Scheduling in Proportionate Flowshops

Ram Mohan V. Rachamadugu*
A. Vepsalainen** and Thomas E. Morton***

CMU-RI-TR-83-10

*Graduate School of Business
University of Michigan

**Wharton School of Business
University of Pennsylvania

***Graduate School of Industrial Administration
Carnegie-Mellon University

April 1982

This research was supported, in part, by the Air Force Office of Scientific Research under contract F49620-82-K0017.

Abstract

It is well known that except in the case of makespan problems, there are hardly any analytical results for flowshop problems. This paper considers of a class of flowshop problems where job processing time at a machine is proportionate to the processing time on the first machine. We show that for the pre-emptive version of the problem, in order to minimize any regular measure of performance, it is sufficient to consider permutation schedules. Also, results for various other measures are derived* A characterization of the optimal solution for the weighed tardiness problem is derived which is analogous to its counterpart in the single machine case. It is indicated as how this characterization nay be used to develop heuristics for flowshop problems.

SCHEDULING IN PROPORTIONATE FLOWSHOPS

1.0 Introduction

Flowshop problems have been the center of attention for researchers in Scheduling Theory for a long period of time. Though flowshop problems are a special case of general jobshop problems, even these problems have proven themselves to be too complex to provide many analytic solutions. As has been established by Lenstra [12], most problems in this area fall in the NP-Complete class. There are no known polynomially bounded procedures for this class of problems and it is unlikely that there are any such procedures. Most prior research in the field of flowshop problems was confined to makespan problems. The most widely quoted result is due to Johnson [10] to minimize makespan in two machine flowshop problem and its extension to a special case of three machine flowshop problem. Also, Gilmore and Gomory [6] devised an algorithm with a computational burden of $O(n^2)$ for the two machine flowshop problem where job waiting is not permitted. There are hardly any other known polynomially bounded procedures for the problems in flowshops. Another most widely quoted result is due to Conway, Maxwell and Miller [4] proving the optimality of the same permutation sequence on the first two machines in a flowshop for any regular measure of performance and the additional result that the sequence on the last two machines is the same for makespan problems. The fact that these results were discovered more than two decades ago and no further significant progress has been made in the case of flowshop problems in deriving analytical attests to the complexity of these problems. Most of the recent research in flowshops has been largely directed towards finding optimal solution using enumerative methods such as branch and bound or developing

"good" heuristics for makespan problems [1,2,3,5,7,13,14,15]• There is hardly any significant work done for other important measures of performance.

This paper addresses scheduling problems in the context of a particular kind of flowshop where task processing time of any job at a machine is proportionate to the processing time on the first machine. Results derived in this paper relate to the problems where the jobs can be pre-empted. We show that in such a case, permutation schedules constitute the set of dominant schedules for any regular measure of performance and we further derive results for performance measures based on completion times and/or the due dates of the jobs. These results hold good even in cases where job-passing is prohibited. In case of shops where intermediate queues are prohibited (once a job is began on the first machine, it has to be processed without interruption at any subsequent machine), these results hold good except that the start times on tie first machine have to be appropriately delayed.

2.0 Fenmitation Schedules for the proportionate flowshops

In this section, we consider pre-emptive version of the general problem for the proportionate flowshop problem. We wish to schedule a set of jobs, $\{J_i tJjtJy'J\}_n$ so as to minimize a regular measure of performance. Firstly, it is not unusual to find jobs being pre-empted in practice in order to ejpedite them through the production system* Secondly, pre-emptive case is an iapartamt relaxation of the original problen from the cou^utational poiat of view* The following proposition holds good for the pre-emptive case*

PRPP0SITI08 1: For minimizing any regular measure of performance, it is sufficient to consider permitation schedules.

Proof: Consider an optimal schedule in which the ordering of jobs is not the same on the last two machines m-1 and m. Consider any two jobs J_i and J_j such that $J_i < J_j$ on machine m and $J_i < J_i$ on machine m-1 as in Figure 1.

