Journal of Statistics and Mathematics
Journal of Statistics and Mathematics
ISSN: 0976-8807 & E-ISSN: 0976-8815, Volume 2, Issue 2, 2011, PP-55-58
Online available at : http://www.bioinfo.in/contents.php?id=85

A NOTE ON FLOW SHOP SCHEDULING PROBLEM WITH INCREASING AND DECREASING LINEAR DETERIORATION ON WEIGHTED DOMINANT MACHINES

JANARDHAN K. MANE[1*], KIRTIWANT P. GHADLE [2]
[1,2] Department of Mathematics, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, M.S., India-431004
*Corresponding Author: Email- drkp.ghadle@yahoo.com

Received: November 06, 2011; Accepted: November 30, 2011

Abstract- In this paper, we study the flow shop scheduling problem with increasing and decreasing linear deterioration on weighted dominant machines and also deal with some special case of general, no-wait permutation flow shop scheduling problem, respectively. Special cases mean that the machines form an increasing series of dominant machines, and decreasing series of dominant machines. The objectives are to minimize one of the two regular performance criteria, namely, makespan, total completion time and weighted completion time. This objective is considered under following dominant machine constraint: *idm* and *ddm* are considered. Numerical examples of the solution approaches are provided.
Key words - Scheduling Problem, flow shop, Learning effect, Dominant machines.

Introduction

Flow shops are frequently found in industry and are characterized by a set of jobs J= {1,2,..., n} and a set of machines, M= {1,2,....m}. The set of jobs is processed sequentially on m machines. In the traditional flow shop problem use assume deterministic processing times, denoted by P_{ij} where indices I'∈ M and $j \in J$ represent a machine and a job, respectively. Furthermore all jobs are ready for processing at time zero and no other arrive later; a job may not be preempted by another job; jobs are not allowed to pass others; no job may be processed by more than one machine; machines may process no more than one hob at a time; and there are no down times due to machine breakdown or maintenance. In a flow shop problem, we usually determine the sequence of jobs to satisfy certain performance criteria including the minimization of makespan c_{max}, the sum of the job flow times, the mean tardiness or lateness, the maximum tardiness and the number of tardy or late jobs.

In the classical scheduling theory, a job processing time does not independent on job position in a sequence, however, in many realistic scheduling settings, the production facility improves continuously with time. As a result the processing time of a given job is shorter if it is scheduled later in the production sequence. In literature, this phenomenon is known as learning effect. Biskup [1] was the first one who investigated the effect of learning process as a function dependent on a number of repetitions during a production of similar items, in other words, processing times depend on a hob position in the sequence, i.e. $P_{jr} = P_j r^{\alpha}$, where P_j is the normal processing times o f hob J_j, r is the position of hob J_j in the sequence and $\alpha \leq 0$ is the learning index of job J_j .He studied the single machine problem of minimizing the total flow time, the weighted sum of completion time deviations from a common due date and the sum of job completion times. Similar works can be found in Mosheior [2], Mosheior and Sidncy [3], Bachman and Janiak [4].etc.

Wang and Xia [5] consider no-wait of no-idle flow shop scheduling problems with processing times dependent on starting time. In these problems hob processing time is a simple linear function of the hobs starting time and some dominating relationships between machines can be satisfied. They showed that for the problems to minimum Makeskpan of minimize weighted sum of completion time polynomial algorithms still exist. When the objectives are to minimize maximum lateness, the solutions of a classical version may not hold. Ng et al. [6] also consider three scheduling problems with a decreasing linear modal of the hob processing times, where the objective function is to minimize the total completion time, and two of the problems are solved optimally. Bachman et al. [7] consider the single – machine scheduling problem with start time dependent job processing times. They prove that the problem of minimizing the total weighted completion time is NP- hard. They also consider some special cases. Zhao et al. [8] consider a special type of the actual processing time, which is $P_i(t) = a_i(a + t)$, where a and b are positive constant. They prove that the single-machine scheduling problems of minimizing makespan, sum of weighted completion times, maximum lateness and maximum cost are polynomials solvable, and the two- machine flow shop scheduling to minimize the makespan can be solved by Johnson's rule.

We introduced such an interesting scheduling model in which the processing time of a job is a polynomial function of its starting time. This model reflects some real-life situations in which expected processing time of a job increases / decreases linearly or piecewise linearly on its starting time. Examples can be found in financial management, steel production resource allocations and national defiance, where any delay in tackling a task may result in an increasing/ decreasing time, cost etc. to accomplish the task.

