REVUE FRANÇAISE D'AUTOMATIQUE, D'INFORMATIQUE ET DE RECHERCHE OPÉRATIONNELLE. RECHERCHE OPÉRATIONNELLE

C. L. MONMA
A. H. G. RINNOOY KAN

A concise survey of efficiently solvable special cases of the permutation flow-shop problem

Revue française d'automatique, d'informatique et de recherche opérationnelle. Recherche opérationnelle, tome 17, n° 2 (1983), p. 105-119.

<http://www.numdam.org/item?id=RO_1983__17_2_105_0>

R.A.I.R.O. Recherche opérationnelle/Operations Research
(vol. 17, n° 2, mai 1983, p. 105 à 119)

A CONCISE SURVEY OF EFFICIENTLY SOLVABLE SPECIAL CASES OF THE PERMUTATION FLOW-SHOP PROBLEM (*)

by C. L. MONMA (1) and A. H. G. RINNOOY KAN (2)

Abstract. — *One of the earliest results in scheduling theory is an algorithm by S. M. Johnson for scheduling jobs in a two-machine flow-shop to minimize the time at which all jobs are completed. Subsequently, many researchers have efficiently solved special cases of this problem for more than two machines. A concise survey of such results is presented with simple proofs based on the ideas of critical paths, nonbottleneck machines, and machine dominance. This covers most previously known special cases and leads to a few new ones as well.*

Keywords: Flow-shop.

Résumé. — *Bien que la minimisation de la durée totale d'un problème « flow-shop » à m machines pose en général de nombreuses difficultés, un certain nombre de cas particuliers qui ont pu être résolus de manière efficace ont été étudiés dans la littérature. Nous en faisons ici la synthèse pour montrer comment la plupart d'entre eux peuvent être obtenus par application systématique d'un petit nombre de principes simples.*

Mot clé : ordonnancement.

1. INTRODUCTION

The *permutation flow-shop problem* can be formulated as follows. Each of n *jobs* $J_1, \ldots, J_n$ has to be processed on m *machines* $M_1, \ldots, M_m$ in that order. The processing of job J_i on machine M_j requires an uninterrupted period of *processing time* $p_{i,j}$. Each machine can process at most one job at a time. The objective is to find a *permutation schedule* (i. e., a single ordering in which to process the jobs on all of the machines) such that the time required to complete all jobs is minimized. A permutation schedule will be represented by a *permutation* $\pi = (\pi(1), \ldots, \pi(n))$, where $\pi(i)$ is the index of the i-th job in the processing order.

For $m = 2$, an optimal schedule can be found in $O(n \log n)$ time by an algorithm due to S. M. Johnson [12]. However, for $m \geqq 3$ it is unlikely that an

(*) Received January 1982.
(1) Bell Laboratories, Holmdel, NJ 07733, U.S.A.
(2) Erasmus University, Rotterdam, The Netherlands.

R.A.I.R.O. Recherche opérationnelle/Operations Research, 0399-0559/1983/105/$ 5.00

efficient (i. e., polynomial-time) algorithm exists, since this problem is known to be *NP*-complete [10, 18]. Because of this, much research has been done to find efficiently solvable special cases of the permutation flow-shop problem. In this paper we provide a concise and self-contained survey of these results including proofs of correctness. In Section 2, we review the connection between the length of a permutation schedule and the weight of a critical path in an associated directed graph. Subsequently, in Sections 3 and 4, we show how this notion can be combined with the concept of machine dominance to lead to very simple proofs for nearly all special cases which have been derived so far. Although similar attempts have been made before [3, 41, 42], we have made a special effort to demonstrate that the vast majority of the results in the extensive literature on this subject can be generated by systematic application of a few very simple ideas.

2. CRITICAL PATHS AND THE TWO-MACHINE PROBLEM

It is convenient to represent a permutation schedule $\pi=(\pi(1), \ldots, \pi(n))$ by a directed graphs as follows. We define a vertex $(\pi(i), j)$ with an associated weight $p_{\pi(i),j}$ for each element $\pi(i)$ of the permutation and each machine M_j. Also, we define arcs directed from each vertex $(\pi(i), j)$ towards $(\pi(i+1), j)$ and towards $(\pi(i), j+1)$. This graphs is depicted in figure 1 for the case where $m=4$ and $n=5$.

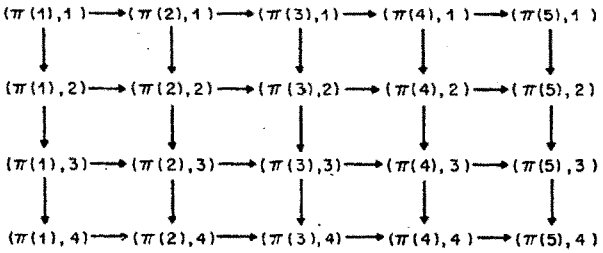

Figure 1. — Graph for a Permutation Schedule π.

