



# Semi-online Algorithms for Hierarchical Scheduling on Three Parallel Machines with a Buffer Size of 1

Man Xiao<sup>1</sup>, Lu Ding<sup>1</sup>, Shu Zhao<sup>1</sup>, and Weidong Li<sup>1,2(✉)</sup>

<sup>1</sup> School of Mathematics and Statistics, Yunnan University, Kunming 650504, People's Republic of China

<sup>2</sup> Dianchi College of Yunnan University, Kunming 650000, People's Republic of China

**Abstract.** In this paper, we study the online problem on three hierarchical machines with a buffer size of 1, and have two hierarchy, the objective is to minimize the maximum machine load. When there is only one low-hierarchy machine, we give a low bound  $\frac{3}{2}$  and present an online algorithm with competitive ratio at most  $\frac{5}{3}$ . When there are two low-hierarchy machines, we give a lower bound  $\frac{3}{2}$  and present an online algorithm with competitive ratio at most  $\frac{12}{7}$ .

**Keywords:** Semi-online · Hierarchical machines · Competitive ratio · Buffer

## 1 Introduction

Given  $m$  hierarchical machines and  $n$  independent jobs, each job can be processed on some machines and each job only be processed by a machine. The hierarchical scheduling problem, with the objective is to maximize the minimum machine load [14, 15].

When the objective is to minimize the makespan, denoted by  $P|GoS|C_{max}$ . Hwang et al. [1] investigated the offline problem  $P|GoS|C_{max}$  and proposed a  $\frac{5}{4}$ -competitive algorithm for  $m = 2$ ; also designed a  $2 - \frac{1}{m-1}$ -competitive algorithm for  $m \geq 3$ . Then, Ou et al. [2] designed a  $\frac{4}{3}$ -competitive algorithm and obtain a polynomial time approximation scheme (PTAS, for short) for  $P|GoS|C_{max}$ . For a special case of the problem  $P|GoS|C_{max}$ , Li et al. [3] presented an efficient polynomial time approximation scheme (EPTAS, for short) with running time  $O(n \log n)$ ; for the problem  $Pm|GoS|C_{max}$ , and  $m$  is a constant, they also designed a fully polynomial time approximation scheme (FPTAS, for short) with running time  $O(n)$ .

Online scheduling over list is a scheduling that jobs arrive one by one, any new incoming job will be scheduled only according to the information of arrived jobs. An online algorithm's performance is measured by the competitive ratio. For any given instance  $I$ , let  $C_A(I)$  ( $C_A$ , for short) be the objective value of algorithm  $A$ , for a minimization scheduling problem and a corresponding online algorithm  $A$ , the *competitive ratio* of the online algorithm  $A$  is defined as  $\rho = \sup_I \frac{C_A(I)}{C_{OPT}(I)}$ , i.e.,  $\rho C_{OPT}(I) \geq C_A(I)$  for any instance  $I$ , where  $C_{OPT}(I)$  ( $C_{OPT}$ , for short) is the optimal offline objective value of the instance  $I$ . For the problem, if there is no online algorithm make the competitive

**Table 1.** Buffer size of 0 and 1

	One machine with hierarchy 1		Two machines with hierarchy 1	
	LB	UB	LB	UB
Buffer size of 0	1.801	1.857	1.824	1.857
Buffer size of 1	1.5	1.667	1.5	1.714

ratio less than  $\rho$ , the  $\rho$  is called a *lower bound* of this problem. If an online algorithm has a lower bound equal to its competitive ratio, it is called an optimal online algorithm.

For the online problem, Park et al. [4] and Jiang et al. [5] presented an optimal online algorithm for two hierarchical machines with competitive ratio of  $\frac{5}{3}$  respectively. Lim et al. [6] presented an optimal online algorithm with a competitive ratio of 2 on three hierarchical machines. Wu et al. [7] designed some optimal semi-online scheduling algorithms for two hierarchical machines. Zhang et al. [8] first designed an online algorithm with a competitive ratio of  $1 + \frac{m^2-m}{m^2-km+k^2} < \frac{7}{3}$  for any  $k$  and  $m$ , the  $k$  is the number of high-hierarchy machine, and got a competitive ratio of 1.857 for  $m = 3$ . For jobs have tightly-grouped processing time, Zhang et al. [9] designed some optimal online algorithms on the two hierarchical machines.

For the online problem  $Pm|buffer|C_{max}$ , Englert et al. [10] gave a lower bound of  $\frac{3}{2}$  with a buffer size of  $\lfloor \frac{m}{2} \rfloor - 1$ , Lan et al. [11] gave a 1.5-competitive online algorithm with a buffer size of  $1.5m$ . When  $m = 2$ , Zhang [12] presented an optimal online algorithm with competitive ratio of  $\frac{4}{3}$  and the buffer size of 1. For the online problem  $P2|GoS,buffer|C_{max}$ , Chen et al. [13] designed an optimal online algorithm with competitive ratio of  $\frac{3}{2}$  and the buffer size of 1.

