ is bounded by $2 \cdot r_{p_2(t)}$. Besides, $\lceil \frac{s(\mathcal{J},t)}{g_{p_2(t)}} \rceil = 1$ also implies that $s(\mathcal{J}, t) \leq g_{p_2(t)}$, i.e., one type-$p_2(t)$ machine can accommodate all the active jobs at time t. On the other hand, by the definition of $p_2(t)$, we have $s(\mathcal{J}, t) > (\frac{r_{p_2(t)}}{r_{p_2(t)-1}} - 1) \cdot g_{p_2(t)-1}$. This suggests that the optimal machine configuration at time t has a cost rate at least $(\frac{r_{p_2(t)}}{r_{p_2(t)-1}} - 1) \cdot r_{p_2(t)-1} = r_{p_2(t)} - r_{p_2(t)-1}$

if no machine of type $p_2(t)$ or above is used. Therefore, the optimal machine configuration at time t satisfies

$$\sum_{i=1}^{m} w^*(i,t)\cdot r_i > \min\Big\{r_{p_2(t)},\ r_{p_2(t)} - r_{p_2(t)-1}\Big\} \geq \frac{r_{p_2(t)}}{2}.$$

Thus, the total cost rate of the machines in $\mathcal{M}(t)$ is bounded by $4\cdot\sum_{i=1}^{m} w^*(i,t)\cdot r_i$.

Finally, consider the scenario $\lceil\frac{s(\mathcal{J},t)}{g_{p_2(t)}}\rceil \geq 2$. By the definition of $p_2(t)$, we have $s(\mathcal{J},t) \leq (\frac{r_{p_2(t)+1}}{r_{p_2(t)}} - 1)\cdot g_{p_2(t)}$, which implies that no machine of type above $p_2(t)$ should be used in the optimal machine configuration at time t. This suggests

$$\sum_{i=1}^{m} w^*(i,t)\cdot r_i \geq s(\mathcal{J},t)\cdot\frac{r_{p_2(t)}}{g_{p_2(t)}} > \Big(\Big\lceil\frac{s(\mathcal{J},t)}{g_{p_2(t)}}\Big\rceil - 1\Big)\cdot r_{p_2(t)}$$

and thus $\big(\lceil\frac{s(\mathcal{J},t)}{g_{p_2(t)}}\rceil + 1\big)\cdot r_{p_2(t)} \leq 3\cdot\big(\lceil\frac{s(\mathcal{J},t)}{g_{p_2(t)}}\rceil - 1\big)\cdot r_{p_2(t)} < 3\cdot\sum_{i=1}^{m} w^*(i,t)\cdot r_i$. Hence, the lemma is proven. □

Based on the machine configuration $\mathcal{M}(t)$ built for each time point t, we construct a set of intervals $\mathcal{I}_{i,j}$ for each $i \in \{1,\ldots,m\}$ and $j \geq 1$. Specifically, $\mathcal{I}_{i,j}$ includes all the time points t when $\mathcal{M}(t)$ contains at least j type-i machines. Note that $\mathcal{I}_{i,j}$ may consist of one or several contiguous intervals that are pairwise disjoint. Let $\mathrm{len}(\mathcal{I}_{i,j})$ denote the total length of the intervals in $\mathcal{I}_{i,j}$. By Lemma 1, we have

$$\begin{aligned}\sum_{i=1}^{m}\sum_{j\geq 1}\big(\mathrm{len}(\mathcal{I}_{i,j})\cdot r_i\big) &\leq 4\cdot\int_{\bigcup_{J\in\mathcal{J}} I(J)}\Big(\sum_{i=1}^{m} w^*(i,t)\cdot r_i\Big)\mathrm{d}t\\ &\leq 4\cdot\mathrm{OPT}_{\mathrm{BSHM}}(\mathcal{J}).\end{aligned}$$

Next, we define $\mathcal{I}'_{i,j} = \bigcup_{I\in\mathcal{I}_{i,j}}\big[I^-, I^+ + \mu\cdot\mathrm{len}(I)\big)$ which extends every contiguous interval in $\mathcal{I}_{i,j}$ by μ times of its own length. Here, μ is the max/min job duration ratio among all the jobs. Obviously, we have $\mathrm{len}(\mathcal{I}'_{i,j}) \leq (\mu+1)\cdot\mathrm{len}(\mathcal{I}_{i,j})$.

We now show that the DEC-ONLINE algorithm produces a schedule with the total cost bounded by

$$\begin{aligned}8\cdot\sum_{i=1}^{m}\sum_{j\geq 1}\big(\mathrm{len}(\mathcal{I}'_{i,j})\cdot r_i\big) &\leq 8\cdot(\mu+1)\cdot\sum_{i=1}^{m}\sum_{j\geq 1}\big(\mathrm{len}(\mathcal{I}_{i,j})\cdot r_i\big)\\ &\leq 32\cdot(\mu+1)\cdot\mathrm{OPT}_{\mathrm{BSHM}}(\mathcal{J}). \quad (5)\end{aligned}$$

Specifically, for each $i\in\{1,\ldots,m\}$ and $j\geq 1$, we define $\mathcal{M}_{i,j}$ as the set of type-i machines with indexes $\{4j-3, 4j-2, 4j-1, 4j\}$ in Group A and in Group B. Note that $\mathcal{M}_{i,j}$ contains at most 8 machines. Let $\mathcal{J}_{i,j}$ denote all the jobs placed onto the machines in $\mathcal{M}_{i,j}$. Then, the total cost of the machines in $\mathcal{M}_{i,j}$ must be bounded by $8\cdot\mathrm{len}\big(\bigcup_{J\in\mathcal{J}_{i,j}} I(J)\big)\cdot r_i$. Consequently, if we can show that each job $J \in \mathcal{J}_{i,j}$ has its active interval $I(J)$ fully contained in $\mathcal{I}'_{i,j}$, then we can bound $\mathrm{len}\big(\bigcup_{J\in\mathcal{J}_{i,j}} I(J)\big)$ by $\mathrm{len}(\mathcal{I}'_{i,j})$. Finally, by (5), we have

