



Article Online Bi-Criteria Scheduling on Batch Machines with Machine Costs

Wenhua Li *¹⁰, Weina Zhai and Xing Chai

School of Mathematics and Statistics, Zhengzhou University, Zhengzhou 450001, China; zhaiweinalucky@163.com (W.Z.); chaixingstudy@163.com (X.C.)

* Correspondence: liwenhua@zzu.edu.cn

Received: 26 August 2019; Accepted: 8 October 2019; Published: 13 October 2019



Abstract: We consider online scheduling with bi-criteria on parallel batch machines, where the batch capacity is unbounded. In this paper, online means that jobs' arrival is over time. The objective is to minimize the maximum machine cost subject to the makespan being at its minimum. In unbounded parallel batch scheduling, a machine can process several jobs simultaneously as a batch. The processing time of a job and a batch is equal to 1. When job J_j is processed on machine M_i , it results cost c_{ij} . We only consider two types of cost functions: $c_{ij} = a + c_j$ and $c_{ij} = a \cdot c_j$, where *a* is the fixed cost of machines and c_j is the cost of job J_j . The number of jobs is *n* and the number of machines is *m*. For this problem, we provide two online algorithms, which are showed to be the best possible with a competitive ratio of $(1 + \beta_m, \lceil \frac{n}{m} \rceil)$, where β_m is the positive root of the equation $(1 + \beta_m)^{m+1} = \beta_m + 2$.

Keywords: bi-criteria scheduling; online algorithm; makespan; maximum machine cost; competitive ratio

1. Introduction

In this article, we consider an online bi-criteria scheduling problem to minimize the maximum machine cost subject to the makespan achieving its minimum. Online means that jobs' arrival is over time. It means, until when a job arrives, all information about it, including its arrival time, processing time and processing cost, is not known by us. For a minimization problem that is relevant to a single objective function, the competitive ratio of an online algorithm *A* is defined to be

$$\rho_A = \sup\{\frac{f(A, I)}{OPT(A, I)}: I \text{ is any job instance and } OPT(A, I) > 0\}.$$

Here, f(A, I) is the objective value in algorithm A for any input instance I, OPT(A, I) is the optimal objective value in the offline circumstance, respectively. We say algorithm A is the best possible if there doesn't exist any algorithm A' such that $\rho_{A'} < \rho_A$.

Parallel-batch was first studied by Uzsoy et al. [1,2]. There are two classes of parallel-batch models that have been widely considered in the literature, the unbounded version $b = \infty$ and the bounded version $b < \infty$, where *b* is the batch capacity. That is, at most *b* jobs can be processed simultaneously in one batch. The processing time of one batch is defined as the longest job in it. Since in this paper we consider that the jobs are with identical processing time, the processing time of the batches is 1.

In traditional scheduling theory, most problems are concerned with the minimization of one certain function. There have been many achievements such as, for minimizing maximum completion time when jobs have the same processing times, Zhang et al. [3] provided two best possible online algorithmss with $1 + \beta_m$ and $1 + \alpha$ -competitive ratio for the unbounded model $b = \infty$ and bounded model $b < \infty$, respectively, and β_m satisfies $(1 + \beta_m)^{m+1} = \beta_m + 2$, $\alpha = \frac{\sqrt{5}-1}{2}$. When jobs have diverse processing times, Tian et al. [4] and Liu et al. [5] independently gave two best possible

online algorithms with competitive ratio of $1 + \alpha_m$, and α_m is the positive solution of the equation $\alpha_m^2 + m \cdot \alpha_m - 1 = 0$. Fang et al. [6] presented a best possible online algorithm with a competitive ratio of $1 + \phi$ for a set of processing time scheduling problems, where $\phi = \frac{\sqrt{5}-1}{2}$. For minimizing a maximum weighted completion time problem, Li et al. [7] established a best possible online algorithm with a competitive ratio of $\frac{\sqrt{5}+1}{2}$. For minimizing total weighted completion time problem, Cao et al. [8] gave a best possible online algorithm with a competitive ratio of $\rho_m^{m+1} - \rho_m - 1 = 0$. Some reviews for parallel-batch scheduling research can be found in [9–14].

Today, with the rapid development of science and technology, minimization of one certain function doesn't satisfy the needs of things. In addition, jobs' objective functions may have certain kinds of aspects to minimize. In recent years, there have been some results about minimizing bi-criteria objective functions such as Ma et al. [15], who considered an online trade-off scheduling problem that minimize makespan and total weighted completion time on a single machine, presenting a nondominated $(1 + \alpha, 1 + \frac{1}{\alpha})$ – competitive online algorithm for each α with $0 < \alpha \leq 1$. Liu et al. [16] considered the single machine online trade-off scheduling problem, which minimizes the makespan and maximum lateness. They established a nondominated $(\rho, 1 + \frac{1}{\rho})$ – competitive online algorithm with $1 \leq \rho \leq \frac{\sqrt{5}+1}{2}$. Here, a (ρ_1, ρ_2) -competitive online algorithm is called nondominated if there is no other (ρ'_1, ρ'_2) -competitive online algorithm A' such that $(\rho'_1, \rho'_2) \leq (\rho_1, \rho_2)$ and either $\rho'_1 < \rho_1$ or $\rho'_2 < \rho_2$. In addition, Lee et al. [17] considered two bi-criteria scheduling problems: one is minimizing the total machine cost subject to the total completion time achieving its minimum, another is minimizing the total machine cost subject to the makespan achieves its minimum. As these two problems are strongly NP-hard, they proposed fast heuristics and found their worst-case performance bounds.

