Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

ELSEVIER

Information Processing Letters 87 (2003) 89–93

Information Processing Letters

www.elsevier.com/locate/ipl

Scheduling linear deteriorating jobs to minimize makespan with an availability constraint on a single machine

Chin-Chia Wu *, Wen-Chiung Lee

Department of Statistics, Feng Chia University, Taichung 407, Taiwan, R.O.C.

Received 9 August 2002; received in revised form 10 February 2003

Communicated by M. Yamashita

Abstract

The scheduling problem with deteriorating jobs to minimize the makespan on a single machine where the facility has an availability constraint is studied in this paper. By a deteriorating job we mean that the processing time for the job is a function of its starting time. Even with the introduction of the availability to a facility, the linear deteriorating model can be solved using the 0–1 integer programming technique if the actual job processing time is proportional to the starting time.

Keywords: Scheduling; Single machine; Makespan; Deteriorating job; Availability constraint

1. Introduction

For most scheduling problems job processing times are treated as certain values (Conway et al. [3], Pinedo [12]), but that is not proper for all actual situations. Job processing times are not constant because jobs deteriorate as they wait to be processed. Kunnathur and Gupta [8] and Mosheiov [11] pointed out several real-life situations where deteriorating jobs might occur. These include shops with deteriorating machines, and/or delay of maintenance or cleaning, fire fighting, hospital emergency wards and steel rolling mills. This problem is also known as the deteriorating job scheduling problem. However, researches on the deteriorating job scheduling problem assume that the machines are available simultaneously at all times. This availability may not be true in real production settings. This situation occurs often in industry due to machine breakdowns (stochastic) or preventive maintenance (deterministic) during the scheduling period. In this paper we study the deteriorating job scheduling problems with an availability constraint for the deterministic case. To the best of our knowledge this type of problem has never been discussed.

Browne and Yechiali [2] and Gupta and Gupta [5] were the pioneers in considering the deteriorating job scheduling problem. They focused on minimizing the makespan in the single-machine scheduling problem under linear deterioration conditions. Since then there has been growing interest in scheduling problems with time-dependent processing times. Alidaee and Womer [1] provided an extensive survey of different models and problems where job processing time depended on the starting time. Kunnathur and

* Corresponding author.
E-mail address: cchwu@fcu.edu.tw (C.-C. Wu).

0020-0190/03/$ – see front matter
doi:10.1016/S0020-0190(03)00262-X

Gupta [8] studied a single-machine scheduling problem involving minimizing the makespan where the processing time for a job consisted of a fixed and a variable part. The variable part depended on the starting time for the job. They proposed two algorithms based on the dynamic programming and branch-and-bound techniques to obtain the optimal solution for the problem. They also proposed five heuristic rules to find the near-optimal solution. Mosheiov [10] considered a simple linear deterioration where jobs have a fixed job-dependent growth rate but no basic processing time. He showed that most commonly used performance, such as the makespan, the total flow time, the total lateness, the sum of the weighted completion times, the maximum lateness, the maximum tardiness and the number of tardy jobs, remain polynomially solvable if jobs have a fixed job-dependent growth rate but no basic processing time. Mosheiov [11] further studied the problem of the makespan minimization of stepwise deteriorating jobs. He introduced the compact integer programming formulations and proposed the heuristic algorithms for three types of NP-complete problems. Hsieh and Bricker [6] investigated two deteriorating scheduling models for multiple machines with the goal to minimize the makespan. They proposed an efficient heuristic algorithm for the first model and proved that the ratio of the makespan obtained by the heuristic algorithm to the optimal makespan was bounded. They also proposed three heuristic algorithms for the second model and indicated that these algorithms provided quite good solutions using a complete enumeration. Kubiak and van de Velde [7] established the computational complexity of the makespan minimization problem for a linear deterioration with a single common critical date. They showed that the problem was NP-hard and developed pseudo-polynomial dynamic programming algorithms for it.