$$\begin{aligned}\sum_{i=1}^{m}\sum_{j\geq 1}\Big(8\cdot\mathrm{len}\big(\bigcup_{J\in\mathcal{J}_{i,j}} I(J)\big)\cdot r_i\Big) &\leq 8\cdot\sum_{i=1}^{m}\sum_{j\geq 1}\big(\mathrm{len}(\mathcal{I}'_{i,j})\cdot r_i\big)\\ &\leq 32\cdot(\mu+1)\cdot\mathrm{OPT}_{\mathrm{BSHM}}(\mathcal{J}),\end{aligned}$$

which proves that DEC-ONLINE is $O(\mu)$-competitive. In fact, whether $I(J)$ is fully contained in $\mathcal{I}'_{i,j}$ can be checked by examining $s(\mathcal{J}, I(J)^-)$, i.e., the total size of all the active jobs at time $I(J)^-$.

Lemma 2: Given any job $J\in\mathcal{J}_{i,j}$,

(a) for each $j\geq 2$, if $s(\mathcal{J}, I(J)^-) > 2\cdot(j-1)\cdot g_i$, then $I(J)$ is fully contained in $\mathcal{I}'_{i,j}$;
(b) if $s(\mathcal{J}, I(J)^-) > 2\cdot(\frac{r_i}{r_{i-1}} - 1)\cdot g_{i-1}$, then $I(J)$ is fully contained in $\mathcal{I}'_{i,1}$.

Proof: We first prove (a). Since $s(\mathcal{J}, I(J)^-) > 2\cdot(j-1)\cdot g_i$, by definition, $\mathcal{M}(I(J)^-)$ contains at least j type-i machines, so $\mathcal{I}_{i,j}$ includes the time point $I(J)^-$. Let $I\in\mathcal{I}_{i,j}$ denote the contiguous interval containing the time point $I(J)^-$. We show that $\mathrm{len}(I) \geq \delta$, where δ denotes the minimum job duration, i.e., $\delta = \min_{J\in\mathcal{J}}\mathrm{len}(I(J))$. Specifically, we select all the jobs active at time $I(J)^-$ (i.e., $\mathcal{J}(I(J)^-)$) and apply the placement algorithm used in DEC-OFFLINE to place all these jobs inside the demand chart constructed for them.

Suppose each job $J \in \mathcal{J}(I(J)^-)$ is placed at an altitude bounded by $(j-1)\cdot g_i$. Recall that by applying the placement algorithm, no three jobs overlap together in their placement. Thus, the total size of the jobs $\mathcal{J}(I(J)^-)$, i.e., $s(\mathcal{J}(I(J)^-), I(J)^-)$, is bounded by $2\cdot(j-1)\cdot g_i$. However, by the hypothesis of (a), we have $s(\mathcal{J}(I(J)^-), I(J)^-) = s(\mathcal{J}, I(J)^-) > 2\cdot(j-1)\cdot g_i$, which leads to a contradiction. Therefore, there must exist a job $J'\in\mathcal{J}(I(J)^-)$ placed at an altitude higher than $(j-1)\cdot g_i$. Then at any time $t \in I(J')$, the height of the demand chart is greater than $(j-1)\cdot g_i$, which implies that $s(\mathcal{J},t) \geq s(\mathcal{J}(I(J)^-),t) > (j-1)\cdot g_i$ so that $\mathcal{M}(t)$ contains at least j type-i machines. This suggests that $\mathcal{I}_{i,j}$ must include the entire interval $I(J')$. Since $\mathrm{len}(I(J')) \geq \delta$ and $I(J')$ includes the time point $I(J)^-$, the contiguous interval $I\in\mathcal{I}_{i,j}$ has a length at least δ.

Since $I(J)^+ = I(J)^- + \mathrm{len}(I(J)) \leq I(J)^- + \mu\cdot\delta$, it follows that $I(J) \subseteq [I^-, I^+ + \mu\cdot\delta) \subseteq [I^-, I^+ + \mu\cdot\mathrm{len}(I))$. Therefore, by the definition of $\mathcal{I}'_{i,j}$, $I(J)$ is fully contained in $\mathcal{I}'_{i,j}$.

Similarly, we can prove (b). Since $s(\mathcal{J}, I(J)^-) > 2\cdot(\frac{r_i}{r_{i-1}} - 1)\cdot g_{i-1} > (\frac{r_i}{r_{i-1}} - 1)\cdot g_{i-1}$, by definition, $p_2(I(J)^-) \geq i$ and $\mathcal{M}(I(J)^-)$ contains at least one type-i machine, so $\mathcal{I}_{i,1}$ includes the time point $I(J)^-$. Let $I \in \mathcal{I}_{i,1}$ denote the contiguous interval containing $I(J)^-$. By similar arguments as above, we can show that $\mathrm{len}(I) \geq \delta$ and thus $I(J)$ is fully contained in $\mathcal{I}'_{i,1}$. □

Finally, we prove that each job $J\in\mathcal{J}_{i,j}$ has its active interval $I(J)$ fully contained in $\mathcal{I}'_{i,j}$.

Lemma 3: For each job $J\in\mathcal{J}_{i,j}$, $I(J)$ is fully contained in $\mathcal{I}'_{i,j}$.

Proof: First, consider the case that J is placed onto a type-i machine in Group B, which suggests that J has a size $s(J) \in (\frac{g_i}{2}, g_i]$. By the definition of DEC-ONLINE, we know that at least $4\cdot(j-1)$ type-i machines in Group B are running jobs at time $I(J)^-$ and these jobs all have sizes larger than $\frac{g_i}{2}$. If $j = 1$, $I_{i,1}$ must include the entire interval $I(J)$, since it follows from $s(J) > \frac{g_i}{2} \geq g_{i-1}$ that for any $t\in I(J)$, $p_1(t) \geq i$.