Another class of scheduling problems with bi-criteria is to minimize a secondary objective function f_2 subject to a primary objective function f_1 being at its minimum, and the objective is denoted by $Lex(f_1, f_2)$. In practical production, the producer wants to reduce the cost of the machine as soon as it is finished. Given *m* parallel batch machines M_i , $1 \le i \le m$, and *n* jobs J_j , $1 \le j \le n$. Every machine has a fixed cost a_i , job J_j has cost c_j , and $1 \le j \le n$. When job J_j is processed on machine M_i , this will result in different costs c_{ij} , $1 \le i \le m$, $1 \le j \le n$. Suppose that $x_{ij} = 1$ if job j is processed on machine *i*, otherwise $x_{ij} = 0$. Thus, the total machine cost, is named *TMC*, and *TMC* $=\sum_{i=1}^{m}\sum_{j=1}^{n}c_{ij}x_{ij}$, and the maximum machine cost is named *MMC*, and *MMC* = max_{i=1}^{m}{\sum_{j=1}^{n}c_{ij}x_{ij}}. In Lee et al. [17], they studied the offline bi-criteria scheduling problems, for which the objective functions are minimizing MMC subject to the constraint that $\sum C_i$ is minimized and minimizing TMC subject to the constraint that C_{max} is minimized, where C_i is the completion time of job J_i , $\sum C_i$ is total completion time of jobs, and C_{max} is the maximum completion time of jobs. They considered three kinds of cost functions: $c_{ij} = c_j$, $c_{ij} = a_i + c_j$, and $c_{ij} = a_i \cdot c_j$. In our article, we consider online algorithms to minimize the maximum machine cost subject to the makespan achieving its minimum, and the objective function is denoted as $Lex(C_{max}, MMC)$. Here, we assume that all machines have the same fixed cost *a*, and we consider two kinds of costs: $c_{ij} = a + c_j$ and $c_{ij} = a \cdot c_j$, $1 \le i \le m$, $1 \le j \le n$. Since the jobs are processed in batches in our model, the cost of a batch processed on some machine is defined as the maximum cost of the jobs in it. Then, the cost of one machine is the total cost of the batches on it. This problem can be written in the three-field notation as $Pm|online, p-batch, b = \infty$, $p_j = 1 | Lex(C_{\max}, MMC)$, when $c_{ij} = a + c_j$ or $c_{ij} = a \cdot c_j$, $1 \le i \le m, 1 \le j \le n$.

For the online scheduling problem to minimize a primary objective function f_1 and a secondary objective function f_2 , we say that an online algorithm A is $(\rho_{A,1}, \rho_{A,2})$ -competitive if it is $\rho_{A,1}$ -competitive when minimizing f_1 and $\rho_{A,2}$ -competitive when minimizing f_2 . In the case that $\rho_{A,1}$ is the competitive ratio of A for minimizing f_1 and $\rho_{A,2}$ is the competitive ratio of A for minimizing f_2 , we also say that the online algorithm A has a competitive ratio of $(\rho_{A,1}, \rho_{A,2})$. Suppose that the best possible competitive ratio is ρ when minimizing f_1 . We say that the online algorithm A is the best possible, if $\rho_{A,1} = \rho$ and there is no other online algorithm A' such that $\rho_{A',1} = \rho$ and $\rho_{A',2} < \rho_{A,2}$.

This paper is organized as four sections as follows. In Section 2, the parameters and notations are introduced. In Section 3, the lower bounds of the competitive ratio are presented. In Section 4, two best possible online algorithms with a competitive ratio of $(1 + \beta_m, \lceil \frac{n}{m} \rceil)$ are showed, where β_m is the positive root of the equation $(1 + \beta_m)^{m+1} = \beta_m + 2$.

The objective considered in this paper is to minimize the maximum machine cost subject to the makespan being at its minimum. In addition, the algorithms studied in this paper are extensions of the results about makespan in the literature.

2. Preliminaries and Notations

Some preliminaries and notations that will be used in the paper are shown in the following:

- $c_{\max} = \max\{c_1, c_2, \cdots, c_n\}$: The maximum cost of the jobs;
- B_{l_i} : The *l*th batch on machine M_i ;
- c_{B_l} : The cost of batch B_l , denoted as c_{B_l} , is the maximum cost of jobs belonging to batch B_l ;
- c_{M_i} : The cost of machine M_i , i.e., the total cost of all batches on machine M_i ;
- *U*(*t*): The set of the unscheduled available jobs at time *t*;
- $r_{\max}(t)$: The last release time of jobs in U(t);
- *r_i*: The release time of job *J_i*;
- *r*_{max}: The last release time of all jobs;
- *s*_{*l*}: The starting time of batch *B*_{*l*} by an online algorithm;
- *σ* and *π*: The schedules generated by an online algorithm and an offline optimal algorithm, respectively;
- $C_{\max}(\sigma)$ and $C_{\max}(\pi)$: The maximum completion time in σ and the maximum completion time in π , respectively;
- *MMC*(*σ*) and *MMC*(*π*): The maximum machine cost in *σ* and the maximum machine cost in *π*, respectively.

For minimizing a single criterion, there have been many results about minimizing makespan. For example, the following lemma shows one result to minimize C_{max} . From Theorem 3 of Zhang et al. [3], we have

Lemma 1. For problem $Pm|online, p - batch, b = \infty, p_j = 1|C_{max}$, the competitive ratio of the best possible online algorithm is $1 + \beta_m$, where β_m is the positive root of the equation $(1 + \beta_m)^{m+1} = \beta_m + 2$.