Graves and Lee [4] pointed out that machine scheduling with an availability constraint is very important but still relatively unexplored. They assumed that maintenance needed to be performed within a fixed period and besides the job scheduling, the time for the maintenance was also a decision variable. They studied the problem under two scenarios concerning the planning horizon and under the objective functions of minimizing the total weighted completion time and the maximum lateness, respectively. Lee [9] provided an extensive study for the single and parallel machine scheduling problem with an availability constraint under various performance measures. In each case, he either provided a polynomial optimal algorithm to solve the problem, or proved that the problem was NP-hard. In the latter case he developed pseudo-polynomial dynamic programming models to solve the problem optimally and/or provided heuristics with an error bound analysis.

The remainder of this paper is organized as follows. In Section 2, first we define some notations and provide the formulation of the problem. In Section 3, we transform the problem into linear programming problems and describe some situations in which the problem can be further simplified. The conclusion and our future research topics are presented in the last section.

2. Notation and formation

Let $N = \{J_1, J_2, \dots, J_n\}$ be the set of jobs to be scheduled. The notations P_i, α_i, and C_i denote the actual processing time, the growth rate and the completion time of J_i, respectively. The symbol [ ] is used to signify the order of jobs in a sequence. The problem considered in this paper assumed that all deteriorating jobs with no fixed processing times are available for processing at time t_0. That is $P_i = \alpha_i t$ where t is the starting time for J_i. We assumed that the machine is not available from the period b_1 to b_2, where $0 \leqslant b_1 \leqslant b_2$.

For simplicity, resumable availability constraint is considered in this paper. We call a job resumable if it cannot finish before the unavailable period of a machine and can continue to be processed after the machine becomes available again. The objective of this paper is to find an optimal schedule π^* such that

$$\max\{C_1(\pi^*), C_2(\pi^*), \dots, C_n(\pi^*)\} \leqslant \max\{C_1(\pi), C_2(\pi), \dots, C_n(\pi)\}$$

for any schedule π.

3. Results

In this section we derive a way to solve this problem using the 0–1 integer programming technique. The simple linear deterioration is considered now. The actual processing time for the job scheduled in the first position in a sequence is $P_{[1]} = \alpha_{[1]} t_0$ and its completion time is

$$C_{[1]} = t_0 + P_{[1]} = (1 + \alpha_{[1]}) t_0.$$

Similarly, the completion time for the job in the ith position is

$$C_{[i]} = C_{[i-1]} + P_{[i]} = t_0 \prod_{j=1}^{i} (1 + \alpha_{[j]}),$$

if it is processed before the machine's non-available period. Depending on the relation between the completion time of the last job before the non-available period and b_1, the starting time for the non-available period, we divide these situations into the following two cases.

Case 1. The ith job is the last job to be processed before the machine's non-available period and its completion time is strictly less than the starting time for the non-available period. In this case, we have

$$C_{[i]} = t_0 \prod_{j=1}^{i} (1 + \alpha_{[j]}) < b_1, \tag{1}$$

thus, the processing time for the $(i + 1)$th position is

$$P_{[i+1]} = \alpha_{[i+1]} C_{[i]},$$

and its corresponding completion time is

$$\begin{aligned} C_{[i+1]} &= C_{[i]} + P_{[i+1]} + (b_2 - b_1) \\ &= t_0 \prod_{j=1}^{i+1} (1 + \alpha_{[j]}) + (b_2 - b_1). \end{aligned}$$

Similarly, the completion time for the last job is

$$C_{[n]} = t_0 \prod_{j=1}^{n} (1 + \alpha_{[j]}) + (b_2 - b_1) \prod_{j=i+2}^{n} (1 + \alpha_{[j]}). \tag{2}$$

Clearly, the first term in Eq. (2) is a constant. Thus the makespan (or $C_{[n]}$) is minimized if the second term is minimized. Furthermore, it is observed that the order of arranging the jobs after the non-available period does not alter the makespan, neither does the order of the jobs before the non-available period. The only thing that affects the makespan is which jobs are processed before the non-available period and which jobs are processed after the non-available period. Therefore, using the fact that $\prod_{j=1}^{n}(1+\alpha_{[j]})$ is a constant, it is obvious that $\prod_{j=i+2}^{n}(1 + \alpha_{[j]})$ will be minimized if we choose $\alpha_{[i+1]} = \alpha_{\max}$ and maximize $\prod_{j=1}^{i}(1 + \alpha_{[j]})$ with the constraint of Eq. (1). Thus, by taking the logarithms, the problem is transformed into the following 0–1 integer programming problem.