Given this graph, the *completion time* $C_{\pi(i),j}$ of job $J_{\pi(i)}$ on machine M_j in the permutation schedule π is equal to the maximum-weight directed path from $(\pi(1), 1)$ to $(\pi(i), j)$ in the graph. $C_{\pi(i),j}$ is defined recursively by:

$$C_{\pi(i),j}=\max \{ C_{\pi(i),j-1}, C_{\pi(i-1),j} \}+p_{\pi(i),j},$$

where $C_{\pi(0),j}$ and $C_{\pi(i),0}$ are taken to be zero.

Therefore, the time at which all jobs in π are completed is given by $C_{\max}(\pi) = C_{\pi(n),m}$. Any path from $(\pi(1), 1)$ to $(\pi(n), m)$ which attains this maximum weight is called a *critical path* and contains $m+n-1$ vertices.

The notion of critical paths was introduced by Johnson [12] and was used extensively by Szwarc [41, 42] to obtain results for certain special cases of the permutation flow-shop problem. In particular, it can be used to find an efficient solution method for the case where $m=2$.

The efficient solution of the two-machine flow-shop problem is one of the oldest results in machine scheduling theory. An optimal permutation can be found in $O(n \log n)$ time by applying *Johnson's Rule* [12]: a permutation is optimal if job J_h precedes J_i whenever *Johnson's Condition:*

$$\min \{ p_{h,1}, p_{i,2} \} < \min \{ p_{h,2}, p_{i,1} \},$$

is satisfied. A simple proof is provided below. An optimal permutation always exists since Johnson's Condition can be seen to be transitive. We note that an optimal permutation ordered by Johnson's Rule has the property that an optimal permutation for any subset of the jobs is given by the order of these jobs in the original permutation. Also, the worst possible permutation is obtained by reversing the order obtained by Johnson's Rule.

The critical-path approach leads to a simple proof of Johnson's result [19]. Consider the graph representing a permutation with job J_i immediately preceding job J_h as shown in figure 2(*a*). Interchanging these jobs, as shown in figure 2(*b*), does not increase the completion time of the schedule if the weight of no critical path is increased, i. e., if:

$$\begin{aligned} &\max \{ p_{h,1}+p_{i,1}+p_{i,2}, p_{h,1}+p_{h,2}+p_{i,2} \} \\ \leqq &\max \{ p_{i,1}+p_{h,1}+p_{h,2}, p_{i,1}+p_{i,2}+p_{h,2} \}. \end{aligned}$$

This is easily seen to imply Johnson's Condition.

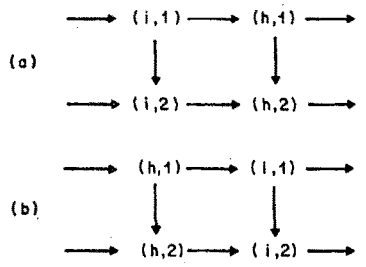

Figure 2. – Critical Paths for Two Machines.

3. EXTENSION OF JOHNSON'S RULE

Johnson's Rule can be extended to obtain an optimal permutation for more than two machines in a straightforward way under rather restrictive conditions. Specifically, we shall prove below that if a permutation exists such that job J_h precedes J_i whenever $\min \{ p_{h,j}, p_{i,k} \} < \min \{ p_{h,k}, p_{i,j} \}$ for some j, k $(1 \leqq j \leqq k \leqq m)$, this permutation must be optimal. Such a permutation clearly will exist if and only if for every pair of jobs J_h and J_i either $\min \{ p_{h,j}, p_{i,k} \} \leqq \min \{ p_{h,k}, p_{i,j} \}$ for $1 \leqq j < k \leqq m$ or the reverse inequality always holds. This case has been studied by several authors [2, 6, 20, 31].

To prove the above mentioned result it suffices to show that in any permutation π, with job J_i immediately preceding job J_h and

$$\min \{ p_{h,j}, p_{i,k} \} < \min \{ p_{h,k}, p_{i,j} \} \qquad \text{for} \quad 1 \leqq j < k \leqq m, \tag{1}$$

the interchange of these jobs does not increase the length of any critical path. To see why this is true, compare the subgraphs $G(h, i)$ and $G(i, h)$ in figure 3. It suffices to show [19] that a critical path from (h, j) to (i, k) in $G(h, i)$ is of no greater weight than a critical path from (i, j) to (h, k) in $G(i, h)$ for all

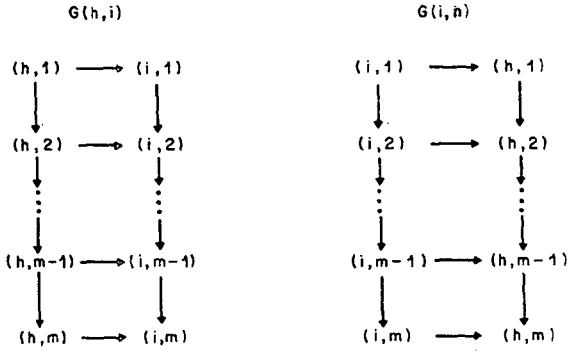

Figure 3. — Critical Paths for Extended Johnson's Rule.