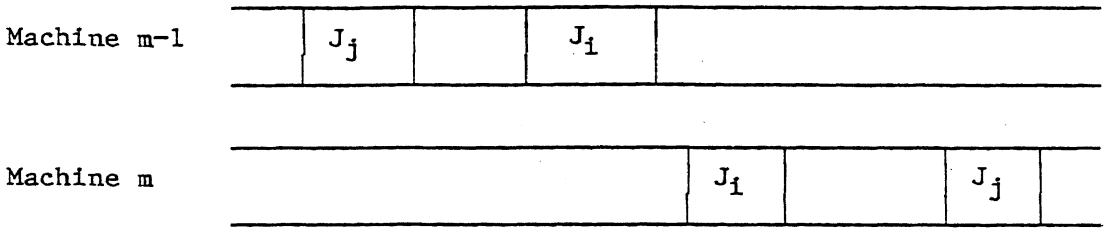

FIGURE 1

Since all jobs have the same processing time on any particular machine, pairwise interchange of any two jobs on a particular machine does not affect the completion times of any other jobs on that particular machine. So, pairwise interchange of jobs J_i and J_j on machine m-1 does not affect completion time of any other job on machine m-1. If such pairwise interchange on machine m-1 is forbidden by the schedule on machine m-2, we can switch jobs J_i and J_j on machine m-2 as well and so on back to the first machine. Thus, we can always form an optimal schedule in which machines m-1 and m have the same sequence and completion times of jobs on machine m are no greater than the original given optimal schedule. Now, we extend the same argument inductively between machines m-1 and m-2, m-2 and m-3,...2 and 1. Since the completion times of the jobs are no greater than the completion times in the original schedule, permutation schedules constitute the set of dominant schedules for any regular measure of performance.

Now we derive some results relating to the completion times of the jobs. Let P_k represent the processing time for any job (piece) on machine k.

Consider any permutation schedule. Let $C^k_{[i]}$ represent the completion time for the piece in the i th position on machine k. The following result holds:

PROPOSITION II: For any piece.

$$C^j_{[i]} = \Sigma_{I=1}^{I=j} p_I + (i-1) \max_{q=i,2..j} \{\rho_q\}$$

PROOF: In a permutation schedule, same sequence is used on all machines.

$$C^1_{[1]} = p_1$$

$$C^2_{[1]} = p_1 + p_2$$

$$\vdots$$

$$C^j_{[1]} = \Sigma_{I=1}^{I=j} p_I$$

The rest of the proof is by induction. Suppose that in a permutation schedule $C^k_{[1]} - C^k_{[i-1]} = D^k$ for some particular machine k (this is obviously true for k=1 and i=2,3,4....n). We show that $C^{k+1}_{[i]} - C^{k+1}_{[i-1]} = D^{k+1}$ where D^{k+1} is a constant and is given by $\max(D^k, p_{k+1})$.

We have two cases to consider- 1) $p_{k+1} \geq D^k$ and 2) $p_{k+1} < D^k$

Case 1: $p_{k+1} \geq D^k$

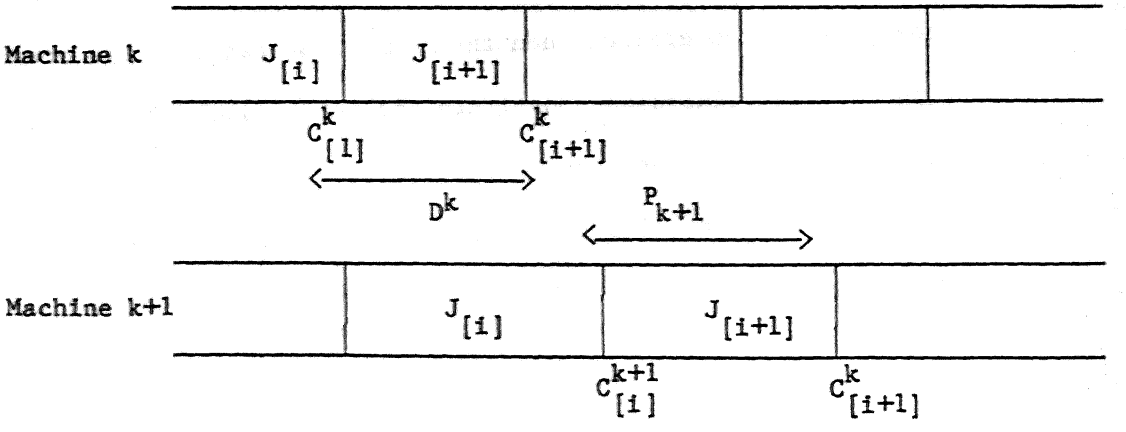

Figure 2

In this case, there is no idle time on machine k-1. Therefore, $C^{k+1}_{[i-1]} - C^{k+1}_{[i-1]} = p_{k+1} = \max (D^k, p_{k+1})$