Consequently, $I(J)$ is fully contained in $\mathcal{I}'_{i,1}$. If $j \geq 2$, we have $s(\mathcal{J}, I(J)^-) > 4\cdot(j-1)\cdot\frac{g_i}{2} = 2\cdot(j-1)\cdot g_i$. By Lemma 2, $I(J)$ is fully contained in $\mathcal{I}'_{i,j}$.

Next, consider the case that J is placed onto a type-i machine in Group A. This suggests that J has a size no larger than $\frac{g_i}{2}$ and thus at least $4\cdot(j-1)$ type-i machines in Group A are half full at time $I(J)^-$. If $j \geq 2$, we can easily infer that $s(\mathcal{J}, I(J)^-) \geq 4\cdot(j-1)\cdot\frac{g_i}{2} + s(J) > 2\cdot(j-1)\cdot g_i$. By Lemma 2, J has its active interval $I(J)$ fully contained in $\mathcal{I}'_{i,j}$.

Now we assume $j = 1$. If $s(J) > g_{i-1}$ such that J must be placed onto a machine of type at least i, we can infer that for any $t \in I(J)$, $p_1(t) \geq i$. Thus, $I(J)$ is fully contained in $\mathcal{I}_{i,1}$ and hence $\mathcal{I}'_{i,1}$.

If $s(J) \in (\frac{g_{i-1}}{2}, g_{i-1}]$ such that job J can be placed onto a type-$(i-1)$ machine in Group B, by the definition of DEC-ONLINE, all the type-$(i-1)$ machines in Group B must be running jobs at time $I(J)^-$. Note that these jobs all have sizes larger than $\frac{g_{i-1}}{2}$. This suggests that $s(\mathcal{J}, I(J)^-) > 4\cdot(\frac{r_i}{r_{i-1}} - 1)\cdot\frac{g_{i-1}}{2} = 2\cdot(\frac{r_i}{r_{i-1}} - 1)\cdot g_{i-1}$. By Lemma 2, $I(J)$ is fully contained in $\mathcal{I}'_{i,1}$.

Finally, if $s(J) \leq \frac{g_{i-1}}{2}$ such that job J can be placed onto a type-$(i-1)$ machine in Group A, then all the type-$(i-1)$ machines in Group A must be at least half full at time $I(J)^-$. Consequently, $s(\mathcal{J}, I(J)^-) \geq 4\cdot(\frac{r_i}{r_{i-1}} - 1)\cdot\frac{g_{i-1}}{2} + s(J) > 2\cdot(\frac{r_i}{r_{i-1}} - 1)\cdot g_{i-1}$. Again, the claim follows from Lemma 2. □

Recall that the MinUsageTime DBP problem is a special case of BSHM-DEC and no deterministic online algorithm can achieve a competitive ratio less than μ for MinUsageTime DBP in the non-clairvoyant online setting [11]. Thus, Theorem 2 implies that DEC-ONLINE achieves an asymptotically tight competitive ratio for non-clairvoyant BSHM-DEC.

IV. BSHM-INC

Now, we consider another particular case BSHM-INC in which $\frac{r_i}{g_i} \leq \frac{r_{i+1}}{g_{i+1}}$ holds for each $i \in \{1, \ldots, m-1\}$. In this case, it is preferable to assign each job to the lowest-indexed machine type fitting the job size to take advantage of its lowest amortized cost rate, unless the job can be scheduled onto a machine of a higher-indexed type "for free" — that is, the job can be accommodated by the residual capacity of the machine after hosting the jobs that must run on it. We show that partitioning a given set of jobs $\mathcal{J}$ into m disjoint subsets $\mathcal{J}_i = \{J \in \mathcal{J} : s(J) \in (g_{i-1}, g_i]\}$ $(i \in \{1, \ldots, m\})$ and scheduling the jobs in each subset separately form a decent strategy. At any time, the cost rate of such a partitioning strategy loses at most a small constant factor compared to the optimal machine configuration. For the offline setting, the Dual Coloring algorithm [13] can then be applied on each subset $\mathcal{J}_i$ to achieve a $O(1)$-approximation ratio. We refer to this algorithm as INC-OFFLINE. For the non-clairvoyant online setting, the First Fit algorithm can be applied on each subset $\mathcal{J}_i$ to achieve a competitive ratio of $O(\mu)$, where μ is the max/min job duration ratio. We refer to this algorithm as INC-ONLINE. We start by analyzing the cost rate of the partitioning strategy.

Lemma 4: At any time t, $\sum_{i=1}^{m} \lceil \frac{s(\mathcal{J}_i, t)}{g_i} \rceil \cdot r_i \leq \frac{9}{4} \cdot \sum_{i=1}^{m} w^*(i,t) \cdot r_i$.

Proof: Given an optimal machine configuration $w^*(i,t)$ $(i \in \{1, \ldots, m\})$ at time t, we convert it to another configuration, in which at least $\lceil \frac{s(\mathcal{J}_i, t)}{g_i} \rceil$ type-i machines are used at time t. Then, we show that the new machine configuration incurs a total cost rate no more than $\frac{9}{4} \cdot \sum_{i=1}^{m} w^*(i,t) \cdot r_i$.

Suppose the machine types used by the optimal machine configuration at time t include $k_1, k_2, \ldots, k_q$ $(1 \leq k_q < \cdots < k_1 \leq m)$. For notational convenience, we define $k_{q+1} = 1$. Then, the total cost rate of the optimal machine configuration can be written as

$$\sum_{i=1}^{m} w^*(i,t) \cdot r_i = \sum_{i=1}^{q} w^*(k_i, t) \cdot r_{k_i}.$$

By definition, for each $j \in \{k_1 + 1, \ldots, m\}$, since no type-$j$ machine is used, we have $\mathcal{J}_{\geq j}(t) = \emptyset$. Besides, for each machine type k_i, at most one type-k_i machine can be used to serve the workloads due to jobs $\mathcal{J}_{k_{i+1}}, \mathcal{J}_{k_{i+1}+1}, \ldots, \mathcal{J}_{k_i - 1}$. In other words, the last type-k_i machine must be used to serve all the workloads $\sum_{j=k_{i+1}+1}^{k_i - 1} s(\mathcal{J}_j, t)$ and maybe part of the workloads $s(\mathcal{J}_{k_{i+1}}, t)$ as well. Otherwise, the last type-k_i machine can be replaced by $\frac{r_{k_i}}{r_{k_i - 1}}$ type-$(k_i - 1)$ machines with possibly larger capacity but no extra cost to form another optimal machine configuration.