3. The Lower Bound

Theorem 1. For problem $Pm|online, p - batch, b = \infty, p_j = 1|Lex(C_{max}, MMC), when <math>c_{ij} = a + c_j$ or $c_{ij} = a \cdot c_j$, $1 \le i \le m$, $1 \le j \le n$, there exists no online algorithm with a competitive ratio less than $(1 + \beta_m, \lceil \frac{n}{m} \rceil)$.

Proof. Supposing that the fixed cost of each machine is *a*, the cost of each job is 1, which means $c_{\max} = 1$. Let ψ be the set of the best possible solutions of the objective function C_{\max} . Then, for $\forall \sigma \in \psi$, we prove that there exists no online algorithm that satisfies $\frac{MMC(\sigma)}{MMC(\pi)} < \lceil \frac{n}{m} \rceil$, subjected to the constraint

$$\frac{C_{\max}(\sigma)}{C_{\max}(\pi)} \le 1 + \beta_m. \tag{1}$$

We use adversary strategy to prove this conclusion. Let A be an arbitrary online algorithm, and ϵ is an arbitrarily small positive number. Suppose the first job J_1 arrives at 0 and starts at s_1 . From (1), we can know that $s_1 \leq \beta_m$. Otherwise, we have

$$\frac{C_{\max}(\sigma)}{C_{\max}(\pi)} = \frac{s_1 + 1}{r_1 + 1} > 1 + \beta_{m}$$

a contradiction. Job J_{i+1} arrives at $s_i + \epsilon$ and starts at s_{i+1} , $1 \le i \le n-1$. We claim that $s_i \le (1 + \beta_m)r_i + \beta_m$. Otherwise,

$$\frac{C_{\max}(\sigma)}{C_{\max}(\pi)} = \frac{s_i + 1}{r_i + 1} > \frac{(1 + \beta_m)r_i + \beta_m + 1}{r_i + 1} = 1 + \beta_m,$$

Then, $C_{\max}(\sigma) > (1 + \beta_m)C_{\max}(\pi)$, a contradiction.

Hence, *n* jobs are processed as *n* batches on *m* machines.

When $c_{ij} = a + c_j$, after *n* jobs are processed, there must be not less than $\lceil \frac{n}{m} \rceil$ jobs on one machine. Because the cost of each job is 1, the maximum machine cost is $MMC(\sigma) \ge \lceil \frac{n}{m} \rceil \times (a+1)$. In π , all jobs can form one batch starting at the last time when the job arrives, so $MMC(\pi) = a + 1$. Then, we get $\frac{MMC(\sigma)}{MMC(\pi)} \ge \lceil \frac{n}{m} \rceil$.

When $c_{ij} = a \cdot c_j$, similarly after n jobs are finished, there must be one machine that does not have less than $\lceil \frac{n}{m} \rceil$ jobs. Since each job's cost is 1, the maximum machine cost is $MMC(\sigma) \ge \lceil \frac{n}{m} \rceil \times (a \cdot 1)$. In π , all jobs can form one batch starting at the time which the last job arrives, so $MMC(\pi) = a \cdot 1$. Then, we get $\frac{MMC(\sigma)}{MMC(\pi)} \ge \lceil \frac{n}{m} \rceil$.

Therefore, for problem $Pm|online, p - batch, b = \infty, p_j = 1|.Lex(C_{max}, MMC)$, when $c_{ij} = a + c_j$ or $c_{ij} = a \cdot c_j$, $1 \le i \le m$, $1 \le j \le n$, and there exists no online algorithm in which the competitive ratio is less than $(1 + \beta_m, \lceil \frac{n}{m} \rceil)$. \Box

4. Best Possible Online Algorithms

Here, there are two online algorithms for this problem.

Algorithm H^1

At current time *t*, if some machine is idle, $U(t) \neq \emptyset$, when $t \ge (1 + \beta_m)r_{\max}(t) + \beta_m$; then, start the jobs in U(t) as a batch on the idle machine that has the minimum machine cost at the moment. Otherwise, do nothing but wait.

Algorithm H²

At current time *t*, if some machine is idle, $U(t) \neq \emptyset$, when $t \ge (1 + \beta_m)r_{\max}(t) + \beta_m$; then, start the jobs in U(t) as a batch on the idle machine that has the minimum number of batches at the moment. Otherwise, do nothing but wait.

Following the notation in Zhang et al. [3], we also call batches that start at $(1 + \beta_m)r_{max}(t) + \beta_m$ regular batches. From Lemma 1 of Zhang et al. [3], we have

Lemma 2. All batches generated by algorithm H^1 and H^2 are regular batches.

Lemma 3. When $c_{ij} = a + c_j$, Then, $MMC(\pi) = a + c_{max}$; When $c_{ij} = a \cdot c_j$, then $MMC(\pi) = a \cdot c_{max}$.

Proof. The offline optimal objective case of the maximum machine cost *MMC* is: all jobs can form one batch starting at the last arrival time on an arbitrary machine. Thus, when $c_{ij} = a + c_j$, the maximum machine cost is $MMC(\pi) = a + c_{max}$; when $c_{ij} = a \cdot c_j$, the maximum machine cost is $MMC(\pi) = a \cdot c_{max}$. \Box

Theorem 2. For problem $Pm|online, p - batch, b = \infty, p_j = 1|Lex(C_{max}, MMC), when <math>c_{ij} = a + c_j$ or $c_{ij} = a \cdot c_j, 1 \le i \le m, 1 \le j \le n$, algorithm H^1 is a best possible online algorithm with a competitive ratio of $(1 + \beta_m, \lceil \frac{n}{m} \rceil)$.