Case 2 : $p_{k+1} < D^k$

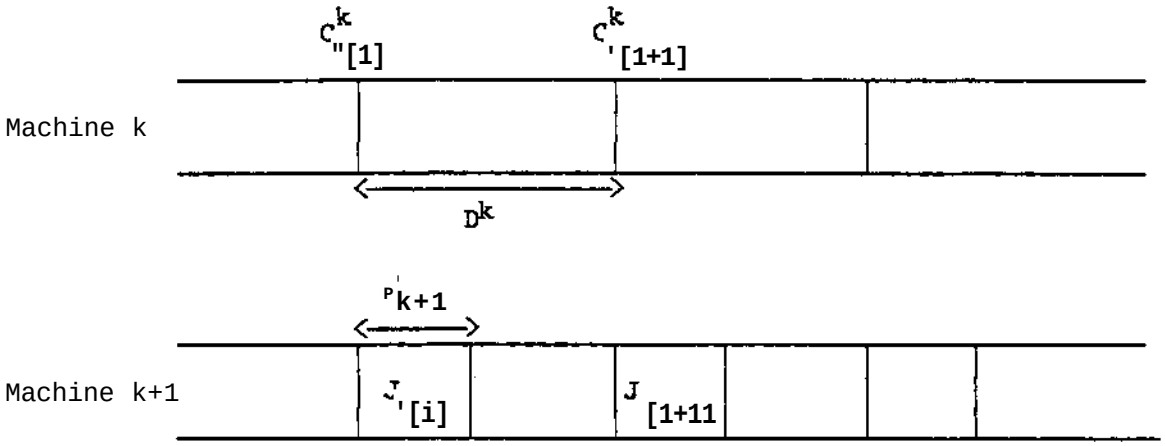

Figure 3

In **this case,**

$$c_{[i]} = C^{k+1}_{[i-1]} = D^k$$

$$C^{k+1}_{[i]} - C^{Jc+1}_{'[1-1]} = \max \{D^k, P_{k+1}\}$$

Thus,

$$C^j_{[i]} = C^j_{[1]} + \Sigma_{i=2}^{i=j} \{ C^j_{[i]} - C^j_{[i-1]} \}$$

$$= C^j_{[1]} + (i-1) \max \{ D^{j-} \setminus P_j \}$$

$$= C^j_{[i]} + (i-1) \max_{q=1\ldots j} \{ p_q \}$$

We had earlier indicated that makespan problems are the most widely researched area in the case of flowshop problems. Further, it is well known that the optimal schedule *need* not necessarily be a permutation schedule except that the sequence is the same on the first two machines and also on the last two machines. However, when all jobs have equal processing tines on the first machine, the following proposition holds good in the case of proportionate flowshop*

PROPOSITION III: Any permutation schedule provides the minimu makespan for the proportionate flowshop problem in the case of jobs equal processing times on the first machine.

PROOF: Let p_{max} be the maximum processing time of a job on son machine. Work content at this particular machine is np_{max}. Also, e has to undergo processing prior to and subsequent to this machine. the minimum processing time for these operations is $\Sigma_{k=1}^{k=m} p_k - p_{max}$. the minimum makespan is given by

$$\Sigma_{k=1}^{k=m} p_k - p_{max} + np_{max} = \Sigma_{k=1}^{k=m} p_k + (n-1)p_{max}$$

From Proposition II, it is clear that the minimum makespan is ac by any permutation schedule and hence the result

Now we discuss some measures relating to the completion times of the case of the proportionate flowshop for jobs with equal processing

COROLLARY 1: Any permutation schedule of the pieces minimizes F

PROOF: $\overline{F}$ if a regular measure of performance and permutation sc constitute the set of dominant schedules. From Proposition II, it is that all permutation schedules have the same $\overline{F}$.

$$\overline{F} = (1/n)*(\Sigma_{[i]=1}^{[i]=n} C_{[i]}^{m}$$

$$= \Sigma_{k=i}^{k=n} p_k + 1/2(n-1)p_{max}$$

COROLLARY 2: $\overline{F}_w$ is minimized by scheduling the jobs according to weighted shortest processing time rule.