In this paper, we consider the online problem on three hierarchical machines with a buffer size of 1. There are two cases for this problem, when there is one machine of hierarchy 1 and two machines of hierarchy 2, we denote it as  $P3(1,2,2)|buffer|C_{max}$ . When there are two machines of hierarchy 1 and one machine of hierarchy 2, we denote it as  $P3(1,1,2)|buffer|C_{max}$ . Results of lower bound and the competitive ratio of buffer size of 0 and 1 can be found in Table 1. The rest of the paper is organized as follows. In Sect. 2, we define some basic notations. In Sect. 3, we consider the  $P3(1,2,2)|buffer|C_{max}$ , we give a lower bound  $\frac{3}{2}$  and propose an online algorithm of competitive ratio at most  $\frac{5}{3}$ . In Sect. 4, we consider the  $P3(1,1,2)|buffer|C_{max}$ , we give a lower bound  $\frac{3}{2}$  and propose an online algorithm of competitive ratio at most  $\frac{12}{7}$ . Finally, we make a summary.

## 2 Preliminaries

We are given three identical parallel machines  $M_1, M_2, M_3$  and a job set  $J = \{J_1, J_2, \dots, J_n\}$ . All jobs arrive online and all be scheduled irrevocably in their order of arrival. A new job arrives only after the current job is scheduled. We denote the  $j$ -th job in the arrival list as  $J_j = (p_j, g_j)$ ,  $p_j > 0$  is the *processing time* (also called the job's size) of the job  $J_j$ , and  $g_j \in \{1, 2\}$  is the hierarchy of the job  $J_j$ . Each machine  $M_k$  has a

hierarchy  $g(M_k)$  and  $g(M_k) \in \{1, 2\}$ ,  $k \in \{1, 2, 3\}$ . For any job  $J_j$ , only can be processed by the machines which the hierarchy no more than it.  $p_j$  and  $g_j$  are not known until the job  $J_j$  arrives. If  $g_j = i$ , job  $J_j$  is called the job of hierarchy  $i$ ,  $i \in \{1, 2\}$ .

A schedule can be regard as a partition  $(S_1, S_2, S_3)$  of  $J$ , where  $S_k$  ( $k = 1, 2, 3$ ) contains the all jobs assigned to  $M_k$ . Let  $L_k = \sum_{J_j \in S_k} p_j$  be the load of  $M_k$ , and  $T_i$  be the total processing time of the jobs of hierarchy  $i$ ,  $i = 1, 2$ . Our goal is to find a schedule such that  $\max\{L_1, L_2, L_3\}$  is minimized.

For each  $j \in \{1, 2, \dots, n\}$ ,  $k \in \{1, 2, 3\}$  and  $i \in \{1, 2\}$ , we define the following notions.

$T_i^j$ : the total processing time of the first  $j$  jobs of hierarchy  $i$ .

$L_k^j$ : the total processing time of the first  $j$  jobs scheduled on machine  $M_k$  after assigning job  $J_j$ .

$p_{max}^j$ : the maximum processing time of the first  $j$  jobs of hierarchy 1.

$q_{max}^j$ : the maximum processing time of the first  $j$  jobs of hierarchy 2.

$B^j$ : the job in the buffer after scheduling the first  $j$  jobs.

For convince, we also let  $L_k^n = L_k$ ,  $T_i^n = T_i$ ,  $p_{max}^n = p_{max}$  and  $q_{max}^n = q_{max}$ .

### 3 One Machine with Hierarchy 1

In this section, we consider the hierarchy of machine  $M_1$  is 1, the hierarchy of machine  $M_2, M_3$  is 2, and a buffer is available for storing at most one job. When a new job arrives, we have to assign it on a machine irrecoverably, or temporarily store it in the buffer. We give a lower bound  $\frac{3}{2}$  and design an online algorithm with the competitive ratio at most  $\frac{5}{3}$ . In this section, for  $j = 1, 2, \dots, n$ , we let

$$LB^j = \max \left\{ T_1^j, q_{max}^j, \frac{T_1^j + T_2^j}{3} \right\}. \quad (1)$$

**Lemma 1.** For the online problem  $P3(1, 2, 2)|buffer|C_{max}$ , the optimal makespan is at least  $LB^j$  after scheduled the first  $j$  jobs.

**Proof.** Let  $C_{OPT}^j$  be the optimal makespan after scheduled the first  $j$  jobs. For the online problem  $P3(1, 2, 2)|buffer|C_{max}$ , after the job  $J_j$  be scheduled, it is clearly that the  $C_{OPT}^j \geq \max \left\{ \frac{T_1^j + T_2^j}{3}, q_{max}^j \right\}$ . Note that all jobs with hierarchy 1 only be processed on machine  $M_1$ , which implies  $C_{OPT}^j \geq T_1^j$ . So, the lemma holds.

Lemma 1 implies

$$C_{OPT} \geq \max \left\{ T_1, q_{max}, \frac{T_1 + T_2}{3} \right\}. \quad (2)$$

**Theorem 2.** Any online algorithm  $A$  for  $P3(1, 2, 2)|buffer|C_{max}$  has a competitive ratio at least  $\frac{3}{2}$ .