Proof. When $c_{ij} = a + c_j$, suppose that the schedule generated by algorithm H_1 is σ'_1 . From Lemmas 1 and 2, we can know that

$$C_{\max}(\sigma_1') = (1 + \beta_m)r_{\max} + \beta_m + 1 = (1 + \beta_m)(r_{\max} + 1) \le (1 + \beta_m)C_{\max}(\pi).$$

In the following, we prove $MMC(\sigma'_1) \leq \lceil \frac{n}{m} \rceil \cdot MMC(\pi)$. Suppose, in σ'_1 , that the machine M_x has the maximum machine cost, Then,

$$MMC(\sigma_1') = c_{M_x}.$$
 (2)

We distinguish the following cases:

Case 1 The number of batches on machine M_x is no more than $\left\lceil \frac{n}{m} \right\rceil$. Thus,

$$c_{M_x} \leq \lceil \frac{n}{m} \rceil \cdot a + \sum_{1 \leq l \leq \lceil \frac{n}{m} \rceil} c_{B_{l_x}}$$
$$\leq \lceil \frac{n}{m} \rceil \cdot (a + c_{\max}).$$

In addition, by (2), we have

$$MMC(\sigma'_1) = c_{M_x} \leq \lceil \frac{n}{m} \rceil \cdot (a + c_{\max})$$

From Lemma 3, we get

$$MMC(\sigma'_1) \leq \lceil \frac{n}{m} \rceil \cdot MMC(\pi).$$

Case 2 The number of batches on machine M_x is more than $\lceil \frac{n}{m} \rceil$. Thus, there must be one machine that has less than $\lceil \frac{n}{m} \rceil$ batches. Suppose machine $M_{x'}$ is the machine that has less than $\lceil \frac{n}{m} \rceil$ batches; let B_y be the last batch to process on machine M_x .

Firstly, if machine $M_{x'}$ is idle directly before s_y , let the total cost of batches that start before s_y on $M_{x'}$ be V_1 . From algorithm H^1 , because the number of batches on machine $M_{x'}$ is no more than $\lceil \frac{n}{m} \rceil - 1$, so

$$V_1 \le \left(\left\lceil \frac{n}{m} \right\rceil - 1 \right) \cdot a + \sum_{1 \le l \le \left\lceil \frac{n}{m} \right\rceil - 1} c_{B_{l_{x'}}}.$$
(3)

Moreover, from algorithm H^1 , the total cost of batches that start before s_y on machine M_x is not greater than the total cost of batches start before s_y on machine $M_{x'}$, that is

$$c_{M_x} - (a + c_{B_y}) \le V_1,$$

then from (3), we have

$$\begin{array}{rcl} c_{M_x} & \leq & V_1 + (a + c_{B_y}) \\ & \leq & \left(\lceil \frac{n}{m} \rceil - 1 \right) \cdot a + \sum_{1 \leq l \leq \lceil \frac{n}{m} \rceil - 1} c_{B_{l_{x'}}} + (a + c_{B_y}) \\ & \leq & \lceil \frac{n}{m} \rceil \cdot a + \left(\lceil \frac{n}{m} \rceil - 1 \right) \cdot c_{\max} + c_{\max} \\ & = & \lceil \frac{n}{m} \rceil \cdot (a + c_{\max}). \end{array}$$

In addition, by (2), we have

$$MMC(\sigma'_1) = c_{M_x} \leq \lceil \frac{n}{m} \rceil \cdot (a + c_{\max}).$$

Lemma 3 shows that

$$MMC(\sigma'_1) \leq \lceil \frac{n}{m} \rceil \cdot MMC(\pi).$$

Secondly, if machine $M_{x'}$ is busy directly before s_y , let $B_{y'}$ be the last batch that is on machine M_x such that machine $M_{x'}$ is idle directly before $s_{y'}$. Supposing that there are k batches between B_y and $B_{y'}$, we denote them as B'_1, B'_2, \dots, B'_k .

Claim $B_{y'}$ must exist and is not the first batch on M_x .

Otherwise, $B_{y'}$ does not exist or it is the first batch on M_x . This means that machine $M_{x'}$ is busy when B_{2_x}, B_{3_x}, \cdots start on machine M_x . Since the number of batches on machine M_x is more than $\lceil \frac{n}{m} \rceil$, the number of batches on machine $M_{x'}$ must be not less than $\lceil \frac{n}{m} \rceil$, contradicting the assumption that the number of batches on machine $M_{x'}$ is less than $\lceil \frac{n}{m} \rceil$. Thus, the claim holds.

