On-Line Scheduling on a Single Bounded Batch Processing Machine with Restarts*

Hua Chen
School of Computer and Communication, Lanzhou University of Technology, Lanzhou, 730050, Gansu, P. R. China
hua_0110@163.com

Yuan-ping Zhang
School of Computer and Communication, Lanzhou University of Technology, Lanzhou, 730050, Gansu, P. R. China
ypzhang@lut.cn

Xue-rong Yong
Department of Mathematics, University of Puerto Rico at Mayaguez, P.O.Box 9018, PR 00681, USA
xryong@math.uprm.edu

Abstract

The problem of on-line job scheduling on a single batch processing machine with bounded batch size is attacked. Jobs arrive dynamically over time, the characteristics of jobs are unknown before their arrivals. A batch processing machine can handle up to b jobs simultaneously. The processing time of a batch is given by the longest processing time of some job in the batch. The objective of on-line scheduling problem is to minimize the time makespan. An on-line algorithm with restarts for a bounded batch machine scheduling is presented, and it is shown that the competitive ratio is 3/2. This model has not been considered previously.

Key words: *on-line; scheduling; bounded; batch machine; restarts; makespan.*

Ⅰ. Introduction

On-line scheduling in batch processing system has been extensively studied in the recent decades [10]. On-line means that jobs arrive over time, the characteristics of jobs are known after their arrivals. The makespan of each job is the time interval that the machine used for processing it. The objective of the on-line scheduling in processing system is to minimize the total processing time of all accepted jobs. The spare time of the machine waiting jobs to arrive is not counted.

Since it is more effective to process jobs in batches than in individual, so most processing systems are in batched. A survey on scheduling problems with batching is given in [11]. The starting time and completion time of jobs in the same batch are equal. The processing time of a batch is given by the longest processing time of the jobs in the batch.

Such a model of batch processing machine is motivated by testing the thermal standing abilities of semiconductors with burn-in ovens [7, 15]. In details, there is an oven with limited capacity, a batch of integrated circuits (jobs) may be put inside to test for their thermal standing abilities. The circuits are heated until all circuits are burned. The burn-in times of circuits (job processing times) may be different. When a circuit is burned, it has to wait inside the oven until all other circuits are burned even it is the earliest burned one. Therefore, the processing time of a batch of circuits (jobs) is the longest processing time of the circuits (jobs) in the batch.

An on-line algorithm is always measured by competitive ratio [1]. Let $C^*(L)$ and $C_H(L)$ denote, respectively, the sum of makespan given by off-line optimal algorithm and on-line approximation algorithm H, where L is the schedule job list. The competitive ratio of on-line algorithm H is defined as $\sup_{\forall L}\{C_H(L)/C^*(L)\}$.

The on-line batch processing system can contain a single machine, two machines or more than two

* This work was supported by the National Science Foundation of Gansu province, P.R.C..(Grant No. 3ZS051-A25-037).

978-0-7695-3557-9/09 $25.00 © 2009 IEEE
DOI 10.1109/ETCS.2009.196

machines. All these situations have been concerned about by some researchers. The cases of two machines or more than two machines have been studied in [2, 6, 14]. The research in this paper will focus on the problem of on-line scheduling on a single batch processing machine.

Until now, all the research works of a single batch processing machine scheduling have considered two cases: bounded batch size model and unbounded batch size model. As the batch size is bounded, if there is no constraint on arrival time, a 2-competitive algorithm is described in [8, 12, 16]. If there are only two distinct arrival times, an algorithm with possible optimal competitive ratio $(\sqrt{5}+1)/2$ is achieved [16]. Moreover, for this bounded batch processing problem, any algorithm based on Full-Batch Longest Processing Time can achieve 2-competitive, especially, for machine with batch precessing size 2, there is an on-line algorithm with competitive ratio 7/4 [12].