$1 \leqq j < k \leqq m$. Reversing the arrows in $G(i, h)$ to yield the graph $G'(i, h)$ does not change the weight of critical paths and yields the representations $G(h, i)$ and $G'(i, h)$ shown in figure 4. This representation corresponds to a permutation flow-shop problem with two machines and m jobs. By Johnson's Rule and the inequalities in (1), the permutation $(1, 2, \ldots, m)$ of these jobs in $G(h, i)$ yields the shortest critical paths between all pairs (h, j) and (i, k), while the permutation $(m, m-1, \ldots, 1)$ in $G'(i, h)$ yields the longest.

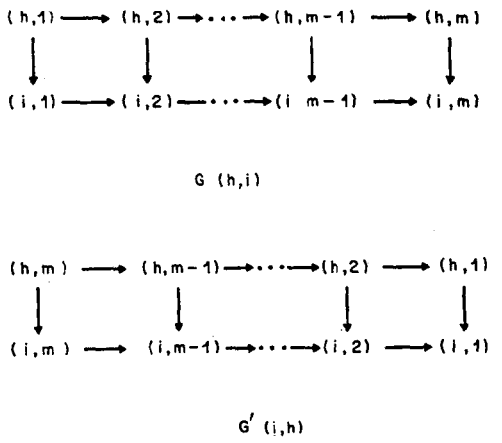

Figure 4. – Revised Critical Paths for Extended Johnson's Rule.

It is often easy to verify that the approach described above applies in a particular case. For example, the case studied by Chin and Tsai [8] requires that for some k with $1 \leqq k \leqq m-1$, $p_{h,j} = p_{h,j+1}$ for $1 \leqq j \leqq k-1$ and $k+1 \leqq j \leqq m-1$. This problem satisfies the above condition, and an optimal schedule is easily obtained by applying Johnson's Rule to the processing times on M_k and M_{k+1}.

A second example is a three-machine problem studied by Szwarc [38]. Here a permutation π is assumed to exist satisfying:

$$p_{\pi(1),1} \leqq p_{\pi(2),1} \leqq \cdots \leqq p_{\pi(n),1},$$

$$p_{\pi(1),2} = p_{\pi(2),2} = \cdots = p_{\pi(n),2},$$

and

$$p_{\pi(1),3} \geqq p_{\pi(2),3} \geqq \cdots \geqq p_{\pi(n),3}.$$

The permutation π is again optimal by virtue of our previous argument.

We note that (1) requires a condition on all pairs of machines. One may attempt to reduce this condition to only consecutive pairs of machines by supposing that for every pair of jobs J_h and J_i either

$$\min \{ p_{h,j}, p_{i,j+1} \} \leqq \min \{ p_{h,j+1}, p_{i,j} \} \text{ for } 1 \leqq j \leqq m-1$$

or that the reverse inequality always holds. This case was studied for $m=3$ by Burns and Rooker [5] and Szwarc [39]; see [3] for the pitfalls surrounding such a simplification.

4. NONBOTTLENECK MACHINES AND MACHINE DOMINANCE

We shall now show how further extensions of Johnson's result have been obtained by relaxing the requirement that at most one job at a time can be processed on each machine. For example, machine M_j may be capable of processing any number of jobs simultaneously; such a machine is called a *nonbottleneck machine.* Three important properties of nonbottleneck machines are collected in the following lemma. Their proof is straightforward.

LEMMA 1: (i) *If M_j and M_{j+1} are nonbottleneck machines, they can be replaced by a single nonbottleneck machine with processing time for each job J_i equal to $p_{i,j}+p_{i,j+1}$.*

(ii) *If M_1 is a nonbottleneck machine, it can be replaced by the constraint that each job J_i does not start processing on M_2 before its* release date $r_i=p_{i,1}$.

(iii) *If M_m is a nonbottleneck machine, it can be replaced by defining a* tail $q_i=p_{i,m}$ *for each job J_i and taking $C_{\pi(i),m}=C_{\pi(i),m-1}+q_{\pi(i)}$.*

Lemma 1 provides a mean of removing certain nonbottleneck machines from the problem. This transformation may simplify matters considerably as demonstrated by Theorem 2.

THEOREM 2 [13, 22]: *An optimal permutation for the three-machine problem where M_2 is a nonbottleneck machine can be obtained by applying Johnson's Rule to the two-machine problem with processing times $p'_{i,1}=p_{i,1}+p_{i,2}$ and $p'_{i,2}=p_{i,2}+p_{i,3}$.*

Proof: For any permutation π for the three-machine problem,

$$C_{\max}(\pi)=\max_{1\leq h\leq n}\left\{\sum_{i=1}^{h}p_{\pi(i),1}+p_{\pi(h),2}+\sum_{i=h}^{n}p_{\pi(i),3}\right\}$$

$$=\max_{1\leq h\leq n}\left\{\sum_{i=1}^{h}p'_{\pi(i),1}+\sum_{i=h}^{n}p'_{\pi(i),2}\right\}-\sum_{i=1}^{n}p_{\pi(i),2}.$$

Since the last term is sequence independent, this completes the proof. □

The proof of Theorem 2 indicates that the crucial property of a nonbottleneck machine M_j is that for every permutation π at least one critical path contains exactly one vertex of type $(\pi(i), j)$. If we can establish conditions under which this property holds, then machine M_j can be treated as though it were a nonbottleneck machine and the resulting problem may be efficiently solvable.

