Online Scheduling of Linear Deteriorating Jobs on Parallel Machines

Sheng Yu* Prudence W. H. Wong (Speaker)† Yinfeng Xu‡

1 Introduction

Scheduling a set of jobs with fixed processing times is a classical problem [13]. Yet, there are numerous situations that the processing time increases (deteriorates) as the start time increases, e.g., maintenance or cleaning schedule, fire fighting, steel production and financial management [10, 11]. The study of minimizing makespan of deteriorating jobs first focused on a single machine, with linear deterioration [2] and non-linear deterioration [7]. Since then, the problem has attracted a lot of attention (see e.g., [1, 4, 6]).

A job J_j with *linear deterioration* has a processing time $p_j = a_j + b_j s_j$, where $a_j \geq 0$ is the "normal" processing time, $b_j > 0$ is the deteriorating rate, and s_j is the start time. Linear deterioration is said to be *simple* if $a_j = 0$, i.e., $p_j = b_j s_j$. In this case, to avoid trivial solution, it is natural to assume that the start time of the first job is $t_0 > 0$ since a start time of zero implies that the processing time of all jobs is zero.

It is observed in [7] that, when all jobs are available at time 0, $1 \,|\, p_j = a_j + b_j s_j \,|\, C_{\max}$ can be solved optimally by scheduling in ascending order of a_j/b_j. For $1 \,|\, p_j = b_j s_j \,|\, C_{\max}$, the makespan is independent of the job processing order [11]. The problem becomes NP-hard and strongly NP-hard on two machines and m machines, respectively (c.f. the complexity when $p_j = a_j$ [5]). The problem $P2 \,|\, p_j = b_j s_j \,|\, C_{\max}$ is NP-hard [9, 12]. FPTAS have been proposed for $Pm \,|\, p_j = a_j + b_j s_j \,|\, C_{\max}$ [8] and $Pm \,|\, p_j = b_j s_j \,|\, C_{\max}$ [14].

Online algorithms have only been studied for $Pm \,|\, p_j = b_j s_j, \text{online-list} \,|\, C_{\max}$ [3], in which a job has to be scheduled before the next job can be seen. They showed that the competitive ratio of List Scheduling (LS) is $(1 + b_{\max})^{\frac{m-1}{m}}$ and this is the best possible.

Our contributions. We first consider $Pm \,|\, p_j = a_j + bs_j, \text{online-list} \,|\, C_{\max}$ and show that Round Robin (RR) is α-competitive and no on-line algorithm is better than $(2 - \frac{1}{\alpha})$-competitive, where $\alpha = \frac{a_{\max}}{a_{\min}}$. Furthermore, when jobs have release times, we study a class of "reasonable" algorithms including LS that do not idle when there are available jobs. For $Pm \,|\, p_j = b_j s_j, r_j, \text{online} \,|\, C_{\max}$, we show that these algorithms achieve a competitive ratio of $(1 + b_{\max})^{2(1-\frac{1}{m})}$ and no online algorithm is better than $(1 + b_{\max})^{\frac{1}{2}}$. We also show that the competitive ratio of Round Robin (RR) can be unbounded.

Notations and problem definition. We are to schedule non-preemptively a set of jobs $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$ onto machines $M_1, M_2, \ldots, M_m$. For J_j, we denote by r_j and p_j the release and processing time, respectively,. where $p_j = a_j + b_j s_j$ and s_j is the start

*School of Management, Xi'an Jiaotong University, a.sheng@stu.xjtu.edu.cn
†Department of Computer Science, University of Liverpool, pwong@liverpool.ac.uk
‡School of Management, Xi'an Jiaotong University, yfxu@mail.xjtu.edu.cn

time of J_j. Let $b_{\max} = \max_{1\le j\le n}\{b_j\}$, $a_{\max} = \max_{1\le j\le n}\{a_j\}$, $a_{\min} = \min_{1\le j\le n}\{a_j\}$, and $\alpha = \frac{a_{\max}}{a_{\min}}$. Consider a schedule S. For $1 \le j \le n$, the completion time of job J_j in S is denoted by $c_j(S)$. For any $1 \le k \le m$, the number of jobs and the set of jobs dispatched to M_k by S is denoted by $n^k(S)$ and $\mathcal{J}^k(S)$. The makespan of machine M_k and schedule S are denoted by $C^k_{\max}(S)$ and $C_{\max}(S)$, respectively. The objective of the problem is to minimize the makespan of the schedule produced.