We consider the general, no-wait flow shop scheduling problem with increasing and decreasing linear deterioration a weighted dominant machines respectively. Deterioration of a hob means that its processing time is a function of its execution start time. The "no-wait" constraint means that each job, once started, has to be processed without interruption until all of its operations are completed. In practice, this requirement may arise out of certain hob characteristics or out of the unavailability of intermediate storage machines.

In this paper, we consider flow shop scheduling problems with a learning effect on no-wait dominant machines. That is, the hob processing time is a function of its position r in the sequence and no machine is allowed to have no-wait time between processing any two operations. Since processing time $p_{m[5]}$ is zero (as operation O_{m5} is not to be performed). The objective is to minimize maximum completion time. The previous works on flow shop scheduling in an environment of a series of dominating machines can be found in Nouweland et al. [9], Ho and Gupta [10] and Xiang et al. [11].

The remaining part of the paper is organized as follows. In the next section 2, we give a general introduction to flow shop problem with a learning effect and dominant machines. In section 3, we consider the minimizing the weighted sum of completion time. Final section includes conclusions and remarks about future research.

Notations:

P_{ij} = Processing tine of job J_j on machine M_i,
d_j = Due date of job J_j,
w_j = Weight of job J_j,
c_j = Completion time of job J_j in a given permutation,
a = Learning index,
idm = Increasing series of dominating machines,
ddm = Decreasing series of dominating machines,
C_{max} = max $\{cj \mid j = 1, 2, \ldots, n\}$, makespan of a given permutation.

To the author's knowledge no literature on a note on flow shop scheduling problem with increasing and decreasing linear deterioration on weighted dominant machines has been published.

Formulation

The sequence flow shop scheduling problem considered in this paper may be state as follows; we are given a set of n jobs $J_1, J_2, \ldots\ldots, J_n$ hat have to be processed on the machines $M_1, M_2, M_3, \ldots\ldots, M_m$ successively. The normal processing time of job J_i On machine (operation Q_{ij}) is P_{ij} ,the actual processing time of job J_i on the machine I is P_{ijr} if operation Q_{ij} is the r^{th} operation on machine i. we are asked to find the order in which these n job should be processed on the m machines such that a given objective is to find the schedule that minimizes the makespan. In this paper, we consider the hobs processing times characterized by position- dependent function:

$P_{ijr} = P_{ijr} r^{\alpha}$ i= 1, 2… m; r, j=1, 2… n. (1)

Where $\alpha \leq 0$ denotes a learning index.

for a given schedule σ, let $C_{ij} = C_{ij}(\sigma)$ represent the completion time of operation O_{ij}, $C_j = C_{mj}$ represents the completion time of hob J_i, σ = ([1],[2],……[n]) denote a schedule, where [j] denotes the job that occupies the jth position in σ. The problems considered in the paper are denoted according to the three field natation $\alpha/\beta/\gamma$ introduced by Graham et al. [12]

Definition 1. M_i is dominated by M_k, iff max$\{P_{ij}/j = 1,2,\ldots\ldots,n\} \leq \min\{P_{kj}/j = 1,2,\ldots\ldots,n\}$.In abbreviated natation, it is denoted as $M_i < M_k$ based on the above concept of dominant machines the five definitions considered in this paper are as fallows.

Definition 2. The machines form an increasing series of dominating machines (*idm*). That is

$$M_1 < M_2 <,\ldots\ldots\ldots, < M_m.$$

Definition 3. The machines form an decreasing series of dominating machines (*ddm*), that is

$$M_1 > M_2 >,\ldots\ldots\ldots, > M_m$$

Minimize the Total Weighted Completion Time

The following results of lemmas can be easily obtained, and the results can be used in latter:

Lemma 1: *(Mosheior [2]). For the problem* $1/P_{jr} = P_j r^a / C_{max}$ *and optimal schedule can be obtained by SPT (shortest processing time first) rule.*

Lemma 2: *For the problem* $F_m // P_{ijr} = P_j r^a, no-wait, idm / C_{max}$ *and a given schedule* σ = ([1], [2],........,[n]) *, the completion time* $C_{(i)}$ *of job* $J_{(i)}$ *is as follows*

$$C_{[j]} = \sum_{i=1}^{m} P_{i1} + \sum_{k=2}^{j} P_{m[k]} k^a$$

Now, we demonstrate the results of lemma 2 in the following example.

Example 1. $N = 4, M = 4, P_{11} = 12, P_{12} = 15, P_{13} = 17, P_{14} = 18, P_{21} = 19, P_{22} = 20, P_{23} = 23, P_{24} = 24, P_{31} = 26, P_{32} = 28, P_{33} = 29, P_{34} = 31, P_{41} = 31, P_{42} = 33, P_{43} = 34, P_{44} = 36, \& W_1 = 4, W_2 = 6, W_3 = 8, W_4 = 9$.