We are now in a position to introduce the concept of machine dominance which will be used to identify machines that can be treated as nonbottleneck machines. The first type of dominance arises when all the processing times on one machine are at least as large as all the processing times on another machine. We say that a machine M_j *dominates* M_k, denoted by $M_j > M_k$, whenever:

$$\min_{1 \leqq i \leqq n} \{p_{i,j}\} \geqq \max_{1 \leqq i \leqq n} \{p_{i,k}\}. \tag{2}$$

THEOREM 3: *If $M_{j+1} > M_j$ or $M_{j-1} > M_j$ then M_j can be treated as a nonbottleneck machine.*

Proof: We shall prove the theorem for the case $M_{j+1} > M_j$ by showing that for any permutation π, some critical path contains only one vertex of the form $(\pi(i), j)$ for $1 \leqq i \leqq n$. The proof is similar for $M_{j-1} > M_j$. Consider a critical path containing the subpath CP_j shown in figure 5 (*a*), i. e., more than one vertex associated with M_j is on the critical path. Consider replacing the subpath CP_j by the subpath CP_{j+1} shown in figure 5 (*b*). The net change in the weight of the original critical path,

$$\sum_{i=i_1}^{i_2-1} p_{\pi(i),j+1} - \sum_{i=i_1+1}^{i_2} p_{\pi(i),j},$$

is nonnegative since $M_{j+1} > M_j$. Therefore, this yields a new critical path of the desired form. □

We note that the dominance condition (2) can be generalized to cover the case where there exists an integer q such that the sum of every q processing times on M_j is at least as great as the sum of every q processing times on M_k. For fixed q and $m = 3$, algorithms have been obtained [3] whose running times are $O(n^q)$.

The type of dominance defined by (2) requires that all job processing times on one machine dominate all job processing times on another machine. A second type of dominance can be defined using only information about individual jobs. We say that machines M_{j-1} and M_{j+1} *jointly dominate* M_j if:

$$p_{i,j-1} \geqq p_{i,j} \leqq p_{i,j+1} \qquad \text{for} \quad 1 \leqq i \leqq n. \tag{3}$$

THEOREM 4: *If M_{j-1} and M_{j+1} jointly dominate M_j, then M_j can be treated as a nonbottleneck machine.*

Proof: We shall prove the theorem, as before, by showing that for any permutation π, some critical path contains only one vertex of the form $(\pi(i), j)$

for $i \leqq i \leqq n$. Consider a critical path containing the subpath CP_j shown in figure 5 (*a*). We claim that replacing CP_j by the longer of the subpaths CP_{j+1} and CP_{j-1} shown in figure 5 yields a new critical path of the desired form. To see this define $w(CP)$ to be the weight of the vertices on the subpath CP and note that:

$$\begin{aligned}(w(CP_{j+1})-w(CP_j))+(w(CP_{j-1})-w(CP_j)) \\ =\left(\sum_{i=i_1}^{i_2-1} p_{\pi(i),j+1}-\sum_{i=i_1+1}^{i_2} p_{\pi(i),j}\right) \\ +\left(\sum_{i=i_1+1}^{i_2} p_{\pi(i),j-1}-\sum_{i=i_1}^{i_2-1} p_{\pi(i),j}\right) \\ =\sum_{i=i_1}^{i_2-1}(p_{\pi(i),j+1}-p_{\pi(i),j})+\sum_{i=i_1+1}^{i_2}(p_{\pi(i),j-1}-p_{\pi(i),j}) \geqq 0,\end{aligned}$$

where the last inequality follows from (3). Hence,

$$W(CP_j) \leqq \max\{W(CP_{j+1}), W(CP_{j-1})\}. \quad \square$$

A final type of dominance generalizes (2). We say that M_{j-1} and M_{j+1} *convexly dominate* M_j if there is a λ, $0 \leqq \lambda \leqq 1$, such that:

$$p_{g,j} \leqq \lambda p_{h,j-1}+(1-\lambda)p_{i,j+1}, \tag{4}$$

for all jobs g, h and i.