PROOF: $\overline{F}_w$ is a regular measure of performance and we need to consider only permutation schedules. Completion times of jobs in a permutation schedule is given by

$$C^m_{[1]} = \Sigma_{k=1}^{k=m} p_k + (i-1)\, p_{max} \qquad \text{(application of Proposition II)}$$

$$F_w = (1/n)^* \{ \Sigma_{i=1}^{i=h} W_{[i]} C^m_{[i]} \}$$

It follows directly from basic algebra that the product of two series is minimized by arranging one in the ascending order and the other in non-ascending order.

Just as in the single machine case, we can show in this case also that arranging the jobs in non increasing order of the weights minimizes the weighted lateness as shown below:

COROLLARY 3: The Earliest Due Date rule minimizes maximum lateness and maximum tardiness.

PROOF: Consider any two adjacent jobs J_i and J_j in a given schedule such that $J_i < J_j$ and $d_i > d_j$. Let t be the completion time of J_i.

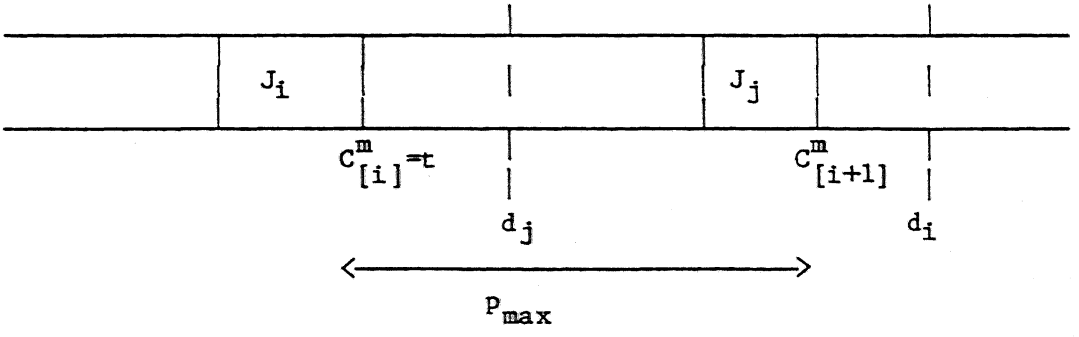

Figure 4

Maximum lateness among jobs J_i and J_j is given by

$$\max \{ t - d_i,\ t + p_{max} - d_j \} = t - p_{max} - d_j \qquad \ldots(1)$$

Suppose we interchange J_i and J_j. Maximum lateness among J_i and J_j is given by

$$\max \{ t - d_j, t + p_{max} - d_i \} \qquad \ldots(2)$$

It is clear that (1) > (2). Thus, by interchanging J_i and J_j, the schedule is no worse off and in fact, it would improve if the maximum lateness in the original sequence occurred for J_j. Since T_{max} equals $\max(0, L_{max})$, the result holds good for maximum tardiness as well.

Another important measure of performance is weighted average tardiness. Since this is a regular measure of performance, it is sufficient to consider only permutation schedules. Following results relate to this measure of performance for jobs with equal processing times on the first machine in the case of proportionate flowshops.

PROPOSITION IV: The optimal pre-emptive solution to the $\Sigma w_i T_i$ problem is found by solving the linear assignment problem.

PROOF: It is clear from the Proposition II that $C^m_{[i]}$ is independent of the job occupying i th position in the sequence. We can form the cost matrix tableau for the linear assignment problem (ψ_{ij} indicates the penalty incurred if J_j is in the i th position in the sequence) as follows:

$$\psi_{ij} = w_j \max \{ 0, \Sigma_{k=1}^{k=m} p_k + (i-1)*p_{max} - d_j \}$$

Solving the linear assignment problem using the above cost tableau yields optimum solution. It may be noted that the solution procedure has a computational burden of the order of $0(n^3)$.

In fact, the result in the Proposition IV can easily be generalized to any penalty function of the completion times of the jobs so long as they are

nondecreasing functions of the completion times of the jobs and the performance measure is additive over the completion of the jobs.