2 Varying deteriorating rates $p_j = b_j s_j$

We consider the problem $Pm \,|\, p_j = b_j s_j, r_j, \text{online} \,|\, C_{\max}$, i.e., $a_j = 0$. In this setting, for any job J_j, $c_j = s_j + b_j s_j = s_j(1 + b_j)$. We first show some lower bounds (Theorem 1) (note that (ii) still holds even under the condition without release times). Then we study the performance of "reasonable" online algorithms (RA) that do not idle when there are available jobs. LS is reasonable but RR is not. We observe that if a RA schedule does not idle at all, its makespan is bounded (Lemma 2) and then we extend the idea and show that RA is $(1 + b_{\max})^{2(1-\frac{1}{m})}$-competitive (Theorem 3).

Theorem 1. *For* $Pm \,|\, p_j = b_j s_j, r_j, \text{online} \,|\, C_{\max}$, (i) *no deterministic online algorithm is better than* $(1 + b_{\max})^{\frac{1}{2}}$*-competitive;* (ii) *the competitive ratio of* RR *is unbounded.*

Lemma 2. *If in the* RA *schedule, every machine is not idle at all, then* $\frac{C_{\max}(\text{RA})}{C_{\max}(\text{OPT})} \le (1 + b_{\max})^{1-\frac{1}{m}}$.

Proof. (Sketch.) For simplicity, we assume the first m jobs have $r_j = 1$. Let $\mathcal{J}^k(\text{OPT})$ be the set of jobs dispatched to M_k by the optimal schedule OPT. Then, $C^k_{\max}(\text{OPT}) \ge \prod_{j:J_j\in\mathcal{J}^k(\text{OPT})}(1 + b_j)$, and thus, $C_{\max}(\text{OPT}) \ge (\prod_{j:J_j\in\mathcal{J}}(1 + b_j))^{\frac{1}{m}}$. Let ℓ be the index of a job with completion time equals to $C_{\max}(\text{RA})$ and p be the machine M_p to which RA dispatches J_ℓ. As there is no idle time in RA schedule $s_\ell(\text{RA}) \le (\prod_{j:J_j\in\mathcal{J}-\{J_\ell\}}(1+b_j))^{\frac{1}{m}}$. As a result, $C_{\max}(\text{RA}) = s_\ell(\text{RA})(1 + b_\ell) \le C_{\max}(\text{OPT})(1 + b_{\max})^{1-\frac{1}{m}}$. □

Theorem 3. *For* $Pm \,|\, p_j = b_j s_j, r_j, \text{online} \,|\, C_{\max}$, RA *is* $(1 + b_{\max})^{2(1-\frac{1}{m})}$*-competitive.*

3 Fixed deteriorating rate $p_j = a_j + b\ s_j$

In the online-list model, when a job is given, the online algorithm has to dispatch the job to a machine and specify the period of time to process the job before the next job is given. Suppose $\mathcal{J}^k(S) = \{J_{k,1}, J_{k,2}, \cdots, J_{k,n^k}\}$. The completion time of the job $J_{k,j}$ is $c_{k,j} = \sum_{1\le i\le j} a_{k,i}(1 + b)^{j-i}$. Therefore, on a particular machine, the optimal schedule is to schedule jobs in increasing order of normal processing time a_j. We give some lower bounds (Theorem 4) and an upper bound of RR (Theorem 5).