THEOREM 5: *If M_{j-1} and M_{j+1} convexly dominate M_j, then M_j can be treated as a nonbottleneck machine.*

Proof: It follows from convex dominance that:

$$\lambda p_{h,j}+(1-\lambda)p_{i,j} \leqq \max\{p_{h,j}, p_{i,j}\}\lambda p_{h,j-1}+(1-\lambda)p_{i,j+1},$$

so that

$$\lambda(p_{h,j-1}-p_{h,j})+(1-\lambda)(p_{i,j+1}-p_{i,j}) \geqq 0,$$

for all jobs h and i. By again referring to figure 5 we can show that

$$\lambda(w(CP_{j-1})-w(CP_j))+(1-\lambda)(w(CP_{j+1})-w(CP_j)) \geqq 0,$$

yielding the desired result. $\square$

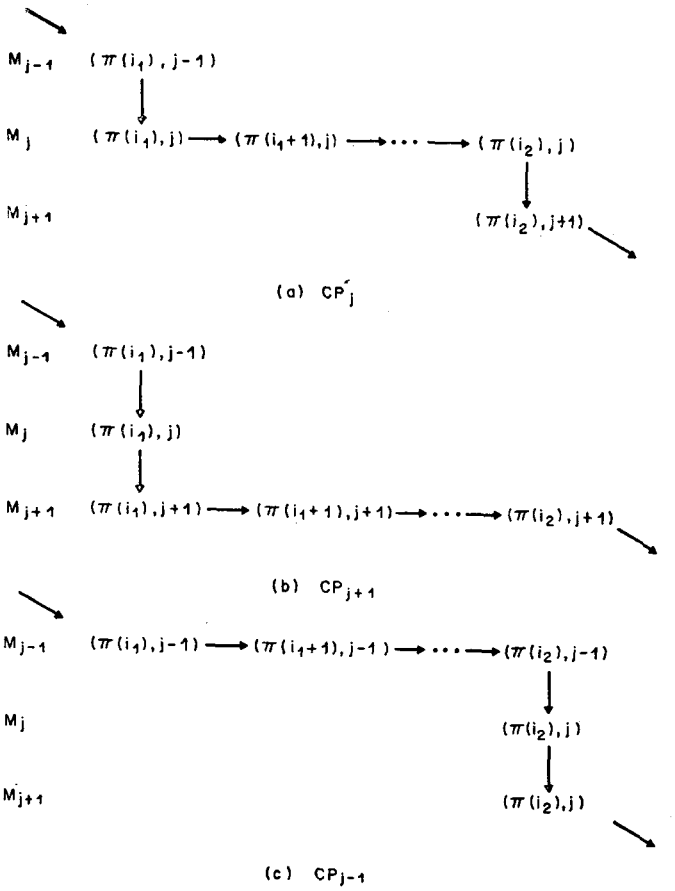

Figure 5. — Critical Paths for Theorem 5.

Lemma 1 can be combined with Theorems 2-5 in many ways to obtain efficient algorithms for a host of polynomially solvable special cases of the permutation flow-shop problem. In this connection the following observations are useful.

(i) By Lemma 1 and Theorem 3, if $M_1 < M_2 < \ldots < M_k$ then $M_1, \ldots, M_{k-1}$ can be replaced by a release date r_i for each job J_i on M_k as in Lemma 1. Furthermore, since these release dates satisfy the inequality $r_i \leqq r_h + p_{h,k}$ for all jobs J_i and J_h, it follows that whenever a job has been scheduled first, all other jobs are available to be processed as soon as the chosen job completes processing. Thus, the release dates can be eliminated by considering n problems each with $(n-1)$ jobs and $(m-k+1)$ machines, corresponding to all possible choices of jobs to be scheduled first.

(ii) Similarly, if $M_k > M_{k+1} > \ldots > M_m$, then $M_{k+1}, \ldots, M_m$ can be replaced by a tail q_i for each job J_i on M_k as indicated in Lemma 1. Since these tails satisfy the inequality $q_i \leqq q_h + p_{h,k}$ for all jobs J_i and J_h, they can be removed from the problem by considering all possible choices of jobs to be scheduled last.

These simple observations account for many special cases of the permutation flow-shop problem that have appeared in the literature. A summary appears in Table 1.

TABLE

Summary of Special Cases Based on Machine Dominance

Reference	Special Cases
[37]	$m = 3$, $M_1 \lessdot M_2 \gtrdot M_3$
[4]	$m = 3$, $M_1 \lessdot M_2$ or $M_3 \lessdot M_2$
[12]	$m = 3$, $M_1 \gtrdot M_2$ or $M_3 \gtrdot M_2$
[7], [40]	$m = 3$, M_1 and M_3 jointly dominate M_2
[27]	$M_j \gtrdot M_{j+1}$ $1 \leq j \leq m-2$, or $M_j \lessdot M_{j+1}$ $2 \leq j \leq m-1$
[11]	$M_j \gtrdot M_{j+1}$ $2 \leq j \leq m-1$, or $M_j \lessdot M_{j+1}$ $1 \leq j \leq m-2$
[11], [28]	$M_j \gtrdot M_{j+1}$ $1 \leq j \leq k-1$, and $M_j \lessdot M_{j+1}$ $k+1 \leq j \leq m-1$
[32]	$M_j \lessdot M_{j+1}$ $1 \leq j \leq k-1$, and $M_j \gtrdot M_{j+1}$ $k+1 \leq j \leq m-1$

We note that *start lags* and *stop lags* between machines M_j and M_{j+1} can be viewed as arising out of processing that has to be done on an intermediate nonbottleneck machine [33]. Thus, the efficiently solvable cases that arise in this context may be viewed as further examples of the above approach [13, 21, 22, 29].