$$\text{Maximize } Z = \sum_{j \in N'} a_j x_j$$

$$\text{subject to: } \sum_{j \in N'} a_j x_j < \theta, \quad x_j = 0 \text{ or } 1$$

$$\text{for } j \in N' = \{J_1, J_2, \ldots, J_n\} \setminus J_{\max},$$

where $a_j = \ln(1 + \alpha_j)$, $\theta = \ln(b_1/t_0)$ and $J_{\max}$ is the job with the maximal growth rate. In the optimal solution obtained from the 0–1 integer programming problem, the x_js with a value of 1 represent the jobs that are processed before the non-available period, while the x_js with a 0 value are the jobs that are processed after the non-available period. The number of x_js with a value of 1 is the total number of jobs that are processed before the non-available period. Hence, the minimum value for the makespan in case 1, denoted by $C^*_{\max}(1)$, can be calculated accordingly.

Case 2. The ith job is the last job to be processed before the machine's non-available period and its completion time is exactly equal to the starting time for the non-available period. Since

$$C_{[i]} = t_0 \prod_{j=1}^{i} (1 + \alpha_{[j]}) = b_1, \tag{3}$$

the starting time for the $(i + 1)$th job is $C_{[i]} + b_2 - b_1$ and the actual processing time for the $(i + 1)$th job is

$$P_{[i+1]} = \alpha_{[i+1]}(C_{[i]} + b_2 - b_1).$$

Thus, the corresponding completion time is

$$C_{[i+1]} = t_0 \prod_{j=1}^{i+1} (1 + \alpha_{[j]}) + (b_2 - b_1)\alpha_{[i+1]}.$$

Similarly, the completion time for the last job is

$$\begin{aligned} C_{[n]} = {} & t_0 \prod_{j=1}^{n} (1 + \alpha_{[j]}) \\ & + (b_2 - b_1)\alpha_{[i+1]} \prod_{j=i+2}^{n} (1 + \alpha_{[j]}) \end{aligned}$$

$$= t_0 \prod_{j=1}^{n} (1 + \alpha_{[j]}) + (b_2 - b_1) \frac{\alpha_{[i+1]}}{1 + \alpha_{[i+1]}} \prod_{j=i+1}^{n} (1 + \alpha_{[j]}). \quad (4)$$

Again, the first term in Eq. (4) is a constant, thus minimizing the makespan is the same to minimize the last term. Since $\prod_{j=1}^{n}(1+\alpha_{[j]})$ is a constant and using the fact that $a_j/(1+a_j) < a_i/(1+a_i)$ if $a_j < a_i$, it is clear that the second term in Eq. (4) is minimized if we find the solution from Eq. (3) and choose $\alpha_{[i+1]} = \alpha'_{\min}$ to be the minimal growth rate among the jobs that are processed after the non-available period. Thus, in order to find the jobs that are to be processed before the non-available period, the problem is equivalent to solve the following linear equation.

$$\sum_{j \in N} a_j x_j = \theta, \quad x_j = 0 \text{ or } 1 \quad \text{for } j \in N = \{J_1, J_2, \ldots, J_n\},$$

where $a_j = \ln(1 + \alpha_j)$ and $\theta = \ln(b_1/t_0)$. Once again, job j is scheduled before the non-available period if $x_j = 1$ in the solution. Once the jobs that are scheduled before the non-available period have been determined from the solution of the linear equation, the job that has the minimal growth rate among the unscheduled jobs is the first one to be processed immediately after the non-available period. The order for the other unscheduled jobs is arbitrary. However, the linear equation may have multiple solutions. In that case, the minimum value of the makespan in case 2, denoted by $C^*_{\max}(2)$, is chosen as the one where the minimal growth rate among the jobs scheduled after the non-available period is minimized.