Though Proposition IV provides us with a polynomially bounded procedure for solving the pre-emptive version of Ew_iT_i, problem, the following characterization of optimal solution for the same problem is interesting from the point of view of developing heuristics for the flowshop problems.

PROPOSITION V: Consider an optimal sequence for Ew_iT_i problem for jobs with equal processing times on the first machine for the proportionate flow-shop. Consider any two jobs, J_i and J_j, $i < j$ (without loss of generality, assume that job index is same as the locational index in the sequence under consideration)• Then, the following property must be satisfied in an optimal sequence-

$$w_i\left\{1 - \frac{(d_i - C[i])^{+}}{(j-i)\,P_{max}}\right\}^{+} \geq w_j\left\{1 - \frac{(d_j - C[i])^{+}}{(j-i)\,P_{max}}\right\}^{+}$$

PROOF: The proof is similar to the proof provided in the appendix of an earlier paper on the myopic heuristics for the single machine tardiness problem [16] and is omitted here for the sake of brevity.

This property can be considered to be valid for a relaxation of the general problem in proportionate flowshops where jobs are permitted to be preempted at unit intervals on the first machine and all such preempted pieces have the same' due date as the original job.

However, if all jobs have equal weights and equal processing times, then the earliest due date sequence provides an optimum sequence for the average tardiness problea as shown in the next proposition—

PROPOSITION VI: If all jobs(pieces) have equal weights, the earliest due date sequence minimizes the average tardiness.

PROOF: From Proposition I, it is clear that we have to consider only permutation schedules. Consider an optimal solution in which two successive jobs do not follow the earliest due date rule, i.e., $J_i < J_j$ and $d_i > d_j$.

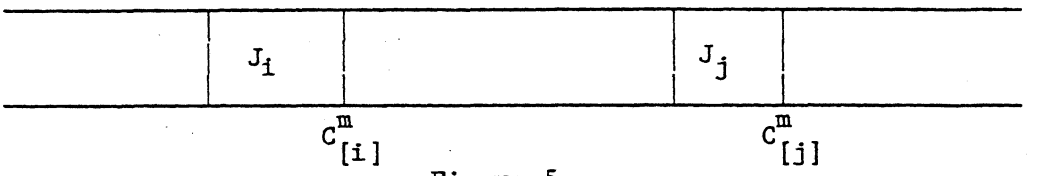

Figure 5

Case 1: Suppose that both J_i and J_j are early or on time. Since J_j is early or on time and $d_i > d_j$, pairwise interchange does not degrade the solution.

Case 2: Both J_i and J_j are tardy. Pairwise interchange does not degrade the solution since the weights are equal.

Case 3: J_i is tardy and J_j is early or on time. This is impossible since $d_i > d_j$ and $C_{[i]}{}^m < C_{[j]}{}^m$

Case 4: J_i is early or on time and J_j is tardy.

Subcase 4.1

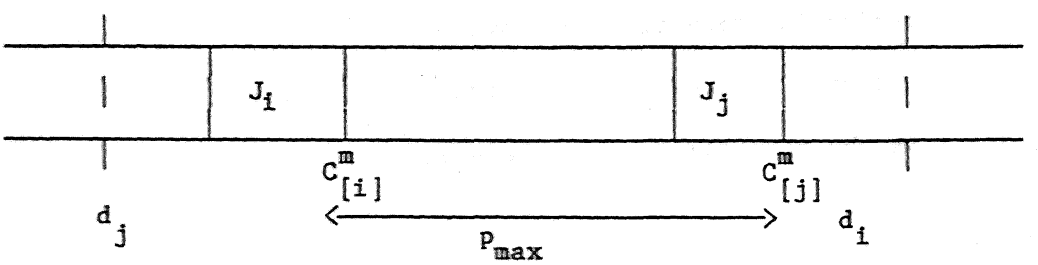

Figure 6

Clearly, pairwise interchange improves the solution.