**Proof.** For any online algorithm  $A$ , the first two jobs in the sequence are  $J_1 = (1, 2)$  and  $J_2 = (1, 2)$ , where the first number in brackets is the size and the second number is the hierarchy. We first consider algorithm  $A$  schedules the first two jobs to the machines. If  $A$  assigns both of them to the same machine, the last job  $J_3 = (1, 1)$  arrives, it's implies  $C_A \geq 2$ ,  $C_{OPT} = 1$ . Else, the first two jobs are assigned to different machines, then, the last two jobs  $J_3 = (2, 1)$  and  $J_4 = (2, 2)$  arrive, then  $C_A \geq 3$  and  $C_{OPT} = 2$ .

Next, we consider algorithm  $A$  stores a job in the buffer, without loss of generality, we consider  $A$  stores  $J_1 = (1, 2)$  into the buffer and assigns  $J_2 = (1, 2)$  to someone machine. If  $A$  assigns  $J_2 = (1, 2)$  to  $M_1$ , the last job is  $J_3 = (1, 1)$ , then  $C_A \geq 2$  and  $C_{OPT} = 1$ . If  $A$  not assigns  $J_2 = (1, 2)$  to  $M_1$ , the next job is  $J_3 = (2, 2)$ , and we distinguish three cases for the storage of the buffer.

**Case 1.**  $B^3 = \phi$ , which means that  $A$  assigns first three jobs to the machines.

**Case 1.1.**  $A$  assigns  $J_1 = (1, 2)$  and  $J_3 = (2, 2)$  to the same machine.

The sequence stops, then  $C_A \geq 3$  and  $C_{OPT} = 2$ .

**Case 1.2.**  $A$  doesn't assign  $J_1 = (1, 2)$  and  $J_3 = (2, 2)$  to the same machine.

If  $A$  assigns  $J_3 = (2, 2)$  to  $M_1$ , the last job is  $J_4 = (2, 1)$ , then  $C_A \geq 4$  and  $C_{OPT} = 2$ . If  $A$  assigns  $J_3 = (2, 2)$  to the machine which  $J_2 = (1, 2)$  is assigned in  $M_2$  and  $M_3$ , the sequence stops, then  $C_A \geq 3$  and  $C_{OPT} = 2$ . If  $A$  assigns  $J_3 = (2, 2)$  to the machine that is not assigned  $J_2 = (1, 2)$  in  $M_2$  and  $M_3$ , and assigns  $J_1 = (1, 2)$  to  $M_1$ , the last job is  $J_4 = (2, 1)$ , then  $C_A \geq 3$  and  $C_{OPT} = 2$ . If  $A$  assigns  $J_3 = (2, 2)$  to the machine that is not assigned  $J_2 = (1, 2)$  in  $M_2$  and  $M_3$ , and assigns  $J_1 = (1, 2)$  to the machine which  $J_2 = (1, 2)$  is assigned in  $M_2$  and  $M_3$ , the last two jobs are  $J_4 = (4, 2)$  and  $J_5 = (4, 2)$ , then  $C_A \geq 6$  and  $C_{OPT} = 4$ , implies  $\frac{C_A}{C_{OPT}} \geq \frac{3}{2}$ .

**Case 2.**  $B^3 = J_3 = (2, 2)$ , and  $A$  assigns  $J_1 = (1, 2)$  to someone machine.

**Case 2.1.**  $A$  assigns the  $J_1 = (1, 2)$  to  $M_1$ .

The last job is  $J_4 = (2, 1)$ , then  $C_A \geq 3$  and  $C_{OPT} = 2$ .

**Case 2.2.**  $A$  assigns  $J_1 = (1, 2)$  to the machine that is not assigned  $J_2 = (1, 2)$  in  $M_2$  and  $M_3$ .

The last job is  $J_4 = (2, 2)$ , then  $C_A \geq 3$  and  $C_{OPT} = 2$ .

**Case 2.3.**  $A$  assigns  $J_1 = (1, 2)$  to the machine which  $J_2 = (1, 2)$  is assigned in  $M_2$  and  $M_3$ .

Next job is  $J_4 = (2, 2)$ , then one of  $J_3 = (2, 2)$  and  $J_4 = (2, 2)$  must be assigned, without loss of generality, we assume that  $J_4 = (2, 2)$  be assigned. If  $A$  assigns  $J_4 = (2, 2)$  to  $M_1$ , the last job is  $J_5 = (3, 1)$  and the sequence stops, then  $C_A \geq 5$  and  $C_{OPT} = 3$ . If  $A$  assigns  $J_4 = (2, 2)$  to the machine that is not assigned  $J_1 = (1, 2)$  and  $J_2 = (1, 2)$  in  $M_2$  and  $M_3$ , the last two jobs are  $J_5 = (2, 1)$  and  $J_6 = (4, 2)$ , then  $C_A \geq 6$  and  $C_{OPT} = 4$ . If  $A$  assigns  $J_4 = (2, 2)$  to the machine which  $J_1 = (1, 2)$  and  $J_2 = (1, 2)$  is assigned in  $M_2$  and  $M_3$ , the sequence stops, then  $C_A \geq 4$  and  $C_{OPT} = 2$ .

**Case 3.**  $B^3 = J_1 = (1, 2)$ , and assigns  $J_3 = (2, 2)$  to someone machine.

**Case 3.1.**  $A$  assigns  $J_3 = (2, 2)$  to  $M_1$ .

The last job  $J_4 = (2, 1)$  arrives, then  $C_A \geq 4$  and  $C_{OPT} = 2$ .