We conclude our survey by showing how the ideas presented thus far apply to so-called *ordered flow-shop* problems [35, 36]. The jobs of a flow-shop are said to be *ordered* if there exists a permutation π of the jobs such

that $p_{\pi(1),j} \geqq p_{\pi(2),j} \geqq \ldots \geqq p_{\pi(n),j}$ for $1 \leqq j \leqq m$. Similarly, the machines of a flow-shop are said to be *ordered* if there exists a permutation σ of the machines such that:

$$p_{i,\sigma(1)} \geqq p_{i,\sigma(2)} \geqq \ldots \geqq p_{i,\sigma(m)}$$
$$\text{for} \quad 1 \leqq i \leqq n.$$

A job J_h is said to be *larger* than a job J_i when $p_{h,j} \geqq p_{i,j}$ for $1 \leqq j \leqq m$. Similarly, a machine M_j is said to be *larger* than a machine M_k when $p_{i,j} \geqq p_{i,k}$ for $1 \leqq i \leqq n$.

First consider a flow-shop where the jobs are ordered and, in addition, each job requires the most processing on machines M_1 and M_m (i. e., $\min \{ p_{i,1}, p_{i,m} \} \geqq \max \{ p_{i,j} : 2 \leqq j \leqq m-1 \}$ for $1 \leqq i \leqq n$). Using the techniques of this section it is then easy to show that machines $M_2, \ldots, M_{m-1}$ can be treated as nonbottleneck machines [40]. Hence, the problem is solved by applying Lemma 1 and Theorem 2.

A more complicated example arises when the jobs and machines are ordered, and each job requires the most processing on M_1. We shall prove that a permutation π is optimal whenever the jobs are ordered from the largest to the smallest [35].

Let us say that a critical path CP for a permutation π *turns down* at job $J_{\pi(i)}$ if it contains the vertices $(\pi(i-1), j)$, $(\pi(i), j)$ and $(\pi(i), j+1)$ for some j, $1 \leqq j \leqq m-1$, and that it *turns right* at machine M_j if it contains the vertices $(\pi(i), j-1)$, $(\pi(i), j)$ and $(\pi(i+1), j)$ for some i, $1 \leqq i \leqq n-1$.

We claim that for any permutation π there is a critical path CP which turns down at job $J_{\pi(i)}$ only if $J_{\pi(i)}$ is larger than any job following it, and which turns right at machine M_j only if M_j is larger than any machine following it. To see this, consider a critical path CP for π which turns down for the first time at job $J_{\pi(h)}$ and turns to the right for the first time at machine M_j as shown in figure 6. Let job $J_{\pi(i)}$ be the largest job among $J_{\pi(h)}, \ldots, J_{\pi(n)}$ and let machine M_k be the largest machine among $M_j, \ldots, M_m$. Then CP', shown in figure 6, is easily seen to be a critical path as well and, moreover, it is of the proper form with respect to the first turn down and the first turn right. Repeating this process completes the proof of the claim.

We now use the claim to prove that ordering the jobs from largest to smallest is optimal. It suffices to show that any permutation π, if $J_{\pi(i+1)}$ is larger than $J_{\pi(i)}$ then interchanging these adjacent jobs to obtain π' does not increase the overall completion time.

Consider an arbitrary critical path CP' for π' satisfying the above claim. CP' must be of the form shown in figure 7 (a). The corresponding critical path

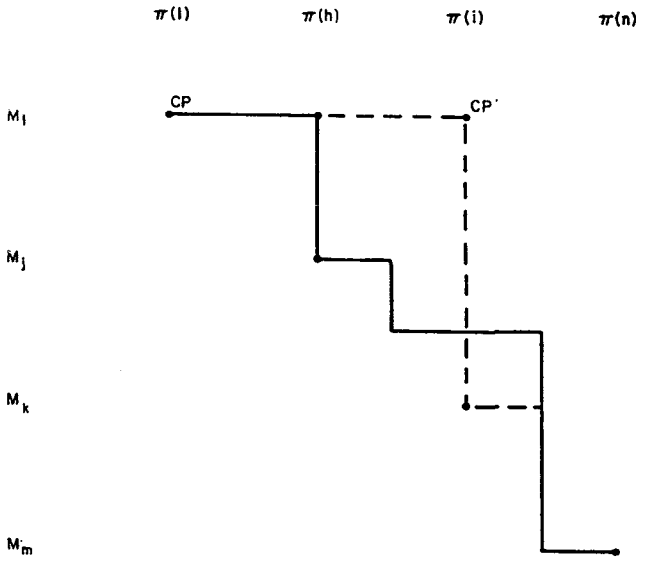

Figure 6. — Critical Path for an Ordered Flow Shop.

CP for π shown in figure 7 (b) is at least as long. Therefore, interchanging $\pi(i)$ and $\pi(i+1)$ does not increase the overall completion time. This completes the proof.

5. CONCLUDING REMARKS

We have demonstrated that many efficiently solvable special cases of the permutation flow-shop problem can indeed be obtained by systematic application of a few simple ideas.