Subcase 4.2

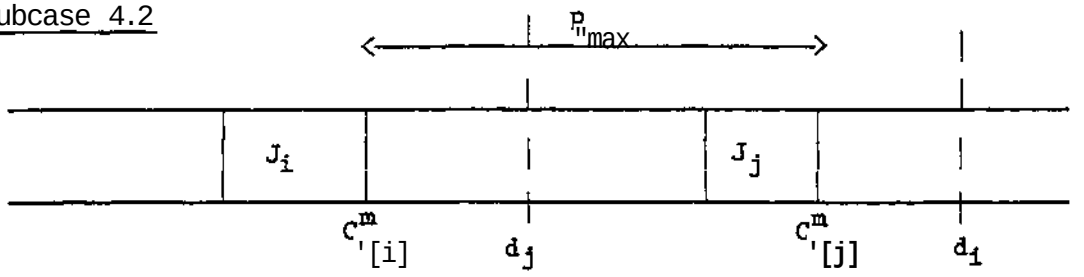

Figure 7

Clearly, pairwise interchange improves the solution.

Subcase 4.3

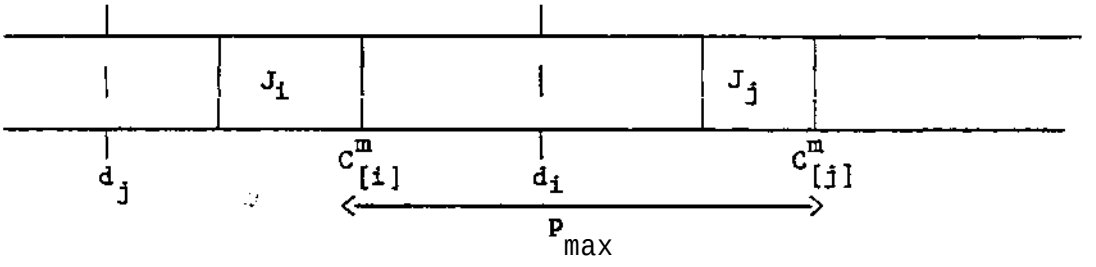

Figure 8.

Cost of J_i and J_j in given schedule $= C^m_{[i]} + p_{max} + p_j$ .... (3)

Contrilxition after interchange $= (C^m_{[i]} + p_{max} - d_j) + (C^m_{[i]} - d_j)$ .... (4)

Subtracting (4) from (3),

$= d_i - C^m_{[i]}$

> 0

Therefore, pairwise Interchange results in an iaprovement.

Subcase 4.4

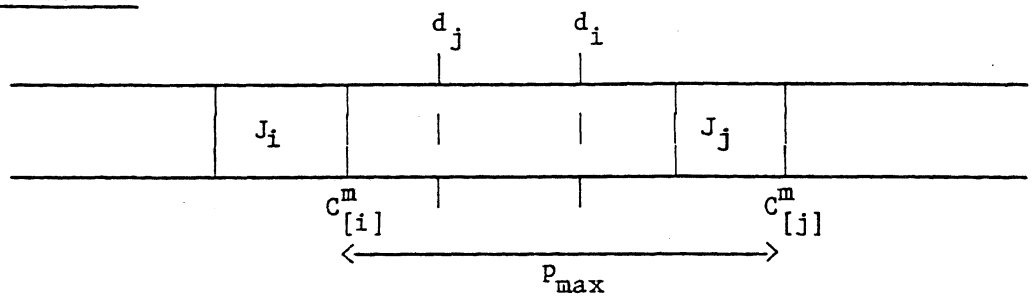

Figure 9

Cost of J_i and J_j in given schedule $= C^m_{[i]} + p_{max} - d_j$ .... (5)

Contribution after interchange $= C^m_{[i]} + p_{max} - d_i$ .... (6)

Since $d_i > d_i$ (5) (6). Therefore, pairwise interchange improves the solution.

Thus, in all cases, pairwise interchange does not degrade the solution and, in fact, may improve it. Since our arguments employ only information about the individual jobs and not the location, ensurance of local optimum at all locations in the sequence ensures global optimum and hence the earliest due date rule is optimal.

3.0 Schedules with no job-passing

There is a special class of flow-shop problems where no job passing is permitted. That is, once a job is begun on the first machine, it maintains same priority relative to other jobs for subsequent processing on any other machine. No job-passing is a matter of practical and design expediency. As stated by King [11], "this is typically the situation in many manufacturing plants where jobs are moved from station to station by conveyor." Even in Flexible Manufacturing Systems, due to problems involved in computation of

optimal resource utilization, not more than two or three jobs are permitted to pass the others in the sequence [8,9]. Also, since technologically designing input buffers to machines to accommodate any scheme other than First Come, First Served is rather complex, in many situations no job-passing restriction is used.

