Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

J. Han, M. Yue

A study of elimination conditions for the permutation flow-shop sequencing problem

Department of Operations Research and System Theory Report OS-R8403 January

A STUDY OF ELIMINATION CONDITIONS FOR THE PERMUTATION FLOW-SHOP SEQUENCING PROBLEM

J. HAN, M. YUE
Institute of Applied Mathematics, Academia Sinica,
Beijing, People's Republic of China

We give a few elimination criteria for the permutation flow-shop problem and prove that one of them is equivalent to Szwarc's elimination criterion. Next we propose a lower bound to be used in a branch-and-bound method.

1980 MATHEMATICS SUBJECT CLASSIFICATION: 90B35.
KEYWORDS & PHRASES: flow-shop, elimination criterion, branch-and-bound.
NOTE: This paper received its final form during the first author's stay at the Centre for Mathematics and Computer Science from October 23, 1983 until January 15, 1984.

Report OS-R8403
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

0. INTRODUCTION

This paper is concerned with the permutation flow-shop sequencing problem of determining an optimal sequence of jobs. The permutation flow-shop problem can be formulated as follows. Each of n jobs $J_1, J_2, \ldots, J_n$ has to be processed on m machines $M_1, M_2, \ldots, M_m$ in that order. Thus job J_i $(i=1,2,\ldots,n)$ consists of a sequence of m operations $O_{i1}, \ldots, O_{im}$; O_{ik} corresponds to the processing of J_i on M_k during an uninterrupted processing time a_{ki}. M_k $(k=1,2,\ldots,m)$ can handle at most one job at a time, and it is assumed that each machine processes the jobs in the same order. We want to find a processing order such that the time required to complete all jobs is minimized. The most common methods to solve problems of this type are branch-and-bound methods and elimination methods. A variety of elimination conditions has been developed, such as Szwarc's optimal elimination criteria [17,18] and the elimination criteria in [7,12,16]. But, as pointed out by Baker in [2], enumerative methods based on these elimination criteria are not as efficient as branch-and-bound methods. In the following, we first establish some new elimination criteria and prove that these criteria include Szwarc's as a special case, and then we propose a lower bound. By combining these elimination criteria with the lower bound, an enumerative algorithm will be obtained.

1. ELIMINATION CRITERIA

Let $s = (s_1, s_2, \ldots, s_k)$ be a partial schedule of jobs. Any permutation $\bar{s} = (\bar{s}_1, \ldots, \bar{s}_{n-k})$ of the index set of unscheduled jobs defines a completion of s, i.e. a complete permutation schedule $s\bar{s} = (s_1, s_2, \ldots s_k, \bar{s}_1, \ldots, \bar{s}_{n-k})$. Elimination criteria are certain conditions under which all completions of a partial schedule s' can be eliminated because a schedule at least as good exists among the completions of another partial schedule s". Before we suggest the new elimination conditions, we describe a useful lemma, which is given in our previous paper [20].

Let $N = \{1,2,\ldots,n\}$, let $w = (w_1, w_2, \ldots, w_n)$ be a permutation of N, and define the matrix A(w) by

$$A(w)=\begin{pmatrix} a_{1w_1} & a_{1w_2} & \cdots & a_{1w_n} \\ a_{2w_1} & a_{2w_2} & \cdots & a_{2w_n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{mw_1} & a_{mw_2} & \cdots & a_{mw_n} \end{pmatrix}.$$

A broken line with starting point a_{1w_1} and ending point a_{mw_n} is called a feasible line of matrix $A(w)$, if each of its vertices is located at an a_{ij}, and each of its segments is either horizontal rightwards or vertical downwards. For example, the broken line connecting $a_{1w_1}, a_{2w_2}, a_{2w_2}, a_{mw_n}$ is a feasible line.