The unbounded model of batching machine has been considered also. In such model, the batch size is supposed sufficiently large. A deterministic greedy heuristic algorithm with no idle-times in the schedule is proposed, and it is shown that the competitive ratio has lower bound 2 and adopting the simple rule based on Longest Processing Time can achieve this bound [9]. If the starts of jobs are allowed to be postponed, an on-line algorithm with ratio $(\sqrt{5}+1)/2$ is given, and the ratio 2 is also obtained by extending the algorithm to general case, *i.e.*, the idle-times and the postponement of start times in the schedule may be allowed or not [16]. A much simpler proof for the result in [16] is provided in [13]. The on-line algorithm with $(\sqrt{5}+1)/2$-competitive ratio is independently provided and it is proved best possible in [3].

Recently, one new related model of on-line processing system is proposed in which the running jobs can be interrupted to let the jobs be re-scheduled. It is called the model with restarts. For a single unbounded batch size processing system with this model, a linear on-line scheduling algorithm with competitive ratio 3/2 is provided, and the lower bound is $(5-\sqrt{5})/2$ [4]. If a job which has already been restarted once cannot be restarted any more, another best possible algorithm with competitive ratio 3/2 for on-line scheduling on a single batch machine is given [5]. However, these results are all in the unbounded batch size model, and there is no research using restarts in a bounded batch size processing system. But, the machine capacity is always limited in practice. So this model will be considered in this paper and a related algorithm will be provided. The competitive ratio of this algorithm is 3/2, and this competitive ratio is proved to be optimal.

Ⅱ. An on-line algorithm

In this section, an on-line algorithm is developed for solving the on-line scheduling on a single batch processing machine problem, where the batch size is bounded. All jobs arrive over time. The characteristics of a job are unknown before its arrival time. For example, the processing time P_i and arrival time r_i of a job J_i are known once it arrives. A parallel batch processing machine can process up to b jobs simultaneously. Jobs in a batch have the same processing time, which is equal to the longest processing time of the jobs. Restarts of a job means that a running job can be interrupted to let the job be re-scheduled. Restarts of a batch is that all jobs in it are released and become independent unscheduled jobs. If a job is re-scheduled, all the work done on it previously will lose. The strategy of restarts can reduce the impact of a wrong decision and make a possible better scheduling in time.

Generally, a full processing batch is not allowed any interruption until it is completed. Some used parameters are defined as follows.

t: free time of the machine during the scheduling
$U(t)$: set of jobs unscheduled at time t
b: batch size of a bounded batch processing machine
m: the number of arrived jobs waiting for scheduling
B_k: present processing batch
J_k: longest processing time job in B_k
J_l: latest one of jobs with longest processing time in present unfull batch
J_n: new arrived job
P_l: P_k, P_n and r_l, r_k, r_n are the processing times and arrival times of J_l, J_k, J_n respectively

Algorithm *H*
Step 1: The machine starts to work at time t.
Step 2: Repeat following 4 sub-steps:
Step 2.1: $m=|U(t)|$.
Step 2.2: If $m=0$, go to Step 5.
Step 2.3: If $0<m<b$, schedule these m jobs as an unfull batch, and find J_l in it, go to Step 3.
Step 2.4: If $m \geqslant b$, arrange all the jobs in decreasing order according to their processing

times, and schedule b longest jobs as a first batch B_k to process. $t=t+P_k$.

Step 3: In time interval $[t, t+P_l)$, if no new job arrives, set $t=t+P_l$, go to Step 5.

Step 4: If a new job J_n arrives at time $r_n<t+P_l$, then do the following.

Step 4.1: If $P_n < P_l$ and $P_n < \max\{\frac{1}{2}P_l, r_n\}$, go on processing the present batch and then go to Step 2.

Step 4.2: If either $P_n \geq P_l$ or $P_l > P_n \geq \max\{\frac{1}{2}P_l, r_n\}$ restart the present batch, reset $t=r_n$ and go to Step 2.

Step 5: If there are still some jobs arriving, set t as the arrival time of the first job and go to Step 2, otherwise stop and complete the whole on-line schedule at time t.