**Case 3.2.**  $A$  assigns  $J_3 = (2, 2)$  to the machine which  $J_2 = (1, 2)$  is assigned in  $M_2$  and  $M_3$ .

The sequence stops, then  $C_A \geq 3$  and  $C_{OPT} = 2$ .

**Case 3.3.**  $A$  assigns  $J_3 = (2, 2)$  to the machine that is not assigned  $J_2 = (1, 2)$  in  $M_2$  and  $M_3$ . The next job  $J_4 = (2, 2)$  arrives.

**Case 3.3.1.**  $B^4 = \phi$ , which means that  $A$  schedules first four jobs to the machines.

If  $A$  assigns  $J_4 = (2, 2)$  to  $M_1$ , and assigns  $J_1 = (1, 2)$  to the machine which  $J_2 = (1, 2)$  is assigned in  $M_2$  and  $M_3$ . The last job is  $J_5 = (3, 1)$ , then  $C_A \geq 5$  and  $C_{OPT} = 3$ . Otherwise, the sequence stops, then  $C_A \geq 3$  and  $C_{OPT} = 2$ .

**Case 3.3.2.**  $B^4 = J_1 = (1, 2)$ , and  $A$  assigns  $J_4 = (2, 2)$  to someone machine.

If  $A$  assigns  $J_4 = (2, 2)$  to  $M_1$ , the last job  $J_5 = (3, 1)$  arrives, then  $C_A \geq 5$  and  $C_{OPT} = 3$ .

If  $A$  not assigns  $J_4 = (2, 2)$  to  $M_1$ , the sequence stops, then  $C_A \geq 3$  and  $C_{OPT} = 2$ .

**Case 3.3.3.**  $B^4 = J_4 = (2, 2)$ , and  $A$  assigns  $J_1 = (1, 2)$  to someone machine.

If  $A$  assigns  $J_1 = (1, 2)$  to  $M_1$ , the sequence stops, then  $C_A \geq 3$  and  $C_{OPT} = 2$ . If  $A$  assigns  $J_1 = (1, 2)$  to the machine which  $J_2 = (1, 2)$  is assigned in  $M_2$  and  $M_3$ , the last two jobs  $J_5 = (2, 1)$  and  $J_6 = (4, 2)$  arrive, then  $C_A \geq 6$  and  $C_{OPT} = 4$ . If  $A$  assigns  $J_1 = (1, 2)$  to the machine which  $J_3 = (2, 2)$  is assigned in  $M_2$  and  $M_3$ , the sequence stops, then  $C_A \geq 3$  and  $C_{OPT} = 2$ .

Our algorithm is a modified version of the algorithm proposed in [13].

---

#### Algorithm $H_1$

---

**Step 1.** Set  $j = 1$  and  $L_1^{j-1} = L_2^{j-1} = L_3^{j-1} = 0$ .

**Step 2.** If  $g_j = 1$ , assign  $J_j$  on  $M_1$ , and set  $L_1^j = L_1^{j-1} + p_j$ ,  $L_2^j = L_2^{j-1}$  and  $L_3^j = L_3^{j-1}$ .

**Step 3.** Else  $g_j = 2$ . Compare job  $J_j$  and the job that in the buffer (if no job in the buffer, regarding it as a job with size 0 in buffer). Put the bigger one job in buffer and denote it be  $B^j$  ( $B^j$  also indicate its size), and the smaller one be  $S^j$  ( $S^j$  also indicate its size).

**Step 3.1.** If  $L_2^{j-1} + S^j \leq \frac{5}{3}LB^j$ , schedule  $S^j$  on  $M_2$ , and update  $L_i^j$ .

**Step 3.2.** If  $L_3^{j-1} + S^j \leq \frac{2}{3}LB^j$ , assign  $S^j$  on  $M_3$ , and update  $L_i^j$ .

Else, assign  $S^j$  on  $M_1$ , and update  $L_i^j$ .

**Step 4.** If there exist another job, set  $j = j + 1$ , return to **step 2**.

Otherwise, assign  $B^j$  to the least loaded machine, and stop.

---

**Theorem 3.** ALGORITHM  $H_1$  for  $P3(1, 2, 2)|buffer|C_{max}$  has a competitive ratio at most  $\frac{5}{3}$ .

**Proof.** Proof by contradiction, we assume this theorem does not hold, then, we consider three cases according to which machine  $B^n$  is assigned to. Remember also using  $B^n$  to denote its size and  $B^n = q_{max}$ .

First, by algorithm  $H_1$ , we can get  $L_2^{n-1} \leq \frac{5}{3}LB^n$  and  $L_3^{n-1} \leq \frac{5}{3}LB^n$ .