In general, minimizing the makespan is the smaller value of the makespan obtained in cases 1 and 2. However, it will be the minimal value for case 1 if one of the following conditions holds.

Theorem 1. *If $\alpha_i \geqslant 1$ for $1 \leqslant i \leqslant n$, then $C^*_{\max}(1) \leqslant C^*_{\max}(2)$.*

Proof. It is clear from Eqs. (2) and (4). □

Theorem 2. *If $\{\prod_{i=1}^{n}(1+\alpha_i)\}\{t_0 + (b_2 - b_1)\alpha_{\min}/(1+\alpha_{\min}) \cdot (t_0/b_1)\} \geqslant C^*_{\max}(1)$ where $\alpha_{\min}$ denotes the minimal growth rate of all jobs, then $C^*_{\max}(1) \leqslant C^*_{\max}(2)$.*

Proof. From Eq. (4), it is obvious that

$$C^*_{\max}(2) \geqslant \left\{ \prod_{i=1}^{n} (1 + \alpha_i) \right\} \cdot \left\{ t_0 + (b_2 - b_1) \frac{\alpha_{\min}}{1 + \alpha_{\min}} \cdot \frac{t_0}{b_1} \right\} \geqslant C^*_{\max}(1). \qquad \square$$

4. Conclusion

In this paper we examined the problem of scheduling deteriorating jobs on a single machine with an availability constraint. The goal was to find an optimal schedule that minimizes the makespan. We showed that this problem can be transformed into a 0–1 integer programming and a linear equation problem and further simplified to a 0–1 integer programming problem under certain conditions.

Future research might focus on other criteria such as the mean flow time, the total weighted completion time and tardiness. Extending this problem to a general m-machine flowshop problem is also an interesting issue.

Acknowledgements

The authors are grateful to an anonymous referee whose constructive comments have led to a substantial improvement in the presentation of the paper. This work is supported by the National Science Council of Taiwan, Republic of China, under grant number NSC 90-2218-E-035-008.

References

[1] B. Alidaee, N.K. Womer, Scheduling with time dependent processing times: Review and extensions, J. Oper. Res. Soc. 50 (1999) 711–720.

[2] S. Browne, U. Yechiali, Scheduling deteriorating jobs on a single processor, Oper. Res. 38 (1990) 495–498.

[3] R.W. Conway, W.L. Maxwell, L.W. Miller, Theory of Scheduling, Addison-Wesley, Reading, MA, 1967.

[4] G.H. Graves, C.Y. Lee, Scheduling maintenance and semi-resumable jobs on a single machine, Naval Res. Logist. 46 (1999) 845–863.
[5] J.N.D. Gupta, S.K. Gupta, Single facility scheduling with nonlinear processing times, Comput. Industr. Engrg. 14 (1988) 387–393.
[6] Y.C. Hsieh, D.L. Bricker, Scheduling linearly deteriorating jobs on multiple machines, Comput. Industr. Engrg. 32 (1997) 727–734.
[7] W. Kubiak, S. van de Velde, Scheduling deteriorating jobs to minimize makespan, Naval Res. Logist. 45 (1998) 511–523.
[8] A.S. Kunnathur, S.K. Gupta, Minimizing the makespan with late start penalties added to processing times in a single facility scheduling problem, European J. Oper. Res. 47 (1990) 56–64.
[9] C.Y. Lee, Machine scheduling with an availability constraint, J. Global Optimization (1996) 395–419.
[10] G. Mosheiov, Scheduling deteriorating jobs under simple linear deterioration, Comput. Oper. Res. 21 (1994) 653–659.
[11] G. Mosheiov, Scheduling jobs with step-deterioration: minimizing makespan on single and multi-machine, Comput. Industr. Engrg. 28 (1995) 869–879.
[12] M. Pinedo, Scheduling: Theory, Algorithms, and Systems, Prentice-Hall, Upper Saddle River, NJ, 2002.