Let the total cost of batches start before $s_{y'}$ on machine $M_{x'}$ be V_2 . Then, from the definition of $B_{y'}$ and $M_{x'}$, the number of batches that start before $s_{y'}$ on machine $M_{x'}$ is no more than $\lceil \frac{n}{m} \rceil - (k+2)$. Thus,

$$V_{2} \leq \left[\left\lceil \frac{n}{m}\right\rceil - (k+2)\right] \cdot a + \sum_{1 \leq l \leq \left\lceil \frac{n}{m}\right\rceil - (k+2)} c_{B_{l_{x'}}}$$

$$\leq \left[\left\lceil \frac{n}{m}\right\rceil - (k+2)\right] \cdot a + \left[\left\lceil \frac{n}{m}\right\rceil - (k+2)\right] \cdot c_{\max}$$

$$= \left[\left\lceil \frac{n}{m}\right\rceil - (k+2)\right] \cdot (a + c_{\max}).$$

That is,

$$V_2 \le \left[\left\lceil \frac{n}{m} \right\rceil - (k+2) \right] \cdot (a + c_{\max}).$$
(4)

Furthermore, by algorithm H^1 , the total cost of batches starting before $s_{y'}$ on machine M_x is not greater than the total cost of batches starting before $s_{y'}$ on machine $M_{x'}$, then

$$c_{M_x} - (a + c_{B_y}) - (k \cdot a + \sum_{1 \le l \le k} c_{B'_l}) - (a + c_{B_{y'}}) \le V_2.$$

Thus, from (4), we know that

$$\begin{array}{ll} c_{M_x} & \leq & V_2 + (a + c_{B_{y'}}) + (k \cdot a + \sum_{1 \leq l \leq k} c_{B_l'}) + (a + c_{B_y}) \\ \\ & \leq & \left[\left\lceil \frac{n}{m} \right\rceil - (k + 2) \right] \cdot (a + c_{\max}) + (a + c_{\max}) + (k \cdot a + k \cdot c_{\max}) + (a + c_{\max}) \\ \\ & = & \left\lceil \frac{n}{m} \right\rceil \cdot (a + c_{\max}). \end{array}$$

In addition, by (2), we have

$$MMC(\sigma'_1) = c_{M_x} \leq \lceil \frac{n}{m} \rceil \cdot (a + c_{\max}).$$

Lemma 3 shows that

$$MMC(\sigma'_1) \leq \lceil \frac{n}{m} \rceil \cdot MMC(\pi).$$

We know that $k \ge 1$. When k = 0, similar to the above discussion, the conclusion also holds.

When $c_{ij} = a \cdot c_j$, suppose the schedule produced by algorithm H^1 is σ''_1 . From Lemmas 1 and 2, we obtain that $C_{\max}(\sigma''_1) \leq (1 + \beta_m)C_{\max}(\pi)$. In the following, we want to prove that $MMC(\sigma''_1) \leq \lceil \frac{n}{m} \rceil \cdot MMC(\pi)$. Supposing that machine M_w has maximum machine cost in σ''_1 , Then,

$$MMC(\sigma_1'') = c_{M_w}.$$
(5)

We distinguish the following cases:

Case 3 The number of batches is no more than $\lceil \frac{n}{m} \rceil$ on machine M_w . Then,

$$c_{M_{w}} \leq \sum_{1 \leq l \leq \lceil \frac{n}{m} \rceil} (a \cdot c_{B_{l_{w}}})$$

$$= a \cdot \sum_{1 \leq l \leq \lceil \frac{n}{m} \rceil} c_{B_{l_{w}}}$$

$$\leq \lceil \frac{n}{m} \rceil \cdot a \cdot c_{\max}.$$

$$c_{M_{w}} \leq \lceil \frac{n}{m} \rceil \cdot a \cdot c_{\max}.$$
(6)

Thus,

From (5) and (6), we have

$$MMC(\sigma_1'') = c_{M_w} \le \lceil \frac{n}{m} \rceil \cdot (a \cdot c_{\max}).$$

Lemma 3 shows that

$$MMC(\sigma_1'') \leq \lceil \frac{n}{m} \rceil \cdot MMC(\pi)$$

Case 4 The number of batches is more than $\lceil \frac{n}{m} \rceil$ on machine M_w . Thus, there must be one machine that has fewer than $\lceil \frac{n}{m} \rceil$ batches. Supposing that machine $M_{w'}$ is the machine for which the number of batches on it is less than $\lceil \frac{n}{m} \rceil$, let B_z be the last batch to process on machine M_w .

If machine $M_{w'}$ is idle directly before s_z , we denote the total cost of batches start before s_z on machine $M_{w'}$ as V'_1 , by algorithm H^1 because the number of batches on machine $M_{w'}$ is no more than $\lceil \frac{n}{m} \rceil - 1$, hence

$$V_1' \le a \cdot \sum_{1 \le l \le \lceil \frac{n}{m} \rceil - 1} c_{B_{l_{w'}}}.$$
(7)

Furthermore, by algorithm H^1 , the total cost of batches starting before s_z on machine M_w is not greater than the total cost of batches starting before s_z on machine $M_{w'}$; then,

$$c_{M_w} - (a \cdot c_{B_z}) \le V_1'.$$

From (7), we have

$$c_{M_w} \leq V'_1 + a \cdot c_{B_z}$$

$$\leq a \cdot \sum_{1 \leq l \leq \lceil \frac{n}{m} \rceil - 1} c_{B_{l_{w'}}} + a \cdot c_{B_z}$$

$$\leq a \cdot (\lceil \frac{n}{m} \rceil - 1) \cdot c_{\max} + a \cdot c_{\max}$$

$$= a \cdot \lceil \frac{n}{m} \rceil \cdot c_{\max}.$$

In addition, by (6), we have

$$MMC(\sigma_1'') = c_{M_w} \leq \lceil \frac{n}{m} \rceil \cdot a \cdot c_{\max}$$