We convert the optimal machine configuration $w^*(i,t)$ $(i \in \{1, \ldots, m\})$ as follows. For each machine type k_i, we let the last type-k_i machine serve only the workloads due to jobs $\mathcal{J}_{k_i}$. For each machine type $j \in \{k_{i+1} + 1, \ldots, k_i - 1\}$, we further use $\lceil \frac{s(\mathcal{J}_j, t)}{g_j} \rceil$ type-j machines to serve the workloads $s(\mathcal{J}_j, t)$ due to jobs $\mathcal{J}_j$. In the case that the last type-k_i machine was also used to serve a workload amount of $\widehat{s}_{k_i}$ due to jobs $\mathcal{J}_{k_{i+1}}$, we further use $\lceil \frac{\widehat{s}_{k_i}}{g_{k_{i+1}}} \rceil$ type-k_{i+1} machines to serve this workload amount. After the above conversion, it is easy to see that for each $i \in \{1, \ldots, m\}$, all the workloads $s(\mathcal{J}_i, t)$ due to jobs $\mathcal{J}_i$ are served by type-i machines. As a result, at least $\lceil \frac{s(\mathcal{J}_i, t)}{g_i} \rceil$ type-i machines are used at time t. Next, we prove that the new machine configuration incurs a total cost rate no more than $\frac{9}{4} \cdot \sum_{i=1}^{q} w^*(k_i, t) \cdot r_{k_i}$. To do so, we show that the extra cost incurred due to the above conversion can be bounded by $\frac{5}{4} \cdot \sum_{i=1}^{q} w^*(k_i, t) \cdot r_{k_i}$.

We first show that the extra cost incurred by using the machines of types $k_2, \ldots, k_1 - 2, k_1 - 1$ is bounded by $\frac{5}{4} \cdot w^*(k_1, t) \cdot r_1$.

In the case that $w^*(k_1, t) \geq 2$, the extra cost is bounded by

$$\begin{aligned} &\left(\sum_{j=k_2+1}^{k_1 - 1} \left\lceil \frac{s(\mathcal{J}_j, t)}{g_j} \right\rceil \cdot r_j \right) + \left\lceil \frac{\widehat{s}_{k_1}}{g_{k_2}} \right\rceil \cdot r_{k_2} \\ &\leq \left(\sum_{j=k_2}^{k_1 - 1} r_j \right) + \left(\sum_{j=k_2+1}^{k_1 - 1} \frac{s(\mathcal{J}_j, t)}{g_j} \cdot r_j \right) + \frac{\widehat{s}_{k_1}}{g_{k_2}} \cdot r_{k_2}. \end{aligned}$$

On one hand, since $r_j \le \frac{r_{j+1}}{2}$ holds for each j, it follows that for any $h_1, h_2 \in \{1,\dots,m\}$ where $h_1 \le h_2$,

$$\sum_{j=h_1}^{h_2} r_j \le 2\cdot r_{h_2} - r_{h_1} \le r_{h_2+1} - r_{h_1}. \tag{6}$$

This implies that

$$\sum_{j=k_2}^{k_1-1} r_j \le r_{k_1} - r_{k_2}. \tag{7}$$

On the other hand, since $\frac{r_j}{g_j} \le \frac{r_{j+1}}{g_{j+1}}$ holds for each j, we have

$$\begin{aligned}\left(\sum_{j=k_2+1}^{k_1-1} \frac{s(\mathcal{J}_j,t)}{g_j}\cdot r_j\right) + \frac{\widehat{s}_{k_1}}{g_{k_2}}\cdot r_{k_2} &\le \sum_{j=k_2+1}^{k_1-1} \frac{s(\mathcal{J}_j,t)}{g_{k_1}}\cdot r_{k_1} + \frac{\widehat{s}_{k_1}}{g_{k_1}}\cdot r_{k_1}\\ &= \frac{\sum_{j=k_2+1}^{k_1-1} s(\mathcal{J}_j,t) + \widehat{s}_{k_1}}{g_{k_1}}\cdot r_{k_1}\\ &\le r_{k_1}, \end{aligned} \tag{8}$$

where the last inequality follows from the fact that all the workloads $\sum_{j=k_2+1}^{k_1-1} s(\mathcal{J}_j,t) + \widehat{s}_{k_1}$ can be served by one type-k_1 machine. By adding (7) and (8) together, we can bound the extra cost by $r_{k_1} - r_{k_2} + r_{k_1} \le w^*(k_1,t)\cdot r_{k_1} - r_{k_2} < \frac{5}{4}\cdot w^*(k_1,t)\cdot r_{k_1} - r_{k_2}$.