For a fixed permutation w, the set of all feasible lines is

$$\{\ell(w)\} = \{(a_{1w_1},\ldots,a_{1w_{j_1}}, a_{2w_{j_2}},\ldots,a_{2w_{j_2}},\ldots,a_{mw_{j_{m-1}}},\ldots,a_{mw_n}),$$
$$1 \le j_1 \le j_2 \le \ldots \le j_m = n\}.$$

Corresponding to a feasible line $\ell(w)$, the sum $\Sigma_{a_{ki}\in\ell(w)} a_{ki}$ is called a feasible sum corresponding to $\ell(w)$.

LEMMA 1. *Let* $w = (w_1,w_2,\ldots,w_n)$ *be a permutation of* N. *Assume that on each machine the jobs are processed in the order* $J_{w_1},\ldots,J_{w_n}$. *Let* $t(w)$ *be the completion time corresponding to* w, *i.e., the completion time of the last job* J_{w_n} *on* M_m. *Then we have*

$$t(w) = \max_{\ell(w)} \sum_{a_{ki}\in\ell(w)} a_{ki},$$

where $\ell(w)$ *runs through all feasible lines corresponding to* w.

Let $s = (s_1,s_2,\ldots,s_k)$ be a partial schedule of N, $s_i \neq s_j$ if $i \neq j$ ($s = \phi$ if $k = 0$), and for $1 \le p \le q \le m$, let $t_{pq}(s)$ be the maximal feasible sum of the matrix

$$\begin{pmatrix} a_{ps_1} & a_{ps_2} & \cdots & a_{ps_k} \\ a_{p+1s_1} & a_{p+1s_2} & \cdots & a_{p+1s_k} \\ \cdots & \cdots & \cdots & \cdots \\ a_{qs_i} & a_{qs_2} & \cdots & a_{qs_k} \end{pmatrix}.$$

Define $t_{pq}(s) = 0$, if $s = \phi$.

The following theorem indicates whether the job J_j could be put on the (k+1)th place when $s = (s_1,\ldots,s_k)$ is fixed.

THEOREM 1. *Let* $s = (s_1,s_2,\ldots,s_k)$, $I = (i_1,i_2,\ldots,i_u)$ *are two partial schedules of* N, $I \cap s = \phi$, $j \bar{\in} I \cup s$. *If*

(1) $$t_{1q}(sI) \le t_{1q}(sj) - a_{qj} + A_{qm}(I), \quad q = 1,2,\ldots,m,$$

where

$$A_{qm}(I) = \min_{q \le r_1 \le \ldots \le r_k \le m} (a_{r_1 i_1} + \ldots + a_{r_k i_k}),\ sI = (s_1,s_2,\ldots,s_k,i_1,\ldots,i_u),$$

$sj = (s_1,s_2,\ldots,s_k,j)$, *then for finding a optimal sequence we can eliminate all permutations of form* $(sj,\ldots,i_1,\ldots,i_2,\ldots,i_u,\ldots)$.

PROOF. Let w' be a permutation of the form $(sj,\ldots,i_1,\ldots,i_u,\ldots)$. Let $R_1,R_2,\ldots$ denote the partial sequences between j and i_1, between i_1 and i_2, etc., i.e., w' has the form $(sjR_1i_1R_2\ldots i_uR_{u+1})$. Let permutation $w = (sIjR_1R_2\ldots R_{u+1})$, and $\ell(w)$ be any feasible line in the matrix A(w). By Lemma 1, it is easily seen that there exist integers $q,r_1,r_2,\ldots,r_u$, $1 \le q \le r_1 \le r_2 \le \ldots \le r_u \le m$, such that the feasible sum t_ℓ corresponding to $\ell(w)$ satisfies

(2) $$t_\ell \le t_{1q}(sI) + t_{qr_1}(jR_1) + t_{r_1r_2}(R_2) + \ldots + t_{r_um}(R_{u+1}).$$

By (1), we have

(3) $$t_\ell \le t_{1q}(sj) - a_{qj}(I) + t_{qr_1}(jR_1) + t_{r_1r_2}(R_2) + \ldots + t_{r_um}(R_{u+1}).$$