Lemma 3 shows that

$$MMC(\sigma_1'') \leq \lceil \frac{n}{m} \rceil \cdot MMC(\pi)$$

If machine $M_{w'}$ is busy directly before s_z , let $B_{z'}$ be the last batch on machine M_w such that machine $M_{w'}$ is idle directly before $s_{z'}$. Similarly, suppose there are k batches between $B_{z'}$ and B_z , we denote them as B'_1, B'_2, \dots, B'_k . From the discussion of case 2 in $c_{ij} = a + c_j$ situation, such $B_{z'}$ must exist and is not the first batch on M_w . Denoting the total cost of batches starting before $s_{z'}$ on machine $M_{w'}$ is V'_2 ; thus, by the definition of $B_{z'}$ and $M_{w'}$, the number of batches starting before $s_{z'}$ on machine $M_{w'}$ is no more than $\lceil \frac{n}{m} \rceil - (k+2)$. Then,

$$V_{2}' \leq \sum_{1 \leq l \leq \lceil \frac{n}{m} \rceil - (k+2)} a \cdot c_{B_{l_{w'}}}$$

= $a \cdot \sum_{1 \leq l \leq \lceil \frac{n}{m} \rceil - (k+2)} c_{B_{l_{w'}}}$
 $\leq a \cdot [\lceil \frac{n}{m} \rceil - (k+2)] \cdot c_{\max}.$

Thus,

 $V_2' \le a \cdot \left[\left\lceil \frac{n}{m} \right\rceil - (k+2) \right] \cdot c_{\max}.$ (8)

Moreover, by algorithm H^1 , the total cost of batches starting before $s_{z'}$ on machine M_w is not greater than the total cost of batches starting before $s_{z'}$ on machine $M_{w'}$, so

$$c_{M_w} - (a \cdot c_{B_z}) - (a \cdot \sum_{1 \le l \le k} c_{B'_l}) - (a \cdot c_{B_{z'}}) \le V'_2.$$

Therefore, from (8), we get

$$c_{M_w} \leq V'_2 + a \cdot c_{B_{z'}} + a \cdot \sum_{1 \leq l \leq k} c_{B'_l} + a \cdot c_{B_z}$$

$$\leq a \cdot \left[\left\lceil \frac{n}{m} \right\rceil - (k+2) \right] \cdot c_{\max} + a \cdot c_{\max} + a \cdot k \cdot c_{\max} + a \cdot c_{\max}$$

$$= \left\lceil \frac{n}{m} \right\rceil \cdot a \cdot c_{\max}.$$

In addition, by (6), we have

$$MMC(\sigma_1'') = c_{M_w} \leq \lceil \frac{n}{m} \rceil \cdot (a \cdot c_{\max}).$$

Lemma 3 shows that

$$MMC(\sigma_1'') \leq \lceil \frac{n}{m} \rceil \cdot MMC(\pi)$$

We know that $k \ge 1$. When k = 0, similar to the above discussion, the conclusion also holds. Overall, for problem Pm|online, p - batch, $b = \infty$, $p_j = 1|Lex(C_{max}, MMC)$, when $c_{ij} = a + c_j$ or $c_{ij} = a \cdot c_j$, $1 \le i \le m$, $1 \le j \le n$, algorithm H^1 is a $(1 + \beta_m, \lceil \frac{n}{m} \rceil)$ -competitive online algorithm. Combining theorem 1, we obtain that algorithm H^1 is a best possible online algorithm. \Box

Lemma 4. In algorithm H^2 , there are at most $\lceil \frac{n}{m} \rceil$ batches on each machine.

Proof. When $n \le m$, there is at most one batch on each machine, so the conclusion holds naturally. When n > m, suppose, after the *k*th batch has been processed, that there are at most $\lceil \frac{k}{m} \rceil$ batches on each machine, and $m \le k \le n - 1$. In the following, we have an induction on *k*, to prove that, after the

(k+1)th batch has been processed, there are at most $\lceil \frac{k+1}{m} \rceil$ batches on each machine. The batches are denoted by B_1, B_2, \cdots , such that $s_1 < s_2 < \cdots$.

Case 1 k = qm + l, and $1 \le q \le \lceil \frac{n}{m} \rceil - 1$, $1 \le l \le m - 1$, where q, l are integers—as after the *k*th batch has been processed, there are k - qm = l machines in which their batch numbers are $\lceil \frac{k}{m} \rceil$, and other m - (k - qm) = m - l machines in which their batch numbers are $\lceil \frac{k}{m} \rceil - 1$. Let s_k be the release time of *k*th batch, and r_{k+1} be the latest release time of jobs in (k + 1)th batch. Then, we get $r_{k+1} > s_k$; otherwise, jobs in the (k + 1)th batch will process with jobs in the *k*th batch, a contradiction. Furthermore, by algorithm H^1 , we can get that the starting time of the (k + 1)th batch is $(1 + \beta_m)r_{k+1} + \beta_m$. In addition, because

$$\begin{aligned} (1+\beta_m)r_{k+1} + \beta_m &> (1+\beta_m)s_k + \beta_m \\ &= (1+\beta_m)^2 r_k + (1+\beta_m)\beta_m + \beta_m \\ &> (1+\beta_m)^2 s_{k-1} + (1+\beta_m)\beta_m + \beta_m \\ \vdots \\ &> (1+\beta_m)^m s_{k+1-m} + \sum_{i=0}^{m-1} \beta_m (1+\beta_m)^i \\ &= (1+\beta_m)^m s_{k+1-m} + (1+\beta_m)^m - 1 \\ &= \frac{\beta_m + 2}{\beta_m + 1} (s_{k+1-m} + 1) - 1 \\ &= s_{k+1-m} + \frac{s_{k+1-m} + 1}{\beta_m + 1} \\ &\geq s_{k+1-m} + 1, \end{aligned}$$

$$(1+\beta_m)r_{k+1}+\beta_m > s_{k+1-m}+1.$$
(9)