In the case that $w^*(k_1,t) = 1$ (i.e., only one type-k_1 machine is used in the optimal machine configuration), we can assume $s(\mathcal{J}_{k_1},t) > 0$. Otherwise, the only type-k_1 machine can be replaced by $\frac{r_{k_1}}{r_{k_1-1}}$ type-(k_1-1) machines with possibly larger capacity but no extra cost to form another optimal machine configuration. Since $s(\mathcal{J}_{k_1},t) > 0$, there must exist an active job $\widetilde{J}$ at time t satisfying

$$s(\widetilde{J}) > g_{k_1-1}. \tag{9}$$

In this case, if no type-(k_1-1) machine is used after the conversion, then the extra cost can be bounded by

$$\begin{aligned}&\left(\sum_{j=k_2+1}^{k_1-2} \left\lceil \frac{s(\mathcal{J}_j,t)}{g_j} \right\rceil \cdot r_j\right) + \left\lceil \frac{\widehat{s}_{k_1}}{g_{k_2}} \right\rceil \cdot r_{k_2}\\ &\le \left(\sum_{j=k_2}^{k_1-2} r_j\right) + \left(\sum_{j=k_2+1}^{k_1-2} \frac{s(\mathcal{J}_j,t)}{g_j}\cdot r_j\right) + \frac{\widehat{s}_{k_1}}{g_{k_2}}\cdot r_{k_2}\\ &\le (r_{k_1-1} - r_{k_2}) + \left(\sum_{j=k_2+1}^{k_1-2} \frac{s(\mathcal{J}_j,t)}{g_j}\cdot r_j\right) + \frac{\widehat{s}_{k_1}}{g_{k_2}}\cdot r_{k_2},\end{aligned}$$

where the last inequality is due to (6). In the optimal machine configuration, since the workloads served by the only type-k_1 machine are capped by its capacity, we have

$$s(\widetilde{J}) + \left(\sum_{j=k_2+1}^{k_1-2} s(\mathcal{J}_j,t) + \widehat{s}_{k_1}\right) \le g_{k_1}. \tag{10}$$

Therefore, it follows that

$$\begin{aligned}&\left(\sum_{j=k_2+1}^{k_1-2} \frac{s(\mathcal{J}_j,t)}{g_j}\cdot r_j\right) + \frac{\widehat{s}_{k_1}}{g_{k_2}}\cdot r_{k_2}\\ &\le \frac{\sum_{j=k_2+1}^{k_1-2} s(\mathcal{J}_j,t) + \widehat{s}_{k_1}}{g_{k_1}}\cdot r_{k_1}\\ &\le \frac{g_{k_1} - s(\widetilde{J})}{g_{k_1}}\cdot r_{k_1} \qquad \text{(due to (10))}\\ &< \frac{g_{k_1} - g_{k_1-1}}{g_{k_1}}\cdot r_{k_1} \qquad \text{(due to (9))}\\ &\le r_{k_1} - \frac{r_{k_1-1}}{g_{k_1-1}}\cdot g_{k_1-1} = r_{k_1} - r_{k_1-1}.\end{aligned}$$

Consequently, the extra cost can be bounded by $(r_{k_1-1} - r_{k_2}) + (r_{k_1} - r_{k_1-1}) = r_{k_1} - r_{k_2} \le \frac{5}{4}\cdot w^*(k_1,t)\cdot r_{k_1} - r_{k_2}$.

If exactly one type-(k_1-1) machine is used after the conversion, then we can further infer that there must exist an active job $\bar{J}$ at time t satisfying $s(\bar{J}) \in (g_{k_1-2}, g_{k_1-1}]$. Note that in the optimal machine configuration, the only type-k_1 machine used can serve the workload $s(\widetilde{J})$ due to job $\widetilde{J}$, the workload $s(\bar{J})$ due to job $\bar{J}$ and all the workloads $\sum_{j=k_2+1}^{k_1-2} s(\mathcal{J}_j,t) + \widehat{s}_{k_1}$. We can thus conclude that $s(\widetilde{J}) + s(\bar{J}) + \sum_{j=k_2+1}^{k_1-2} s(\mathcal{J}_j,t) + \widehat{s}_{k_1} \le g_{k_1}$. Hence,

$$\begin{aligned}\sum_{j=k_2+1}^{k_1-2} s(\mathcal{J}_j,t) + \widehat{s}_{k_1} &\le g_{k_1} - s(\widetilde{J}) - s(\bar{J})\\ &< g_{k_1} - g_{k_1-1} - g_{k_1-2}.\end{aligned} \tag{11}$$

As a result, the extra cost is bounded by

$$\begin{aligned}& r_{k_1-1} + \left(\sum_{j=k_2+1}^{k_1-2} \left\lceil \frac{s(\mathcal{J}_j,t)}{g_j} \right\rceil \cdot r_j\right) + \left\lceil \frac{\widehat{s}_{k_1}}{g_{k_2}} \right\rceil \cdot r_{k_2}\\ &< r_{k_1-1} + \left(\sum_{j=k_2}^{k_1-2} r_j\right) + \left(\sum_{j=k_2+1}^{k_1-2} \frac{s(\mathcal{J}_j,t)}{g_j}\cdot r_j\right) + \frac{\widehat{s}_{k_1}}{g_{k_2}}\cdot r_{k_2}\\ &\le r_{k_1-1} + (2\cdot r_{k_1-2} - r_{k_2}) + \left(\sum_{j=k_2+1}^{k_1-2} \frac{s(\mathcal{J}_j,t)}{g_j}\cdot r_j\right) + \frac{\widehat{s}_{k_1}}{g_{k_2}}\cdot r_{k_2}\\ &\qquad \text{(due to (6))}\\ &\le r_{k_1-1} + 2\cdot r_{k_1-2} - r_{k_2} + \frac{\sum_{j=k_2+1}^{k_1-2} s(\mathcal{J}_j,t) + \widehat{s}_{k_1}}{g_{k_1}}\cdot r_{k_1}\\ &< r_{k_1-1} + 2\cdot r_{k_1-2} - r_{k_2} + \frac{g_{k_1} - g_{k_1-1} - g_{k_1-2}}{g_{k_1}}\cdot r_{k_1}\\ &\qquad \text{(due to (11))}\\ &< r_{k_1-1} + 2\cdot r_{k_1-2} - r_{k_2} + r_{k_1} - \frac{r_{k_1-1}}{g_{k_1-1}}\cdot g_{k_1-1} - \frac{r_{k_1-2}}{g_{k_1-2}}\cdot g_{k_1-2}\\ &= r_{k_1} + r_{k_1-2} - r_{k_2}\\ &\le \frac{5}{4}\cdot r_{k_1} - r_{k_2} \qquad \text{(since each } r_i \text{ is a power of 2)}\\ &= \frac{5}{4}\cdot w^*(k_1,t)\cdot r_{k_1} - r_{k_2}.\end{aligned}$$