In case of proportionate flowshops, the following remark holds good.

REMARK 1: Permutation Schedules $\Longleftrightarrow$ Schedules with no job-passing.

Hence all results derived in §2.0 equally hold good for jobs with equal processing times in proportionate flowshops.

4.0 Schedules with no job-waiting

Another special class of flowshop problems are those where job waiting is forbidden. Once a job is begun on the first machine, it must be processed with no waiting at any other machine. Steelmaking is an example of such a situation [11,17]. It is clear that schedules with no job-waiting are a subset of schedules with no job-passing. So, here again, it suffices to consider only permutation schedules for optimizing any regular measure of performance. But, due to the no-wait condition, it would be necessary have inserted idle time on the first machine. An exact algorithm for minimizing makespan for the case of two machines with no job-wait is given by Gilmore and Gomory [6]. Wismer [17] has shown that the makespan problem for general flowshop problem with no job waiting can be translated into an equivalent Asymmetric Traveling Salesman Problem. Lenstra [12] has shown that the Hamiltonian Path problem is reducible to makespan problem in flowshops with no job-wait, thus establishing the latter problem to be No-Complete. King and Spachis [11] developed

heuristics for this problem and tested them against random sequences and other heuristics.

However, in the case of jobs with equal processing times to be processed in proportionate flowshop, we can easily extend the results obtained in §2.0 even for situations where job-waiting is not permitted.

PROPOSITION VII: Any permutation sequence for proportionate flow-shop (all jobs with equal processing times on the first machine) can be scheduled so that completion times on the last machine are not changed and the jobs do not form queues at any machine.

PROOF: Consider two adjacent jobs, J_i and J_{i+1}. Suppose J_i starts on machine 1 at time t. Then,

$$C_i^m = t + \Sigma_{q=1}^{q=m} p_q$$

J_{i+1} can start on machine 1 only at such a time that once its processing has begun, it does not have to wait at any other machine. In order to determine when J_{i+1} complete on machine m, we simply left shift J_{i+1} such that its processing on machine m can begin immediately after J_i is complete on machine m (Figure 10) and then right shift it to the minimum possible extent to make it feasible (Figure 11).

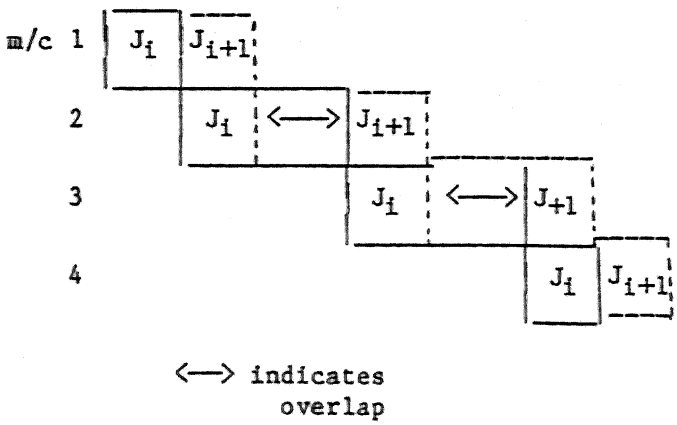

Figure 10

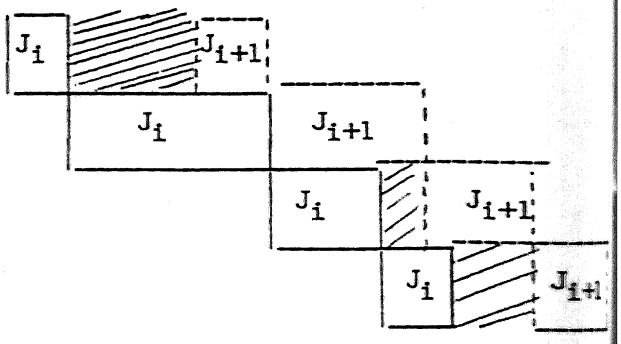

Figure 11

$$C_{[i]} = t \cdot V_1 P_q$$

Overlap of J_i and J_{i+1} on machine j $= \max \{ 0, C_{[i]j} - [C_{[i]x} + p_{[i]m} - Z_{M,J} p_{[i]H}] \}$