Learning curve that is $a = -0.2$. Obviously the condition of example 1 conforms to the case of *idm*, that is $M_1 < M_2$.

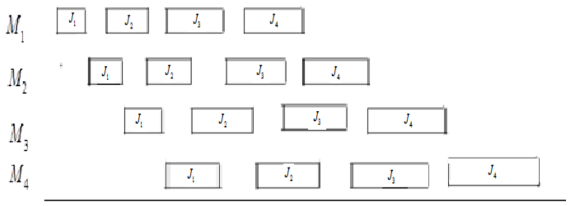

Fig. 1. A sample of $F_m // P_{ijr} = P_{ij}r^a, no-wait, idm/C_{max}$, n=4, m=4

Lemma - 3. *For the problem* $F_m \mid P_{ijr} = P_{ij}r^a$ *, no-wait, ddm/*C_{max} *and a given schedule*

$$\sigma = ([1],[2],\dots\dots\dots\dots\dots\dots,[n]),$$

the completion time $C_{[j]}$ *of job* $J_{[j]}$ *is as follows*

$$C_{[j]} = \sum_{k=1}^{n-1} P_{1[k]}k^a + \sum_{k=1}^{m} P_{k[n]}n^a + \sum_{k=j+1}^{n} P_{m[k]}k^a \quad \text{........ (3)}$$

Example2. $N = 4, M = 4$, $P_{11} = 28$, $P_{12} = 26$, $P_{13} = 24$, $P_{14} = 23$, $P_{21} = 22$, $P_{22} = 21$, $P_{23} = 19$, $P_{24} = 18$, $P_{31} = 18$, $P_{32} = 17$, $P_{33} = 15$, $P_{34} = 12$, $P_{41} = 10$, $P_{42} = 09$, $P_{43} = 07$, $P_{44} = 06$, & $W_1 = 8, W_2 = 7$, $W_3 = 7, W_4 = 6$.

Learning curve that is $a = -0.2$. Obviously the condition of example 2 conforms to the case of *ddm,* that is $M_1 > M_2$.

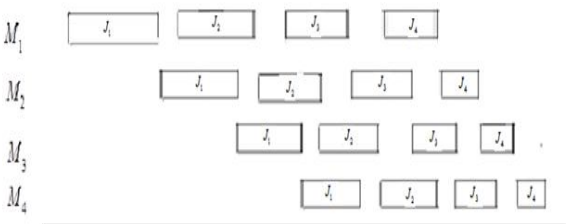

Fig. 2. A sample of $F_m \mid P_{ijr} = P_{ij}r^a$, no - wait, *ddm*/$C_{max}$ n=4, m=4

Theorem -4. *For the problem* $F_m \mid P_{ijr} = P_{ij}r^a$*, no–wait, idm*$\mid \sum W_jC_j$ *if the first processed job* J_1*asccrtained, then the schedule* $\sigma = \{J_t, \sigma_1\}$ *is an optimal one,* σ_1 *is a partial sequence obtained by sequencing the remaining (n-1) jobs is non-decreasing order* $\{P_{mj}\}$.

Proof: Consider the sequence

$$\sigma = (J_{[1]}, J_{[2]}, , \dots\dots\dots\dots\dots\dots, J_{[n]}).$$

By lemma 2, we have

$$\sum W_jC_j = \sum_{j=1}^{n} W_j \left[\sum_{i=1}^{m} P_{i[1]} + \sum_{k=2}^{n} P_{m[k]}K^a\right]$$

where the term $\sum_{j=1}^{n} W_j$ is a constant.

If the job processed first as curtained then combining $\sum_{k=2}^{n} P_{m[k]}K^a$ can be minimized by sequencing the remaining (n-1) jobs in non-decreasing order of their normal processing times on the last machine by Lemma 1. An optimal schedule for the problem $F_m \mid P_{ijr} = P_{ij}r^a$, no-wait, *idm*$\mid \sum W_jC_j$ is obtained.

Therefore, an optimal schedule of the problem $F_m \mid P_{ijr} = P_{ij}r^a$, no-wait, *idm*$\mid \sum W_jC_j$ can be constructed as follows

Select J_1, J_2, J_3 …J_n as the first processed job in turn, then the remaining (n-1) job are sequenced in non-decreasing order of $\{P_{mj}\}$on the last machine, respectively, thus n schedules are generated. The one with the minimum weighted sum of completion time among these n schedules is an optimal schedule.