The two machine flow-shop problem has also been solved efficiently subject to precedence constraints on the jobs of a certain type [14, 15, 24, 25, 26, 34]. This approach capitalizes on the fact that Johnson's Rule can be viewed as the outcome of a simple interchange argument that can be adapted to account for these precedence constraints. Hence, these precedence constraints can be included whenever some form of Johnson's Rule can be shown to yield the optimal schedule, so that various extensions to m-machine problems are possible [16, 23, 30].

In conclusion, we note that the above ideas can also be used to generate approximation algorithms [9], e. g. by treating a machine as though it were a nonbottleneck one, even if this is not strictly justified. In the context of

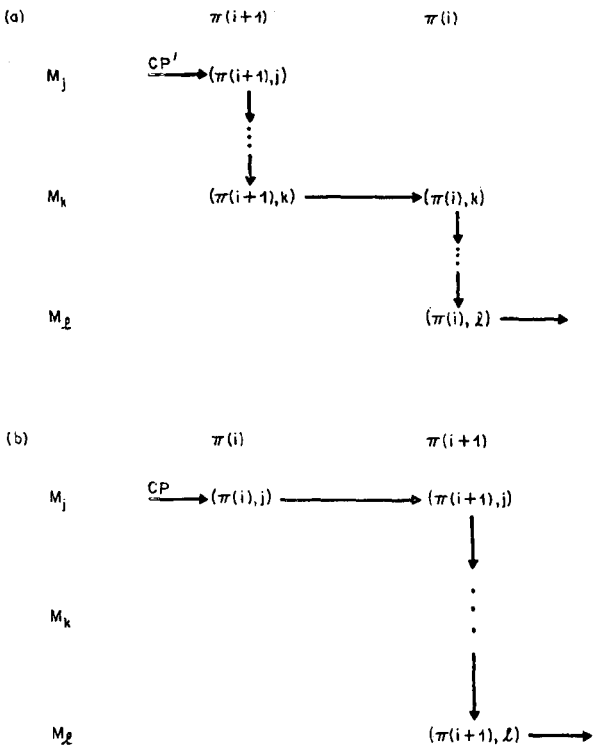

Figure 7. – Adjacent Interchange for an Ordered Flow Shop.

branch-and-bound procedures [17], such an approach yields powerful lower bounds; in addition, as the set of unscheduled jobs is shrinking, it becomes increasingly likely that one of the dominance conditions can be applied to reduce the size of the problem.

REFERENCES

1. J. O. ACHUGBUE and F. Y. CHIN, *Complexity and Solutions of Some Three Stage Flow Shop Scheduling Problems*, Technical Report 79-7, University of Alberta, Canada, 1979.
2. N. R. ACHUTHAN, *A Special Case of the* $(n/m/F/F_{max})$ *problem*, Opsearch, Vol. 14, 1977, pp. 71-87.
3. N. R. ACHUTHAN, *Flow-Shop Scheduling Problems, Ph. D. Thesis*, Indian Statistical Institute, Calcutta, 1980.
4. T. S. ARTHANARY and A. C. MUKHAPADHYAH, *A Note on a Paper*, by W. Szwarc, Naval Research Logistics Quarterly, Vol. 18, 1971, pp. 135-138.