**Case 1.**  $B^n$  is assigned to  $M_1$ .

This means  $L_1^{n-1} \leq \min\{L_2^{n-1}, L_3^{n-1}\}$  and  $L_1 = L_1^{n-1} + q_{max}$ ,  $L_2 = L_2^{n-1}$ ,  $L_3 = L_3^{n-1}$ . By the assumption of this theorem does not hold, we have  $L_1^{n-1} + q_{max} > \frac{5}{3}LB^n$ . Since  $q_{max} \leq LB^n$ , we have  $L_1^{n-1} > \frac{2}{3}LB^n$ . By the definition of  $LB^n$ , we

have  $T^n = L_1^{n-1} + L_2^{n-1} + L_3^{n-1} + q_{max} \leq 3LB^n$ , so  $L_2^{n-1} + L_3^{n-1} < \frac{4}{3}LB^n$ , implies  $\min\{L_2^{n-1}, L_3^{n-1}\} < \frac{2}{3}LB^n$ . Then  $L_1^{n-1} > \frac{2}{3}LB^n > \min\{L_2^{n-1}, L_3^{n-1}\}$ , which contradicts with  $L_1^{n-1} \leq \min\{L_2^{n-1}, L_3^{n-1}\}$ .

**Case 2.**  $B^n$  is assigned to  $M_2$ . This means  $L_2^{n-1} \leq \min\{L_1^{n-1}, L_3^{n-1}\}$  and  $L_1 = L_1^{n-1}$ ,  $L_2 = L_2^{n-1} + q_{max}$ ,  $L_3 = L_3^{n-1}$ .

**Case 2.1.**  $L_2 \geq \max\{L_1, L_3\}$ . If  $L_2 = L_2^{n-1} + q_{max} > \frac{5}{3}LB^n$ , since  $q_{max} \leq LB^n$ , we have  $L_2^{n-1} > \frac{2}{3}LB^n$ . Then by  $T = T^n = L_1^{n-1} + L_2^{n-1} + L_3^{n-1} + q_{max} \leq 3LB^n = 3LB$ , so  $L_1^{n-1} + L_3^{n-1} < \frac{4}{3}LB^n$ , implying that  $\min\{L_1^{n-1}, L_3^{n-1}\} < \frac{2}{3}LB^n$ , which contradicts with  $L_2^{n-1} > \frac{2}{3}LB^n$  and  $L_2^{n-1} \leq \min\{L_1^{n-1}, L_3^{n-1}\}$ .

**Case 2.2.**  $L_1 \geq \max\{L_2, L_3\}$ .

If  $L_1 > \frac{5}{3}LB^n$ , since  $T_1 = T_1^n \leq LB^n$ , there must exist at least one job with  $g_j = 2$  scheduled on  $M_1$ . Let job  $J_t$  be the last job of hierarchy 2 scheduled on  $M_1$ . Let  $T_1'$  be the total size of the jobs on  $M_1$  after assigned job  $J_t$ , and the hierarchy of these jobs is 1. When job  $J_t$  is assigned on  $M_1$ , by algorithm  $H_1$ , we have  $L_2^{t-1} + p_t > \frac{5}{3}LB^t$  and  $L_3^{t-1} + p_t > \frac{5}{3}LB^t$ , implies  $L_2^{t-1} > \frac{2}{3}LB^t$  and  $L_3^{t-1} > \frac{2}{3}LB^t$ . Since  $T^t = L_1^{t-1} + L_2^{t-1} + L_3^{t-1} + p_t + q_{max}^t \leq 3LB^t$ , so  $L_1^{t-1} + q_{max}^t < \frac{2}{3}LB^t$ , implies  $L_1^{t-1} + p_t < \frac{2}{3}LB^t \leq \frac{2}{3}LB^n$ . Since  $L_1 = L_1^n = L_1^{t-1} + p_t + T_1' > \frac{5}{3}LB^n$  and  $T_1' \leq T_1 \leq LB^n$ , we have  $L_1^{t-1} + p_t > \frac{2}{3}LB^n$ , which cause a contradiction with the  $L_1^{t-1} + p_t < \frac{2}{3}LB^n$ .

**Case 3.**  $B^n$  is assigned to  $M_3$ . This case is similar as the **Case 2**.

Thus, this theorem holds, we get  $\frac{C_{H1}}{C_{OPT}} \leq \frac{5}{3}$ .

## 4 Two Machines with Hierarchy 1

In this section, we consider the hierarchy of machine  $M_1$  and  $M_2$  is 1, the hierarchy of machine  $M_3$  is 2, and a buffer can temporarily store at most one job. When a new job arrives, we have to assign it on a machine irrecoverably; or temporarily store it in the buffer. We give a lower bound  $\frac{3}{2}$  and design an online algorithm with the competitive ratio at most  $\frac{12}{7}$ . In this section, we let

$$LB^j = \max\left\{\frac{T_1^j}{2}, p_{max}^j, q_{max}^j, \frac{T_1^j + T_2^j}{3}\right\}, \quad (3)$$

clearly,  $LB = LB^n$ .

**Lemma 4.** For the online problem  $P3(1,1,2)|buffer|C_{max}$ , the optimal maximum machine load is at least  $LB^j$  after scheduled the job  $J_j$  for  $j = 1, 2, \dots, n$ .

**Proof.** We denote the optimal maximum machine load after scheduled  $J_j$  as  $C_{OPT}^j$ . For the online problem  $P3(1,1,2)|buffer|C_{max}$ , after the job  $J_j$  be scheduled, it is clearly that the  $C_{OPT}^j \geq \max\left\{\frac{T_1^j + T_2^j}{3}, p_{max}^j, q_{max}^j\right\}$ . Note that all jobs with hierarchy 1 can be processed on machine  $M_1$  and  $M_2$ , which implies  $C_{OPT}^j \geq \frac{T_1^j}{2}$ . So, the lemma holds.