Theorem 5. *For the problem* $F_m \mid P_{ijr} = P_{ij}r^a$ *, no-wait, ddm*$\mid \sum W_jC_j$*, if the last processed job* J_s *ascertained, then the schedule* $\sigma = \{\sigma_1, J_s\}$ *is an optima one, where* σ_1 *is a partial sequence obtained by sequencing the remaining (n-1) job in non-decreasing order of* $\{P_{ij}\}$.

Proof. Consider the sequence

$\sigma = (J_{[1]}, J_{[2]}, , \dots\dots\dots\dots\dots\dots, J_{[n]})$.

By lemma 3, we have

$$\sum W_jC_j = \sum_{k=1}^{n-1} P_{1[k]}k^a + \sum_{k=1}^{m} P_{k[n]}n^a + \sum_{k=j+1}^{n} P_{m[k]}k^a$$

where the term $\sum_{j=1}^{n} W_j$ is a constant.

If the last processed job ascertained, then combining $\sum_{k=1}^{n-1} P_{1[k]} K^a$ can be minimized by sequencing the remaining (n-1) jobs in non-increasing order of their normal processing times on the first machine by lemma1. An optimal schedule for the problem $F_m \mid P_{ijr} = P_{ij}r^a$, no-wait, *ddm*$\mid \sum W_jC_j$ is obtained.

Therefore, an optimal schedule of the problem $F_m \mid P_{ijr} = P_{ij}r^a$, no-wait, *ddm*$\mid \sum W_jC_j$ can be constructed as follows.

Select J_1, J_2, J_3, …, Jn as the first processed job in turn, then the remaining (n-1) job are sequenced in non-decreasing order of $\{P_{ij}\}$ on the last machine, respectively, thus n schedules are generated. The one with the minimum weighted sum of completion time among these n schedules is an optimal schedule.

Conclusion

This paper considers some permutation flow shop scheduling problem with increasing and decreasing linear deterioration on weighted dominant machines. It was show that some special cases of minimizing the discounted total weighted completion time can be solved in polynomial time. The objective is to minimize maximum completion time. For the objective, the following dominant machines constraint: *idm* and *ddm* are considered. Scheduling problems with such a learning effect in some other machine settings are also interesting and significant for future research.

Acknowledgement

The authors are thankful to the two anonymous referees for their helpful comments and suggestions that have improved the presentation of this paper.

References

[1] Biskup D. (1999) *European journal of operational Research* 115 173-178.

[2] Mosheior G. (2001) *European journal of Operational Research* 132, 687-693.

[3] Mosheior G., Sidney J. B. (2003) *European journal of Operational Research* 147,665-770.

[4] Bachman J., Janiak A. (2004) *Journal of the operational Research society* 55, 254-257.

[5] Wang J.B, Xia Z.Q. *Flow Shop Scheduling with Deteriorating Jobs under Dominating Machines. Omerga, in press*.

[6] Ng C.T., Cheng T.C.E., Bachman A., Janiak A. (2002) *Information processing letters,* 81:327-333.

[7] Bachman A., Chemg T.C.E.,Jainak A., Ng C.T. (2002) *Journal of the Operational research society,* 53:688-693.

[8] Zhao C.L., Zhang Q.L., Tang H.Y. (2003) *Acta Automotca Sinica* , 29:532-535.

[9] Nouweland A. V. D., Krabbenborg M., Potters J. (1992) *Eur. J. Oper. Res.* 62 38-46.

[10] Ho J. C., Gupta J. N. D. (1995) *Comput. Opl. Res.* 22, 237-246.

[11] Xiang S., Tang G., Cheng T. C. E. (2000) *Int. J. prod. Econ.* 66,53-57.

[12] Graham R. L., Lawler E. L., Lenstra J. K., Rinnooy Khan A. H. G. (1979) *Annals of Discrete Mathematics* 5, 287-326.

Table- 1-A sample of $F_m // P_{ijr} = P_{ij}r^a, no-wait, idm/C_{max}$, *n=4, m=4*

Machines i	weighted	Processing time of the increasing order				Actual completion time	Weighted completion time
		Job					
		1	2	3	4		
1	4	12	15	17	18	88	352
2	6	19	20	23	24	116.72	700.32
3	8	26	28	29	31	144.01	1152.08
4	9	31	33	34	36	171.29	1541.61

Table- 2- A sample of $F_m \left| P_{ijr} = P_{ij}r^a \right.$ *, no-wait, ddm /*$C_{max,}$*, n=4, m=4*

Machines i	weighted	Processing time of the decreasing order				Actual completion time	Weighted completion time
		Job					
		1	2	3	4		
1	8	28	26	24	23	132.61	1060.88
2	7	22	21	19	18	124.77	873.39
3	7	18	17	15	12	119.15	834.05
4	6	10	09	07	06	114.61	687.66