The following example illustrates how algorithm H works. A job J_i is defined as $<r_i, P_i>$. Same as the above definition, r_i is the arrival time of job J_i, P_i is the processing time of job J_i. There are five jobs $J_1 \ldots J_5$ are given as follows, <0, 2>, <1, 3>, <2, 2>, <3, 1>, <4, 2>. Assume that the batch capacity is 2. At time $r_1=0$, only one job J_1 with $P_1=2$ is available, schedule J_1 as a batch to process first. In time interval [0, 2), J_2 arrives and $P_2>P_1$, therefore, restarts the running batch at time $r_2=1$. A first full batch is formed. The processing time of this batch is equal to the longest job processing time $P_2=3$. This batch completes at time 4. At this time, jobs J_3、J_4、J_5 are all available. The machine will choose two longest processing time jobs J_3 and J_5 as a second batch. J_4 as a third unfull batch follows to wait for other new jobs coming. The schedule is demonstrated in Figure 1.

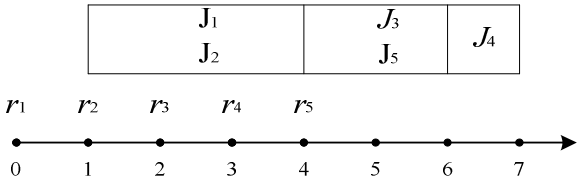

Figure 1

Ⅲ. Competitive analysis

Now we show that the competitive ratio of the above on-line scheduling algorithm H is 3/2.

In off-line scheduling, in order to minimize the makespan, jobs have the near processing time will be inclined to be scheduled together as batches. According to algorithm H, if there are not less than b jobs for scheduling at time t, we use greedy strategy to schedule b longest jobs as batches to process first, it is equal to off-line optimal and will not be discussed any more. So we only need to consider the case of the on-line scheduling for unfull batch.

Lemma 1: If $P_n < P_l$ and $P_n < \max\{\frac{1}{2}P_l, r_n\}$, then $C_H(L)/C^*(L) \leq 3/2$.

Proof: Let s be the starting time of the present unfull batch B_k. If $P_n < P_l$ and $P_n < \max\{\frac{1}{2}P_l, r_n\}$, the algorithm H will go on processing the present batch. Then $C_H(L)=s+P_l+P_n$. So $C^*(L) \geq s+P_l \geq P_l$. From $P_n < \max\{\frac{1}{2}P_l, r_n\}$, it is seen $P_n < \frac{1}{2}P_l$ or $P_n < r_n$. If $P_n < \frac{1}{2}P_l$, then $(C_H(L)-C^*(L))/C^*(L) \leq P_n/P_l < \frac{1}{2}$. If $P_n < r_n$, it is also obtained that $C^*(L) \geqslant r_n + P_n$, then $(C_H(L)-C^*(L))/C^*(L) \leq P_n/(r_n+P_n) < r_n/(2r_n) = \frac{1}{2}$. Therefore, we have $C_H(L)/C^*(L) \leq 3/2$.

Lemma 2: If $P_n \geq P_l$ or $P_l > P_n \geq \max\{\frac{1}{2}P_l, r_n\}$, then the on-line processing is optimal.

Proof: By algorithm H, with $P_n \geq P_l$ or $P_l > P_n \geq \max\{\frac{1}{2}P_l, r_n\}$, the present unfull processing batch will be restarted. If $P_n \geq P_l$, then $C_H(L)= r_n + P_n$, $C^*(L) \geq r_n + P_n$, and $C_H(L)=C^*(L)$. Hence, it is optimal. If $P_l > P_n \geq \max\{\frac{1}{2}P_l, r_n\}$, then $C_H(L)= r_n + P_l$, $C^*(L) \geq r_n + P_l$, and $C_H(L)=C^*(L)$. So it is also optimal.

Combining Lemma 1 and lemma 2, the following result can be established.

Theorem 3: The on-line schedule algorithm H has competitive ratio 3/2.

We can continuously prove the following:

Theorem 4: The competitive ratio of algorithm H is no more than 3/2.