Lemma 4 implies

$$C_{OPT} \geq LB = \max \left\{ \frac{T_1}{2}, p_{max}, q_{max}, \frac{T_1 + T_2}{3} \right\}. \quad (4)$$

**Theorem 5.** Any online algorithm  $A$  for  $P3(1,1,2)|buffer|C_{max}$  has a competitive ratio at least  $\frac{3}{2}$ .

**Proof.** For any algorithm  $A$ , the first three jobs in the sequence are  $J_1 = (1,2)$ ,  $J_2 = (1,1)$  and  $J_3 = (1,1)$ .

**Case 1.**  $B^3 = \phi$ , that means  $A$  assigns first three jobs to the machines.

If  $A$  assigns at least two jobs to  $M_1$  or  $M_2$ , the sequence stops, we have  $C_A \geq 2$ ,  $C_{OPT} = 1$ . Else, the last job  $J_4 = (2,1)$  arrives, then  $C_A = 3$ ,  $C_{OPT} = 2$ , implies  $\frac{C_A}{C_{OPT}} \geq \frac{3}{2}$ .

**Case 2.**  $B^3 = J_1 = (1,2)$ .

If  $A$  assigns  $J_2 = (1,1)$  and  $J_3 = (1,1)$  to  $M_1$  or  $M_2$ , the sequence stops, then  $C_A \geq 2$ ,  $C_{OPT} = 1$ . If  $A$  assigns  $J_2 = (1,1)$  and  $J_3 = (1,1)$  to  $M_1$  and  $M_2$  respectively, the last job  $J_4 = (2,1)$  arrives, then  $C_A \geq 3$ ,  $C_{OPT} = 2$ .

**Case 3.**  $B^3 = J_2 = (1,1)$  ( $B^3 = J_3 = (1,1)$ ) that is same as  $B^3 = J_2 = (1,1)$ .

**Case 3.1.**  $A$  assigns  $J_3 = (1,1)$  and  $J_1 = (1,2)$  to  $M_1$  and  $M_2$ . The sequence stops, we have  $C_A \geq 2$ ,  $C_{OPT} = 1$ .

**Case 3.2.**  $A$  assigns  $J_3 = (1,1)$  to  $M_1$  or  $M_2$  and assigns  $J_1 = (1,2)$  to  $M_3$ . Next job  $J_4 = (2,2)$  arrives.

**Case 3.2.1.**  $B^4 = \phi$ .

If  $A$  assigns the  $J_4 = (2,2)$  and at least one another job to the same machine, the sequence stops, we have  $C_A \geq 3$ ,  $C_{OPT} = 2$ . Else, the last job is  $J_5 = (1,1)$ , we have  $C_A = 3$ ,  $C_{OPT} = 2$ .

**Case 3.2.2.**  $B^4 = J_2 = (1,1)$ .

If  $A$  assigns  $J_4 = (2,2)$  to machine that is assigned one job, sequence stops, we have  $C_A \geq 3$ ,  $C_{OPT} = 2$ . If  $A$  assigns job  $J_4 = (2,2)$  to the machine that is not assigned a job, the last job  $J_5 = (1,1)$  arrives, then  $C_A \geq 3$ ,  $C_{OPT} = 2$ .

**Case 3.2.3.**  $B^4 = J_4 = (2,2)$ .

If  $A$  assigns  $J_2 = (1,1)$  to  $M_1$  and  $M_2$  that is not assigned job  $J_3 = (1,1)$ , the sequence stops, we have  $C_A = 3$ ,  $C_{OPT} = 2$ . If  $A$  assigns the  $J_2 = (1,1)$  to the machine which  $J_3 = (1,1)$  is assigned in  $M_1$  and  $M_2$ , the last job  $J_5 = (1,1)$  arrives, then  $C_A \geq 3$ ,  $C_{OPT} = 2$ .

The main ideal of algorithm  $H_2$  is to schedule as many jobs with hierarchy 2 as possible to  $M_3$ , the details of algorithm  $H_2$  are as follows.

**Algorithm  $H_2$** 

**Step 1.** Let  $j = 1$  and  $L_1^{j-1} = L_2^{j-1} = L_3^{j-1} = 0$ .

**Step 2.** If  $g_j = 1$ , assign  $J_j$  to  $M_1$  and  $M_2$  which the load is the least, and update  $L_i^j$ .

**Step 3.** Else  $g_j = 2$ . Compare job  $J_j$  with the job in the buffer (if no job in buffer, regarding it as a job with size 0 in buffer). Put the bigger one job in buffer and denote it be  $B^j$  ( $B^j$  also indicate its size), and the smaller one be  $S^j$  ( $S^j$  also indicate its size).