If at least two type-(k_1-1) machines are used after the conversion (i.e., $\lceil \frac{s(\mathcal{J}_{k_1-1},t)}{g_{k_1-1}} \rceil \ge 2$), then we have $s(\mathcal{J}_{k_1-1},t) > g_{k_1-1}$. Since the workload $s(\widetilde{J})$ due to job $\widetilde{J}$ and the workloads $\sum_{j=k_2+1}^{k_1-1} s(\mathcal{J}_j,t) + \widehat{s}_{k_1}$ can all be served by the only type-k_1 machine used in the optimal machine configuration, we have

$$2\cdot g_{k_1-1} < s(\widetilde{J}) + s(\mathcal{J}_{k_1-1},t)$$

$$\leq s(\widetilde{J})+\sum_{j=k_2+1}^{k_1-1} s(\mathcal{J}_j,t)+\widehat{s}_{k_1} \leq g_{k_1}. \tag{12}$$

Recall that $\frac{r_{k_1-1}}{g_{k_1-1}} \leq \frac{r_{k_1}}{g_{k_1}}$. Thus, we have $\frac{r_{k_1-1}}{r_{k_1}} \leq \frac{g_{k_1-1}}{g_{k_1}} < \frac{1}{2}$, which suggests that

$$\frac{r_{k_1-1}}{r_{k_1}} \leq \frac{1}{4}, \tag{13}$$

since each r_i is a power of 2. Therefore, the extra cost can be bounded by

$$\begin{aligned}
&\left(\sum_{j=k_2+1}^{k_1-1}\left\lceil\frac{s(\mathcal{J}_j,t)}{g_j}\right\rceil\cdot r_j\right)+\left\lceil\frac{\widehat{s}_{k_1}}{g_{k_2}}\right\rceil\cdot r_{k_2}\\
&< \left(\sum_{j=k_2}^{k_1-1} r_j\right)+\left(\sum_{j=k_2+1}^{k_1-1}\frac{s(\mathcal{J}_j,t)}{g_j}\cdot r_j\right)+\frac{\widehat{s}_{k_1}}{g_{k_2}}\cdot r_{k_2}\\
&\leq (2\cdot r_{k_1-1}-r_{k_2})+\left(\sum_{j=k_2+1}^{k_1-1}\frac{s(\mathcal{J}_j,t)}{g_j}\cdot r_j\right)+\frac{\widehat{s}_{k_1}}{g_{k_2}}\cdot r_{k_2}\\
&\hfill \text{(due to (6))}\\
&\leq 2\cdot r_{k_1-1}-r_{k_2}+\frac{\sum_{j=k_2+1}^{k_1-1} s(\mathcal{J}_j,t)+\widehat{s}_{k_1}}{g_{k_1}}\cdot r_{k_1}\\
&\leq 2\cdot r_{k_1-1}-r_{k_2}+\frac{g_{k_1}-s(\widetilde{J})}{g_{k_1}}\cdot r_{k_1} \qquad \text{(due to (12))}\\
&< 2\cdot r_{k_1-1}-r_{k_2}+\frac{g_{k_1}-g_{k_1-1}}{g_{k_1}}\cdot r_{k_1} \qquad \text{(due to (9))}\\
&< 2\cdot r_{k_1-1}-r_{k_2}+r_{k_1}-\frac{r_{k_1-1}}{g_{k_1-1}}\cdot g_{k_1-1}\\
&= r_{k_1}+r_{k_1-1}-r_{k_2}\\
&\leq \frac{5}{4}\cdot r_{k_1}-r_{k_2} = \frac{5}{4}\cdot w^*(k_1,t)\cdot r_{k_1}-r_{k_2}. \qquad \text{(due to (13))}
\end{aligned}$$

In summary, in any case, the extra cost incurred by using the machines of types $k_2, \ldots, k_1-2, k_1-1$ in the conversion is bounded by $\frac{5}{4}\cdot w^*(k_1,t)\cdot r_{k_1}-r_{k_2}$.

By similar arguments, for each k_i $(i \geq 2)$, the extra cost incurred by using the machines of types $k_{i+1}, \ldots, k_i-2, k_i-1$ in the conversion can be bounded by

$$\begin{aligned}
&\left(\sum_{j=k_{i+1}+1}^{k_i-1}\left\lceil\frac{s(\mathcal{J}_j,t)}{g_j}\right\rceil\cdot r_j\right)+\left\lceil\frac{\widehat{s}_{k_i}}{g_{k_{i+1}}}\right\rceil\cdot r_{k_{i+1}}\\
&\leq \left(\sum_{j=k_{i+1}}^{k_i-1} r_j\right)+\left(\sum_{j=k_{i+1}+1}^{k_i-1}\frac{s(\mathcal{J}_j,t)}{g_j}\cdot r_j\right)+\frac{\widehat{s}_{k_i}}{g_{k_{i+1}}}\cdot r_{k_{i+1}}\\
&\leq (r_{k_i}-r_{k_{i+1}})+\sum_{j=k_{i+1}+1}^{k_i-1}\frac{s(\mathcal{J}_j,t)}{g_j}\cdot r_j+\frac{\widehat{s}_{k_i}}{g_{k_{i+1}}}\cdot r_{k_{i+1}}\\
&\hfill \text{(due to (6))}\\
&\leq (r_{k_i}-r_{k_{i+1}})+\frac{\sum_{j=k_{i+1}+1}^{k_i-1} s(\mathcal{J}_j,t)+\widehat{s}_{k_i}}{g_{k_i}}\cdot r_{k_i}\\
&\leq (r_{k_i}-r_{k_{i+1}})+\frac{g_{k_i}}{g_{k_i}}\cdot r_{k_i} = 2\cdot r_{k_i}-r_{i+1},
\end{aligned}$$