Theorem 4. *Consider* $Pm \,|\, p_j = a_j + bs_j, \text{online-list} \,|\, C_{\max}$. (i) *No online algorithm is no better than* $(2-\frac{1}{\alpha})$*-competitive. (ii) Both* LS *and* RR *are no better than* α*-competitive.*

Proof. (Sketch.) (i) The adversary releases m jobs with the same normal processing time a. If an algorithm dispatches any two of these jobs to the same machine, the adversary stops releasing jobs. Otherwise, the adversary releases the $(m + 1)$-th job with normal processing time being $a(2 + b)$. In either case, the ratio is no better than $(2 - \frac{1}{\alpha})$.

(ii) We release $2qm$ jobs for some positive integer q. The job list contains qm jobs with $a_{\max}$ followed by those qm jobs with $a_{\min}$. Both LS and RR assign for each machine q jobs with $a_{\max}$ followed by q jobs with $a_{\min}$ while the optimal algorithm schedules the jobs with $a_{\min}$ first. The ratio is arbitrarily close to α by setting q as a large integer. □

Theorem 5. RR *is at most* α*-competitive for* $Pm \,|\, p_j = a_j + bs_j, \text{online-list} \,|\, C_{\max}$.

Proof. (Sketch.) On a single machine, no matter what order the jobs are scheduled, the makespan is no more than α times the optimal as long as the machine is not idle. We further observe that in any schedule, the maximum number of jobs on a machine is at least $\lceil \frac{n}{m} \rceil$ while in RR, the maximum number is at most $\lceil \frac{n}{m} \rceil$. Then we can show that RR is α-competitive. □

References

[1] B. Alidaee and N. K. Womer. Scheduling with time dependent processing times: Review and extensions. *Journal of the Operational Research Society*, 50(7):711–720, 1999.

[2] S. Browne and U. Yechiali. Scheduling deteriorating jobs on a single processor. *Operations Research*, 38(3):495–498, 1990.

[3] M. B. Cheng and S. J. Sun. A heuristic MBLS algorithm for the two semionline parallel machine scheduling problems with deterioration jobs. *Journal of Shanghai University*, 11(5):451–456, 2007.

[4] T. C. E. Cheng, Q. Ding, and B. M. T. Lin. A concise survey of scheduling with time-dependent processing times. *European Journal of Operational Research*, 152:1–13, 2004.

[5] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, 1979.

[6] S. Gawiejnowicz. *Time-Dependent Scheduling*. Springer-Verlag, Berlin, 2008.

[7] J. N. D. Gupta and S. K. Gupta. Single facility scheduling with nonlinear processing times. *Computers and Industrial Engineering*, 14(4):387–393, 1988.

[8] L. Y. Kang and C. T. Ng. A note on a fully polynomial-time approximation scheme for parallel-machine scheduling with deteriorating jobs. *International Journal of Production Economics*, 109:108–184, 2007.

[9] A. Kononov. Scheduling problems with linear increasing processing times. In e. a. Zimmermann U, editor, *Operations Research Proceedings 1996. Selected Papers of the Symposium on Operations Research (SOR 96)*, pages 208–212, Berlin, 1997. Springer.

[10] A. S. Kunnathur and S. K. Gupta. Minimizing the makespan with late start penalties added to processing times in a single facility scheduling problem. *European Journal of Operation Research*, 47(1):56–64, 1990.

[11] G. Mosheiov. Scheduling jobs under simple linear deterioration. *Computers and Operations Research*, 21(6):653–659, 1994.

[12] G. Mosheiov. Multi-machine scheduling with linear deterioration. *INFOR: Information Systems and Operational Research*, 36(4):205–214, 1998.

[13] M. Pinedo. *Scheduling: Theory, Algorithms, and Systems*. Prentice-Hall, Upper Saddle River, 2002.

[14] C. R. Ren and L. Y. Kang. An approximation algorithm for parallel machine scheduling with simple linear deterioration. *Journal of Shanghai University*, 11(4):351–354, 2007.