Proof: Let ε be a sufficiently small positive number. Assume that b=2, the machine is free now. One job J_1 with processing time $P_1=\frac{1}{2}$ arrives at some time t, J_1 is scheduled as a single batch $\{J_1\}$ starting at time t to process first. Job J_2 with processing time P_2=1 arrives at time $t+\frac{1}{4}-\varepsilon$. Since $P_2>P_1$, J_1 will be rescheduled at time $t+\frac{1}{4}-\varepsilon$, J_1 and J_2 will be processed as a full batch $\{J_1, J_2\}$ without interruption. Before it is completed, job J_3 with processing time P_3=1 arrives. Then $C_H(L)=\frac{9}{4}-\varepsilon$ and $C^*(L)=\frac{3}{2}$. Therefore, $C_H(L)/C^*(L)=(\frac{9}{4}-\varepsilon)/\frac{3}{2}\leqslant\frac{3}{2}$, as $\varepsilon\to 0$.

Theorems 3 and 4 imply the following main result:

Theorem 5: The on-line schedule algorithm H has exact competitive ratio 3/2.

Ⅳ. Conclusion

There are still many interesting questions about on-line scheduling problem. A lot of questions may be worthy to be considered, for example, a batch processing machine with no constraint on the number of jobs at each arrival time; a single batch processing machine or parallel batch processing machines with rejection penalty, delivery time and family partition etc.. The corresponding on-line algorithms with appropriate competitive ratios are inadequate for the moment.

References

[1] A. Borodin and R.El-Yaniv, "Online computation and competitive analysis", Cambridge University Press, 1998.

[2] W. T. Chan, F. Y. L. Chin, D. Ye, G. Zhang and Y. Zhang, "On-line scheduling of parallel jobs on two machines", Journal of Discrete Algorithm 6, 2008, pp.3-10.

[3] X. Deng, C.K. Poon and Y. Zhang,"Approximation algorithms in batch processing", Journal of Combinatorial Optimization 7, 2003, pp.247-257.

[4] R. Fu, J. Tian, J. Yuan and Y. Lin, "On-line scheduling in a parallel batch processing system to minimize makespan using restarts", Theoretical Compute Science 374, 2007, pp.196-202.

[5] R. Fu, J. Tian, J. Yuan and C. He, "On-line scheduling on a batch machine to minimize makespan with limited restarts", Operations Research Letters36, 2008, pp.255-258.

[6] J. L. Hurink and J. J. Paulus, "On-line scheduling of parallel jobs on two machines is 2-competitive", Operations Research Letters 36, 2008, pp.51-56.

[7] C.Y. Lee, R. Uzsoy and L.A. Martin-Vega, "Efficient algorithms for scheduling semiconductor burn-in operations", Operations Research 40, 1992, pp.219-236.

[8] C.Y. Lee and R. Uzsoy, "Minimizing makespan on a single batch processing machine with dynamic job arrivals", International Journal of Production Research 37, 1999, pp.219-236.

[9] Z. Liu and W. Yu, "Scheduling one batch processor subject to job release dates", Discrete Applied Mathematics 105, 2000, pp.129-136.

[10] Q. Nong, J. Yuan, R. Fu, L. Lin and J. Tian, "The single-machine paraller-batching on-line scheduling problem with family jobs to minimize makespan", International Journal of Production Economics 111, 2008, pp.435-440.

[11] C.H. Potts and M.Y. Kovalyov, "Scheduling with batching: A review", European Journal of Operational Research 120, 2000, pp. 228-249.

[12] C.K. Poon and W. Yu, "On-line scheduling algorithms for a batch machine with finite capacity", Journal of Combinatorial Optimization 9, 2005, pp.167-186.

[13] F. Ridouard, P. Richard and P. Martineau, "On-line scheduling on a batch processing machine with unbounded batch size to minimize the makespan", European Journal of Operational Research 189, 2008, pp.1327-1342.

[14] Z.Tian and S. Yu, "On-line scheduling with reassignment", Operations Research Letters 36, 2008, pp.250-254

[15] R. Uzsoy, C.Y. Lee and L.A. Martin-Vega, "A review of production planning and scheduling models in the semiconductor industry, part i:system characteristics, performance evaluation and production planning", IEE Transactions on Scheduling and Logistics 26, 1992, pp.44-55.

[16] G. Zhang, X. Cai, and C.K.Wong, "On-line algorithms for minimizing makespan on batch processing machines," Naval Research Logistics 48, 2001, pp.241-258.