where the last inequality is due to the fact that all the workloads $\sum_{j=k_{i+1}+1}^{k_i-1} s(\mathcal{J}_j,t)+\widehat{s}_{k_i}$ can be served by the last type-k_i machine in the optimal machine configuration. Consequently, the total extra cost incurred due to the conversion is bounded by

$$\begin{aligned}
&\frac{5}{4}\cdot w^*(k_1,t)\cdot r_{k_1}-r_{k_2}+\sum_{i=2}^{q}\left(2\cdot r_{k_i}-r_{k_{i+1}}\right)\\
&< \frac{5}{4}\cdot w^*(k_1,t)\cdot r_{k_1}+\sum_{i=2}^{q} r_{k_i} \leq \frac{5}{4}\cdot\sum_{i=1}^{q} w^*(k_i,t)\cdot r_{k_i}.
\end{aligned}$$

Hence, the lemma is proved. □

Our earlier work [13] has shown that by applying the Dual Coloring algorithm to schedule a set of jobs $\mathcal{J}$ on homogeneous machines of capacity g, the number of machines used at any time t is bounded by $4\cdot\lceil\frac{s(\mathcal{J},t)}{g}\rceil$. This suggests that the total cost rate incurred by INC-OFFLINE at any time t is bounded by $4\cdot\sum_{i=1}^{m}\lceil\frac{s(\mathcal{J}_i,t)}{g_i}\rceil\cdot r_i$. Based on Lemma 4 and the lower bounding scheme (1), we can bound the accumulated cost for scheduling all the jobs by

$$\begin{aligned}
&\int_{\bigcup_{J\in\mathcal{J}} I(J)} 4\cdot\left(\sum_{i=1}^{m}\left\lceil\frac{s(\mathcal{J}_i,t)}{g_i}\right\rceil\cdot r_i\right)\mathrm{d}t\\
&\leq \int_{\bigcup_{J\in\mathcal{J}} I(J)} 9\cdot\left(\sum_{i=1}^{m} w^*(i,t)\cdot r_i\right)\mathrm{d}t \leq 9\cdot\mathrm{OPT}_{\mathrm{BSHM}}(\mathcal{J}).
\end{aligned}$$

Thus, INC-OFFLINE achieves an approximation ratio of 9 for offline BSHM-INC.

Our earlier work [14] has shown that by applying the First Fit rule (see Section III-B) to schedule a set of jobs $\mathcal{J}$ on homogeneous machines of capacity g, the total machine usage time is bounded by

$$\int_{\bigcup_{J\in\mathcal{J}} I(J)}\left((\mu+2)\cdot\frac{s(\mathcal{J},t)}{g}+1\right)\mathrm{d}t,$$

where μ is the max/min job duration ratio. This suggests that by applying INC-ONLINE, the accumulated cost for scheduling all the jobs is bounded by

$$\begin{aligned}
&\sum_{i=1}^{m} r_i\cdot\int_{\bigcup_{J\in\mathcal{J}_i} I(J)}\left((\mu+2)\cdot\frac{s(\mathcal{J}_i,t)}{g_i}+1\right)\mathrm{d}t\\
&\leq \sum_{i=1}^{m} r_i\cdot\int_{\bigcup_{J\in\mathcal{J}_i} I(J)}\left((\mu+3)\cdot\left\lceil\frac{s(\mathcal{J}_i,t)}{g_i}\right\rceil\right)\mathrm{d}t\\
&= (\mu+3)\cdot\int_{\bigcup_{J\in\mathcal{J}} I(J)}\left(\sum_{i=1}^{m}\left\lceil\frac{s(\mathcal{J}_i,t)}{g_i}\right\rceil\cdot r_i\right)\mathrm{d}t\\
&\leq \frac{9}{4}\cdot(\mu+3)\cdot\mathrm{OPT}_{\mathrm{BSHM}}(\mathcal{J}).
\end{aligned}$$

Thus, INC-ONLINE achieves a competitive ratio of $\frac{9}{4}\mu+\frac{27}{4}$ for non-clairvoyant BSHM-INC. Similar to DEC-ONLINE, this $O(\mu)$ competitive ratio is also asymptotically tight.

V. General Case

Based on the scheduling strategies introduced for BSHM-DEC and BSHM-INC, we design the following algorithms to deal with the general case of BSHM.

First, we construct a graph to describe the relationships among the machine types, where each node i is used to represent the machine type i. An edge from node i to node j indicates that among all the machine types above i, type j is the lowest-indexed one satisfying $\frac{r_i}{g_i} \geq \frac{r_j}{g_j}$. In this way, a forest must be constructed for all the machine types and each tree or sub-tree must contain a set of successive machine types. In other words, if a tree contains nodes i and j $(i<j)$, then this tree must contain all the nodes $i+1, \ldots, j-1$. Furthermore, the root node of a tree (or a sub-tree) must have the highest index among all the nodes in the tree. For example, suppose there are 8 machine types as shown in Figure 2. Then, the forest constructed contains 3 trees as shown.

It is easy to infer that the amortized cost rate per resource unit of a machine type is lower than the amotized cost rates of all the machine types in the trees or sub-trees rooted at its sibling nodes of higher indexes. For example, in Figure 2, type 1 has a lower amortized cost rate than types 2, 3, 4; type 5 has a lower amortized cost rate than types 6, 7, 8. Thus, enlightened by the partitioning strategy for BSHM-INC, we conjecture that for each $j \in \{1, \ldots, m\}$, scheduling jobs of size in $(g_{j-1}, g_j]$ onto only machines of type j or j's parent or ancestor types in the forest would cause us to lose at most a factor of $O(1)$. On the other hand, note that the machine types along the path from any node to the root node of a tree has decreasing amortized cost rates per resource unit. Therefore, we can design iterative scheduling algorithms following the styles of DEC-OFFLINE and DEC-ONLINE. To implement this idea, for each node j, we associate it with a set of jobs $\ddot{\mathcal{J}}_j = \{J \in \mathcal{J} : s(J) \in (g_{i-1}, g_j]\}$, if the tree or subtree rooted at node j includes nodes from i to j ($i=j$ if j is a leaf node).