**Step 3.1.** If  $L_3^{j-1} + S^j \leq \frac{4}{7}T_2^j$ , assign  $S^j$  on  $M_3$ , and update  $L_i^j$ .

**Step 3.2.** Else, assign  $S^j$  to  $M_1$  and  $M_2$  which the load is the least, and update  $L_i^j$ .

**Step 4.** When no new job arrives, if  $L_3^{n-1} + B^n \leq \max\{L_1^{n-1}, L_2^{n-1}\}$ , assign  $B^n$  on  $M_3$ ; else, assign  $B^j$  to the least loaded machine, and stop.

**Theorem 6.** The competitive ratio of ALGORITHM  $H_2$  for  $P3(1, 1, 2)|buffer|C_{max}$  at most  $\frac{12}{7}$ .

**Proof.** Consider three cases according to which machine  $B^n$  is assigned to. We denote the total processing time of the jobs of hierarchy 2 that assigned to  $M_i$  as  $L_i(2)$ , for  $i = 1, 2$ , and remember  $B^n = q_{max}$ .

**Case 1.**  $B^n$  is assigned to  $M_3$ .

**Case 1.1.**  $L_3 > \max\{L_1, L_2\}$ , this means  $L_3^{n-1} \leq \min\{L_1^{n-1}, L_2^{n-1}\}$  and  $L_1 = L_1^{n-1}$ ,  $L_2 = L_2^{n-1}$ ,  $L_3 = L_3^{n-1} + q_{max}$ . Since  $L_3^{n-1} \leq \min\{L_1^{n-1}, L_2^{n-1}\}$ , we have

$$C_{H_2} = L_3 = L_3^{n-1} + q_{max} \leq \frac{T_1 + T_2 - q_{max}}{3} + q_{max} = \frac{T_1 + T_2}{3} + \frac{2}{3}q_{max}.$$

Following (4), we have

$$C_{OPT} \geq \max\left\{\frac{T_1}{2}, p_{max}, q_{max}, \frac{T_1 + T_2}{3}\right\}.$$

So the competitive ratio  $\frac{C_{H_2}}{C_{OPT}} \leq \frac{5}{3}$ .

**Case 1.2.**  $L_3 \leq \max\{L_1, L_2\}$ .

If no jobs of hierarchy 2 be assigned to  $M_1$  and  $M_2$ , then  $L_3 = L_3^{n-1} + q_{max} = T_2$ . By algorithm  $H_2$ , in this situation, the load between the  $M_1$  and  $M_2$  not exceed  $p_{max}$ , so

$$C_{H_2} = \max\{L_1, L_2\} \leq \frac{T_1 - p_{max}}{2} + p_{max} = \frac{T_1}{2} + \frac{p_{max}}{2}.$$

We get  $\frac{C_{H_2}}{C_{OPT}} \leq \frac{3}{2}$ .

Else, we let the last job of hierarchy 2 that be assigned to  $M_1$  and  $M_2$  is  $J_t$ , by algorithm  $H_2$ , we have  $L_3^{t-1} + q'_{max} \geq L_3^{t-1} + p_t > \frac{4}{7}T_2^t$ . If there exist jobs of hierarchy 2 after  $J_t$ , they will all be assigned to  $M_3$ . Hence,  $L_3 > \frac{4}{7}T_2$ , implies  $L_1(2) + L_2(2) < \frac{3}{7}T_2$ .



By algorithm  $H_2$ , in any time, we know the load between the  $M_1$  and  $M_2$  not exceed  $\max\{p_{max}, q_{max}\}$ , so

$$\begin{aligned} C_{H_2} = \max\{L_1, L_2\} &\leq \frac{3T_2/7 + T_1 - \max\{p_{max}, q_{max}\}}{2} + \max\{p_{max}, q_{max}\} \\ &= \frac{3}{14}(T_1 + T_2) + \frac{2}{7}T_1 + \frac{\max\{p_{max}, q_{max}\}}{2}. \end{aligned}$$

Following (4), we know  $C_{OPT} \geq \max\{T_1/2, p_{max}, q_{max}, (T_1 + T_2)/3\}$ , then, we get  $\frac{C_{H_2}}{C_{OPT}} \leq \frac{12}{7}$ .

**Case 2.**  $B^n$  is assigned to  $M_1$ . This means  $L_1^{n-1} \leq \min\{L_2^{n-1}, L_3^{n-1}\}$  and  $L_1 = L_1^{n-1} + q_{max}$ ,  $L_2 = L_2^{n-1}$ ,  $L_3 = L_3^{n-1}$ .

**Case 2.1.**  $L_1 \geq \max\{L_2, L_3\}$ . Since  $L_1^{n-1} \leq \min\{L_2^{n-1}, L_3^{n-1}\}$ , by algorithm  $H_2$ , we get

$$C_{H_2} = L_1 \leq \frac{T_1 + T_2 - q_{max}}{3} + q_{max} = \frac{T_1 + T_2}{3} + \frac{2}{3}q_{max},$$

By (4), implying that  $\frac{C_{H_2}}{C_{OPT}} \leq \frac{5}{3}$ .