We use s_{k+1-m} to represent the starting time of the (k + 1 - m)th batch. From (9), it shows that, when the (k + 1)th batch starts, the (k + 1 - m)th batch has been completed. Then, lbatches that start before the (k + 1 - m)th batch also have been completed. We define these lbatches as $B_{k+1-m-l}, B_{k+1-m-(l-1)}, \dots, B_{k-m}$. Then, when the (k + 1)th batch starts, l + 1 batches $B_{k+1-m-l}, B_{k+1-m-(l-1)}, \dots, B_{k-m}, B_{k+1-m}$ have been completed. In addition, because of k = qm + l, when the (k + 1)th batch starts, there is at least one machine that is idle. In addition, it can be known, by algorithm H^2 , that the number of batches on this idle machine is $\lceil \frac{k}{m} \rceil - 1$. Hence, after the (k + 1)th batch is completed, the number of batches on this idle machine is $\lceil \frac{k}{m} \rceil - 1 + 1 = \lceil \frac{k}{m} \rceil$. Moreover, because $\lceil \frac{k}{m} \rceil = \lceil \frac{k+1}{m} \rceil$ when k = qm + l, there are k - qm + 1 = l + 1 machines whose batch numbers are $\lceil \frac{k+1}{m} \rceil$, and other m - (k - qm + 1) = m - l - 1 machines that have $\lceil \frac{k+1}{m} \rceil - 1$ batches. This means that there are at most $\lceil \frac{k+1}{m} \rceil$ batches on each machine. The result follows.

Case 2 k = qm and $1 \le q \le \lfloor \frac{n}{m} \rfloor - 1$, where *q* is an integer. After the (k + 1)th batch is processed, one machine has $\lfloor \frac{k}{m} \rfloor + 1$ batches, and the number of batches on other machines is still $\lfloor \frac{k}{m} \rfloor$. Furthermore, $\lfloor \frac{k}{m} \rfloor + 1 = \lfloor \frac{k+1}{m} \rfloor$ when k = qm. Thus, every machine has at most $\lfloor \frac{k+1}{m} \rfloor$ batches. The results follow. \Box

Theorem 3. For problem $Pm|online, p - batch, b = \infty, p_j = 1|Lex(C_{max}, MMC), when <math>c_{ij} = a + c_j$ or $c_{ij} = a \cdot c_j, 1 \le i \le m, 1 \le j \le n$, algorithm H^2 is the best possible online algorithm with a competitive ratio of $(1 + \beta_m, \lceil \frac{n}{m} \rceil)$.

Proof. When $c_{ij} = a + c_j$, suppose the schedule generated by algorithm H^2 is σ'_2 . From Lemma 1 and Lemma 2, we have

$$C_{\max}(\sigma_2') = (1 + \beta_m)r_{\max} + \beta_m + 1 = (1 + \beta_m)(r_{\max} + 1) \le (1 + \beta_m) \cdot C_{\max}(\pi)$$

In the following, we want to prove $MMC(\sigma'_2) \leq \lceil \frac{n}{m} \rceil \cdot MMC(\pi)$. Suppose machine M_x has the maximum machine cost, Then,

$$MMC(\sigma_2') = c_{M_r}.$$
(10)

From Lemma 4, we know that the number of batches on machine M_x is no more than $\lceil \frac{n}{m} \rceil$; then, from (10), we get

$$MMC(\sigma'_{2}) = c_{M_{x}}$$

$$\leq \lceil \frac{n}{m} \rceil \cdot a + \sum_{1 \leq l \leq \lceil \frac{n}{m} \rceil} c_{B_{l_{x}}}$$

$$\leq \lceil \frac{n}{m} \rceil \cdot a + \lceil \frac{n}{m} \rceil \cdot c_{\max}$$

$$= \lceil \frac{n}{m} \rceil \cdot (a + c_{\max}).$$

In addition, Lemma 3 shows that

$$MMC(\sigma_2') \leq \lceil \frac{n}{m} \rceil \cdot MMC(\pi).$$

When $c_{ij} = a \cdot c_j$, suppose the schedule produced by algorithm H^2 is σ_2'' . From Lemmas 1 and 2, we can get $C_{\max}(\sigma_2'') \leq (1 + \beta_m) \cdot C_{\max}(\pi)$ —the following to prove $MMC(\sigma_2'') \leq \lceil \frac{n}{m} \rceil \cdot MMC(\pi)$.

Assume that machine M_w is the machine with the maximum cost, Then,

$$MMC(\sigma_2'') = c_{M_w}.$$
(11)

From Lemma 4, we know that the number of batches on machine M_w is no more than $\lceil \frac{n}{m} \rceil$; then, from (11), we get

$$MMC(\sigma_2'') = c_{M_w}$$

$$\leq \sum_{1 \leq l \leq \lceil \frac{n}{m} \rceil} a \cdot c_{B_{l_w}}$$

$$\leq \lceil \frac{n}{m} \rceil \cdot (a \cdot c_{\max}).$$

From Lemma 3, we have

$$MMC(\sigma_2'') \leq \lceil \frac{n}{m} \rceil \cdot MMC(\pi).$$

To sum up, for problem $Pm|online, p - batch, b = \infty, p_j = 1|Lex(C_{max}, MMC))$, when $c_{ij} = a + c_j$ or $c_{ij} = a \cdot c_j, 1 \le i \le m, 1 \le j \le n$, algorithm H^2 is an online algorithm with a competitive ratio of $(1 + \beta_m, \lceil \frac{n}{m} \rceil)$.