$= \max \{ 0, p_{[i]j} - p_{[i]m} \}$

Therefore, time difference between completion times of two successive jobs on machine m is given by

$$p_{[r]max} - p_{[i]m} + p_{[i]m} = p_{[r]max}$$

Thus, $C_{[i]}$ is given by

$$C_{[1]j} + (i-1)*p_{max}$$

We note that this value is same as the one derived in Proposition II with no constraints on job-waiting. Thus, all *the,* results derived in §2.0 hold good even in the case when job-waiting is prohibited. However, the start times on the first machine will be delayed so that there are no queues at intermediate machines- The start time for the job in the position i is given by

$$= C_{[I]} + (i-1)*p_{max} - \sum_{q=1}^{q=m} p_{[q]v}$$

$$= (i-1)*p_{max}$$

5.0 Conclusion

There are hardly any known analytic results for flowshop problems except in the case of makespan problem- We have derived results for the situation where job processing times at any machine are proportioaate to the time on the first machine* Though we considered the case where jobs are permitted to be preempted, these results may be used for developing lower bounds for non-preemptive cases. Also, the property developed for characterizing an optimal solution for the weighted tardiness problem can be used for developing heuristics for the flowshop problems- Our preliminary investigations in this direction appear to be promising and these aspects are currently being investigated-

BIBLIOGRAPHY

1. Bestwick, P. F., and Hastings, N. A. J. "A new bound for machine scheduling." Operational Research Quarterly 27 (1976), 479.

2. Bonney, M. C, and Grundy, S. W. "Solutions to the constrained flowshop sequencing problem." Operational Research Quarterly 27 (1976), 869.

3. Brooks, G. H., and White, C R. "An algorith for find optimal as near optimal solutions to the production scheduling problem." Journal of Industrial Engineering 16 (1965), 34.

4. Conway, R. W., Maxwell, W. L. and Miller, L. W. Theory of Scheduling. Addison-Wesley Inc., Reading, Massachusetts, 1967.

5. Dannenbrlng, D. G. "An evaluation of flowshop sequencing heuristics." Management Science 23 (1977), 1174.

6. Gilmore, P. C., and Gomory, R. E«, "Sequencing a one-state variable machine: a solvable case of the travelling salesman problem." Operations Research 12 (1964), 655-679.

7. Gupta, J. N* D. "The generalized n-job, it-machine scheduling problem." Operations Research 8 (1971), 173.

8. Hitz, K. L. Scheduling of Flexible Flowshops I. Research Report Lids-R-879. Massachusetts Institute of Technology, March, 1979. Report from the Laboratory for Information and Decision Systems.

9. Hitz, K. L. Scheduling of Flexible Flowshops II. Research Report Lids-T-1049, Massachusetts Institute of Technology, October, 1980. Report from the Laboratory for Information and Decision Systems.

10. Johnson, S. M. ""Optimal Two and Three-Stage Production Schedules with Set-up Times Included." Haval Research Logistics Quarterly 1, 1 (March 1954).

11. King, J. R., and Spachis, A* S. "Heuristics for flow-shop scheduling." International Journal of Production Research 18, 3 (May 1980), 345-357.

12. Lenstra, J. K» Sequencing by Enumeratlve Methods, Mathematlsch Centrum, Amsterdam, 1977.

13. Losmicki, *Z** A. "A branch-and-bound algorithm for the true exact solution of the three machine scheduling problem." Operational Research Quarterly 16 (1965), 89.

14. McHahon, G. B., and Burton, P. 6* "Flow-shop scheduling with the branch and bound Method.** Operations Research 15 (1967), 473.

15. Pal jobs through a multi-stage process in the minimum total time a H« thod of obtaining a near optimum." /-iQfi5i 101. Operational *a.:-:rch Quarterly 16 (1965), J-ui

16• Rachamadugu, R« ..* and Morton, T E. Myopic Heuristics for the Single Machine Weighted Tardiness Pr *llJ* Working Paper #2881-82, Graduate School of Industrial Administ llon University, Pittsburgh.

17. Wismer, D. A. "Solution of the flow-shop scheduling problem with no intermediate queues." Operations Rese