5. F. BURNS and J. ROOKER, *A Special Case of the* $3 \times n$ *Flow-Shop Problem*, Naval Research Logistics Quarterly, Vol. 22, 1975, pp. 811-817.
6. F. BURNS and J. ROOKER, *Johnson's Three Machine Flow-Shop Conjecture*, Operations Research, Vol. 24, 1976, pp. 578-580.
7. F. BURNS and J. ROOKER, *Three-State Flow-Shop with Recessive Second State*, Operations Research, Vol. 26, 1978, pp. 207-208.
8. F. CHIN and L. TSAI, *On J-maximal and J-minimal Flow-Shop Schedules* (to appear).
9. D. G. DANNENBRING, *An Evaluation of Flow-Shop Sequencing Heuristics*, Management Science, Vol. 23, 1977, pp. 1174-1182.
10. M. R. GAREY, D. S. JOHNSON and R. SETHI, *The Complexity of Flow-Shop and Job-shop Scheduling*, Mathematics of Operations Research, Vol. 1, 1976, pp. 117-129.
11. J. N. D. GUPTA, *Optimal Schedules for Special Structure Flow-shops*, Naval Research Logistics Quarterly, Vol. 22, 1975, pp. 255-269.
12. S. M. JOHNSON, *Optimal Two- and Three-Stage Production Schedules with Setup Times Included*, Naval Research Logistics Quarterly, Vol. 1, 1954, pp. 61-68.
13. S. M. JOHNSON, *Discussion: Sequencing n Jobs on Two Machines with Arbitrary Time Lags*, Management Science, Vol. 5, 1959, pp. 299-303.
14. T. KURISU, *Two-Machine Scheduling under Required Precedence Among Jobs*, J. Operations Research Society Japan, Vol. 19, 1976, pp. 1-13.
15. T. KURISU, *Two-Machine Scheduling under Arbitrary Precedence Constraints*, J. Operations Research Society Japan, Vol. 20, 1977, pp. 113-131.
16. T. KURISU, *Three-Machine Scheduling Problem with Precedence Constraints*, J. Operations Research Society Japan, Vol. 20, 1977, pp. 231-242.
17. B. J. LAGEWEG, J. K. LENSTRA and A. H. G. RINNOOY KAN, *A General Bounding Scheme for the Permutation Flow-Shop Problem*, Operations Research, Vol. 26, 1978, pp. 53-67.
18. J. K. LENSTRA, A. H. G. RINNOOY KAN and P. BRUCKER, *Complexity of Machine Scheduling Problems*, Annals Discrete Mathematics, Vol. 1, 1977, pp. 343-362.
19. G. B. MCMAHON, *A Study of Algorithms for Industrial Scheduling Problems, Ph. D. thesis*, University of South Wales, 1971.
20. Y. MING-I., *On the n Job, m Machine Sequencing Problem of Flow-Shop*, Operational Research, Vol. 75, North Holland, 1976, pp. 179-200.
21. L. G. MITTEN, *Sequencing n Jobs on Two Machines with Arbitrary Lag Times*, Management Science, Vol. 5, 1958, pp. 293-298.
22. L. G. MITTEN, *A Scheduling Problem*, J. Industrial Engineering, Vol. 10, 1959, pp. 131-135.
23. C. L. MONMA, *Optimal* $m \times n$ *Flow Shop Sequencing with Precedence Constraints and Lag Times*, Institute of OR, Report 7778, University of Bonn, 1977.
24. C. L. MONMA, *The Two-Machine Maximum Flow Time Problem with Series-Parallel Precedence Constraints: An Algorithm and Extensions*, Operations Research, Vol. 27, 1979, pp. 792-797.
25. C. L. MONMA, *Sequencing to Minimize the Maximum Job Cost*, Operations Research, Vol. 28, 1980, pp. 942-951.
26. C. L. MONMA and J. B. SIDNEY, *Sequencing with Series-Parallel Precedence Constraints*, Mathematics of Operations Research, Vol. 4, 1979, pp. 215-224.
27. I. NABESHIMA, *The Order of n Items Processed on m Machines* [I], J. Operations Research Society Japan, Vol. 3, 1961, pp. 170-175.

28. I. NABESHIMA, *The Order of n Items Processed on m Machines* [II], J. Operations Research Society Japan, Vol. 4, 1961, pp. 1-8.

29. I. NABESHIMA, *Sequencing Two Machines with Start Lag and Stop Lags*, J. Operations Research Society Japan, Vol. 5, 1963, pp. 97-101.

30. I. NABESHIMA, *Some Extensions of the M-Machine Scheduling Problem*, J. Operations Research Society Japan, Vol. 10, 1967, pp. 1-17.

31. I. NABESHIMA, *The Order of n Items Processed on m Machines* (III), J. Operations Research Society Japan, Vol. 16, 1973, pp. 163-184.

32. I. NABESHIMA, *Notes on the Analytical Results in Flow Shop Scheduling:* Part 2, Rep. Univ. Electro-Comm., Vol. 27, 1977, pp. 253-257.

33. A. H. G. RINNOOY KAN, *Machine Scheduling Problems: Classification, Complexity and Computations*, Nijhoff, The Hague, 1976.

34. J. B. SIDNEY, *The Two-Machine Flow Time Problem with Series-Parallel Precedence Constraints*, Operations Research, Vol. 27, 1979, pp. 782-791.

35. M. L. SMITH, S. S. PANWALKER and R. A. DUDEK, *Flow Shop Sequencing with Ordered Processing Time Matrices*, Management Science, Vol. 21, 1975, pp. 544-549.

36. M. L. SMITH, S. S. PANWALKER and R. A. DUDEK, *Flow Shop Sequencing Problem with Ordered Processing Time Matrices: A General Case*, Naval Research Logistics Quarterly, Vol. 23, 1976, pp. 481-486.

37. W. SZWARC, *On Some Sequencing Problems*, Naval Research Logistics Quarterly, Vol. 15, 1968, pp. 127-155.

38. W. SZWARC, *Mathematical Aspects of the* $3 \times n$ *Job-Shop Sequencing Problem*, Naval Research Logistics Quarterly, Vol. 21, 1974, pp. 145-153 (*see* also pp. 725-726).

39. W. SZWARC, *Optimal Two-Machine Orderings in the* $3 \times n$ *Flow-Shop Problem*, Operations Research, Vol. 25, 1977, pp. 70-77.

40. W. SZWARC, *Special Cases for the Flow-Shop Problem*, Naval Research Logistics Quarterly, Vol. 25, 1977, pp. 70-77.

41. W. SZWARC, *Permutation Flow-Shop Theory Revisited*, Naval Research Logistics Quarterly, Vol. 26, 1978, pp. 557-570.

42. W. SZWARC, *The Critical Path Approach In the Flow-Shop Problem*, Opsearch, Vol. 16, 1979, pp. 98-102.