Combining Theorem 1, it implies that algorithm H^2 is a best possible online algorithm. \Box

5. Conclusions

In this paper, we established two best possible online algorithms for problem $Pm|online, p-batch, b = \infty, p_j = 1|Lex(C_{max}, MMC)$, when $c_{ij} = a + c_j$ or $c_{ij} = a \cdot c_j$, $1 \le i \le m$, $1 \le j \le n$. The algorithms provided in this paper are to minimize the maximum machine cost subject to the makespan being at its minimum. They are extensions of the algorithm in [3], which is only minimizing the makespan. Here, we suppose that all machines have the same fixed cost *a*; for further research, extending this problem to different machine costs a_i is still an important research topic that needs to be studied.

Author Contributions: Conceptualization and methodology, W.L.; formal analysis and writing—original manuscript, W.Z. and X.C.; project management, supervision and writing-review, W.L. and X.C.

Funding: This Research was funded by the National Natural Science Foundation of China (Nos. 11571321, 11971443 and 11771406) and Postgraduate Education Reform Project of Henan Province (No. 2019SJGLX051Y).

Acknowledgments: We are grateful to the Associate Editor and four anonymous reviewers for their valuable comments, which helped us significantly improve the quality of our paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Uzsoy, R.; Lee, C.Y.; Martin-Vega, L.A. A review of production planning and scheduling models in the semiconductor industry, part I: System characteristics, performance evaluation and production planning. *IIE Trans.* 1992, 24, 47–61. [CrossRef]
- 2. Uzsoy, R.; Lee, C.Y.; Martin-Vega, L.A. A survey of production planning and scheduling models in the semiconductor industry, part II: Shop-floor control. *IIE Trans.* **1994**, *26*, 44–55. [CrossRef]
- 3. Zhang, G.; Cai, X.; Wong, C.K. Optimal online algorithms for scheduling on parallel batch processing machines. *IIE Trans.* **2003**, *35*, 175–181. [CrossRef]
- 4. Tian, J.; Cheng, T.C.E.; Ng, C.T.; Yuan, J. Online scheduling on unbounded parallel-batch machines to minimize the makespan. *Inf. Process. Lett.* **2009**, *109*, 1211–1215. [CrossRef]
- 5. Liu, P.; Lu, X.; Fang, Y. A best possible deterministic online algorithm for minimizing makespan on parallel batch machines. *J. Sched.* **2012**, *15*, 77–81. [CrossRef]
- 6. Fang, Y.; Liu, P.; Lu, X. Optimal online algorithms for one batch machine with grouped processing times. *J. Comb. Optim.* **2011**, *22*, 509–516. [CrossRef]
- 7. Li, W. A best possible online algorithm for the parallel-machine scheduling to minimize the maximum weighted completion time. *Asia-Pac. J. Oper. Res.* **2015**. [CrossRef]
- 8. Cao, J.; Yuan, J.; Li, W.; Bu, H. Online scheduling on batching machines to minimise the total weighted completion time of jobs with precedence constraints and identical processing times. *Int. J. Syst. Sci.* **2011**, *42*, 51–55. [CrossRef]
- 9. Brucker, P.; Gladky, A.; Hoogeveen, H.; Kovalyov, M.Y.; Potts, C.N.; Tautenhahn, T.; van de Velde, S.L. Scheduling a batching machine. *J. Sched.* **1998**, *1*, 31–54. [CrossRef]
- 10. Lee, C.Y.; Uzsoy, R. Minimizing makespan on a single batch processing machine with dynamic job arrivals. *Int. J. Prod. Res.* **1999**, *37*, 219–236. [CrossRef]
- 11. Liu, Z.H.; Yu, W.C. Scheduling one batch process or subject to job release dates. *Discrete Appl. Math.* 2000, 105, 129–136. [CrossRef]
- 12. Li, W.; Yuan, J.; Lin, Y. A note on special optimal batching structures to minimize total weighted completion time. *J. Comb. Optim.* **2007**, *14*, 475–480. [CrossRef]
- 13. Tian, J.; Fu, R.; Yuan, J. Online over time scheduling on parallel-batch machines: A survey. *J. Oper. Res. Soc. China* **2014**, *2*, 445–454. [CrossRef]
- 14. Li, W.; Chai, X. The medical laboratory scheduling for weighted flow-time. *J. Comb. Optim.* **2019**, *37*, 83–94. [CrossRef]
- 15. Ma, R.; Yuan, J. Online trade-off scheduling on a single machine to minimize makespan and total weighted completion time. *Int. J. Prod. Econ.* **2014**, *158*, 114–119. [CrossRef]
- 16. Liu, Q.; Yuan, J. Online trade-off scheduling on a single machine to minimize makespan and maximum lateness. *J. Comb. Optim.* **2016**, *32*, 385–395. [CrossRef]
- 17. Lee, K.; Leung, J.Y.T.; Jia, Z.; Li, W.; Pinedo, M.L.; Lin, B.M.T. Fast approximation algorithms for bi-criteria scheduling with machine assignment costs. *Eur. J. Oper. Res.* **2014**, *238*, 54–64. [CrossRef]



 \odot 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).