**Case 2.2.**  $L_1 < \max\{L_2, L_3\}$ . If  $L_2 < L_3$ , by algorithm  $H_2$ , we know  $C_{H_2} = L_3 \leq \frac{4}{7}T_2$ , following (4), we get  $\frac{C_{H_2}}{C_{OPT}} \leq \frac{L_3}{T_2/3} \leq \frac{12}{7}$ . If  $L_2 \geq L_3$ , we assume  $L_2 > \frac{12}{7}LB$ , because  $B^n$  is assigned to  $M_1$  and the load between the  $M_1$  and  $M_2$  not exceed  $\max\{p_{max}, q_{max}\}$ , we have  $L_1 \geq L_2 - \max\{p_{max}, q_{max}\}$ . By algorithm  $H_2$ , because  $B^n$  is not assigned to  $M_3$ , we get  $L_3 > L_2 - B^n = L_2 - q_{max}$ . Hence,

$$\begin{aligned} T_1 + T_2 = L_1 + L_2 + L_3 &> 3L_2 - \max\{p_{max}, q_{max}\} - q_{max} \\ &\geq \frac{36}{7}LB - 2LB \\ &= \frac{22}{7}LB \\ &> T_1 + T_2. \end{aligned}$$

Contradiction, so  $L_2 \leq \frac{12}{7}LB$ , and  $\frac{C_{H_2}}{C_{OPT}} \leq \frac{L_2}{LB} \leq \frac{12}{7}$ .

**Case 3.**  $B^n$  is assigned to  $M_2$ . This case is similar as the **Case 2**.

Hence, this theorem holds, obtain  $\frac{C_{H_2}}{C_{OPT}} \leq \frac{12}{7}$ .

## 5 Conclusion

We consider two cases for the online problem  $P3|Gos, buffer|C_{max}$ , but we can't get optimal online algorithms. It's valuable to design their optimal algorithms in future.

**Acknowledgements.** The work is supported in part by the Program for Excellent Young Talents of Yunnan University, Training Program of National Science Fund for Distinguished Young Scholars, IRTSTYN, and Key Joint Project of the Science and Technology Department of Yunnan Province and Yunnan University [No. 2018FY001(-014)].

## References

1. Hwang, H., Chang, S.Y., Lee, K.: Parallel machine scheduling under a grade of service provision. *Comput. Oper. Res.* **31**(12), 2055–2061 (2004)
2. Ou, J., Leung, J.Y.T., Li, C.L.: Scheduling parallel machines with inclusive processing set restrictions. *Nav. Res. Logist.* **55**(4), 328–338 (2008)
3. Li, W., Li, J., Zhang, T.: Two approximation schemes for scheduling on parallel machines under a grade of service provision. *Asia Pac. J. Oper. Res.* **29**(05), 83–94 (2012)
4. Park, J., Chang, S.Y., Lee, K.: Online and semi-online scheduling of two machines under a grade of service provision. *Oper. Res. Lett.* **34**(6), 692–696 (2006)
5. Jiang, Y., He, Y., Tang, C.: Optimal online algorithms for scheduling on two identical machines under a grade of service. *J. Zhejiang Univ. Sci. A* **7**(3), 309–314 (2006)
6. Lim, K., Park, J., Chang, S.Y., Lee, K.: Online and semi-online scheduling of three machines under a GoS provision. In: Working Paper, Department of Industrial and Management Engineering, Pohang University of Science and Technology, Republic of Korea (2010)
7. Wu, Y., Ji, M., Yang, Q.: Optimal semi-online scheduling algorithm on two parallel identical machines under a grade of service provision. *Int. J. Prod. Econ.* **135**(1), 367–371 (2012)
8. Zhang, A., Jiang, Y., Tan, Z.: Online parallel machines scheduling with two hierarchies. *Theoret. Comput. Sci.* **410**(38), 3597–3605 (2009)
9. Zhang, A., Jiang, Y., Fan, L., Hu, J.: Optimal online algorithms on two hierarchical machines with tightly-grouped processing times. *J. Comb. Optim.* **29**(4), 781–795 (2015)
10. Englert, M., Ozmen, D., Westermann, M.: The power of reordering for online minimum makespan scheduling. In: 2008 49th Annual IEEE Symposium on Foundations of Computer Science, pp. 603–612. IEEE (2008)
11. Lan, Y., Chen, X., Ding, N., Han, X.: Online minimum makespan scheduling with a buffer. In: Proceedings of the Joint International Conference on Frontiers in Algorithmics and Algorithmic Aspects in Information and Management, pp. 161–171 (2012)
12. Zhang, G.: A simple semi on-line algorithm for  $P2//C_{max}$  with a buffer. *Inf. Process. Lett.* **61**(3), 145–148 (1997)
13. Chen, X., Xu, Z., Dosa, G., Han, X., Jiang, H.: Semi-online hierarchical scheduling problems with buffer or rearrangements. *Inf. Process. Lett.* **113**(4), 127–131 (2013)
14. Li, J., Li, W., Li, J.: Polynomial approximation schemes for the max-min allocation problem under a grade of service provision. *Discrete Math. Algorithms Appl.* **1**(03), 355–368 (2009)
15. Xiao, M., Wu, G., Li, W.: Semi-online machine covering on two hierarchical machines with known total size of low-hierarchy jobs. In: Sun, X., He, K., Chen, X. (eds.) NCTCS 2019. CCIS, vol. 1069, pp. 95–108. Springer, Singapore (2019). [https://doi.org/10.1007/978-981-15-0105-0\\_7](https://doi.org/10.1007/978-981-15-0105-0_7)