In the offline setting, the jobs are placed onto machines iteratively by traversing the forest constructed in the post-order. For each machine type j visited, a demand chart is constructed for all the jobs in $\ddot{\mathcal{J}}_j$ that are not yet scheduled. Similar to DEC-OFFLINE, the jobs are first placed in the demand chart. Then, the demand chart is sliced into strips of height $\frac{g_j}{2}$ each. If j is not the root node of a tree in the forest, we schedule all the jobs intersecting with the bottom $\lceil \frac{1}{\sqrt{|C(k)|}} \cdot \frac{r_k}{r_j} \rceil$ strips onto $O(\frac{1}{\sqrt{|C(k)|}} \cdot \frac{r_k}{r_j})$ type-j machines, where k is the parent machine type of j and $|C(k)|$ is the number of k's child machine types. The jobs not assigned to the above machines are passed onto a subsequent iteration when the parent machine type k is visited. If j is the root node of a tree in the forest, we do not limit the number of strips to consider and schedule all the jobs onto type-j machines.

We conjecture that the above algorithm achieves an approximation ratio of $O(\sqrt{m})$. The rationale is that for each machine type k, a child machine type j of k passes jobs onto type k only if the total size of the active jobs in its demand chart exceeds the bottom $\frac{1}{\sqrt{|C(k)|}} \cdot \frac{r_k}{r_j}$ strips. In this case, the cost rate of using only type-j machines exceeds $O\Big(\frac{1}{\sqrt{|C(k)|}} \cdot \frac{r_k}{r_j}\Big) \cdot r_j = O\big(\frac{1}{\sqrt{|C(k)|}}\big) \cdot r_k$, which is $O\big(\frac{1}{\sqrt{|C(k)|}}\big)$ times the cost rate of using only type-k machines. On the other hand, if all the child machine types of k pass jobs onto type k, then the total cost rate of the child-type machines used is bounded by $|C(k)| \cdot O\big(\frac{1}{\sqrt{|C(k)|}}\big) \cdot r_k = O\big(\sqrt{|C(k)|}\big) \cdot r_k$, which is $O\big(\sqrt{|C(k)|}\big)$ times the cost rate of using only type-k machines. Since $|C(k)| < m$, we have $O\big(\sqrt{|C(k)|}\big) = O(\sqrt{m})$.

machine type	cost rate (r_i)	capacity (g_i)	ratio (r_i/g_i)
1	2	1.6	1.25
2	2^2	3	1.333
3	2^3	4	2
4	2^4	12	1.333
5	2^5	32	1
6	2^6	40	1.6
7	2^7	102.4	1.25
8	2^8	128	2

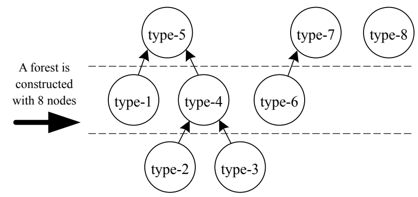

Fig. 2. Constructing a forest for the machine types

Similarly, in the non-clairvoyant online setting, we can design an iterative scheduling algorithm following the style of DEC-ONLINE. We conjecture that the algorithm achieves a competitive ratio of $O(\sqrt{m} \cdot \mu)$.

Acknowledgments

This work is supported by Singapore Ministry of Education Academic Research Fund Tier 1 under Grants 2018-T1-002-063 and 2019-T1-002-042.

References

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

[2] Google cloud pricing. https://cloud.google.com/pricing/.

[3] Microsoft azure pricing. https://azure.microsoft.com/en-us/pricing/.

[4] M. Alicherry and R. Bhatia. Line system design and a generalized coloring problem. In *Proc. ESA*, pages 19–30, 2003.

[5] Y. Azar and D. Vainstein. Tight bounds for clairvoyant dynamic bin packing. In *Proc. ACM SPAA*, pages 77–86, 2017.

[6] J. Chang, S. Khuller, and K. Mukherjee. Lp rounding and combinatorial algorithms for minimizing active and busy time. *Journal of Scheduling*, 20(6):657–680, 2017.

[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. *Theoretical Computer Science*, 411(40-42):3553–3562, 2010.

[8] J. Gergov. Algorithms for compile-time memory optimization. In *Proc. ACM-SIAM SODA*, pages 907–908, 1999.

[9] R. Khandekar, B. Schieber, H. Shachnai, and T. Tamir. Real-time scheduling to minimize machine busy times. *Journal of Scheduling*, 18(6):561–573, 2015.

[10] V. Kumar and A. Rudra. Approximation algorithms for wavelength assignment. In *Proc. FSTTCS*, pages 152–163, 2005.

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

[12] G. B. Mertzios, M. Shalom, A. Voloshin, P. W. Wong, and S. Zaks. Optimizing busy time on parallel machines. *Theoretical Computer Science*, 562:524–541, 2015.

[13] R. Ren and X. Tang. Clairvoyant dynamic bin packing for job scheduling with minimum server usage time. In *Proc. ACM SPAA*, pages 227–237, 2016.

[14] R. Ren, X. Tang, Y. Li, and W. Cai. Competitiveness of dynamic bin packing for online cloud server allocation. *IEEE/ACM Transactions on Networking*, 25(3):1324–1331, 2017.

[15] M. Shalom, A. Voloshin, P. W. Wong, F. C. Yung, and S. Zaks. Online optimization of busy time on parallel machines. *Theoretical Computer Science*, 560:190–206, 2014.

[16] P. Winkler and L. Zhang. Wavelength assignment and generalized interval graph coloring. In *Proc. ACM-SIAM SODA*, pages 830–831, 2003.