By Lemma 1, we obtain

$$t_{1q}(sj) - a_{qj} + t_{qr_1}(jR_1) \le t_{1r_1}(sjR_1),$$

$$\begin{aligned} A_{qm}(I) &+ t_{r_1r_2}(R_2) + \dots + t_{r_um}(R_{u+1}) \\ &\le a_{r_1i_1} + a_{r_2i_2} + \dots + a_{r_ui_u} + t_{r_1r_2}(R_2) + \dots + t_{r_um}(R_{u+1}) \\ &\le t_{r_1m}(i_1R_2i_2R_3\dots i_uR_{u+1}). \end{aligned}$$

Substituting into (3), we have

$$\begin{aligned} t_\ell &\le t_{1r_1}(sjR_1) + t_{r_1m}(i_1R_2i_2R_3\dots i_uR_{u+1}) \\ &\le t_{1m}(sjR_1i_1R_2i_2\dots i_uR_{u+1}) = t(w'). \end{aligned}$$

Since $\ell(w)$ is any feasible line, we have $t(w) \le t(w')$. It follows that if we eliminate all permutations of the form $(sj\dots i_1\dots i_2\dots i_u)$, there is still an optimal sequence left in the remainder. □

If $u = 2$, there are only two cases, i.e., $I = (i_1,i_2)$ and $I = (i_2,i_1)$, to be examined in (1). If for these two cases conditions (1) hold, then we can eliminate all permutations of the form $(sj\dots)$.

If $u = 1$, then putting $I = (i)$ in Theorem 1 and noticing that

(4) $$A_{qm}(i) = \min(a_{qi}, a_{q+1\,i}, \dots, a_{mi}),$$

we have the following.

COROLLARY 1. *Let* $s = (s_1,\dots,s_k)$, $\{i,j\} \cap s = \phi$. *If*

(5) $$t_{1q}(si) \le t_{1q}(sj) - a_{qj} + \min(a_{qi},\dots,a_{mi}), \quad 1 \le q \le m,$$

we can eliminate all permutations of the form $(sj\dots)$.

In [17,18], Szwarc established the following elimination criteria:

(6) $\quad t_{1q-1}(sij) - t_{1q-1}(sj) \le t_{1q}(sij) - t_{1q}(sj) \le a_{qi}, \ q = 2,3,\ldots,m.$

He proved that if (6) holds, then $t_{1m}(sijR'R'') \le t_{1m}(sjR'iR'')$, where R' and R" are any two partial sequences of N such that

$$R' \cap R'' = \phi, \quad \{R'R''\} \cap \{sij\} = \phi, \ R' \cup R'' \cup \{sij\} = N.$$

Now we are going to prove that (6) is equivalent to (5).

THEOREM 2. *The set of conditions (5) is equivalent to the set of conditions (6).*

PROOF. First we prove that if (6) holds, then we have (5). Clearly, $t_{11}(sij)-t_{11}(sj) = a_{1i}$. If (6) holds, we have

(7) $\quad a_{1i} \le a_{2i}, \ a_{1i} \le a_{3i},\ldots, a_{1i} \le a_{mi}.$

From this we have

$$t_{11}(si) \le t_{11}(s) + \min(a_{1i},a_{2i},\ldots,a_{mi}).$$

This proves that (5) is true for q = 1. Now we proceed by induction. For m = 2, from (6), we have

(8) $\quad t_{12}(sij) \le a_{2i} + t_{12}(sj).$

By the definition of t_{pq}, we have

(9) $\quad t_{12}(sij) = \max\{t_{11}(sij), t_{12}(si)\} + a_{2j},$

(10) $\quad t_{12}(si) = \max\{t_{11}(si), t_{12}(s)\} + a_{2i}.$

Substituting (9) into (8), we have

$$t_{12}(si) \le t_{12}(sj) - a_{2j} + a_{2i}.$$

This proves that (5) holds for $m = 2$ and $q = 1,2$. Now we assume that (5) can be proved from (6) for all integers less than m (m>2), and prove that (5) can be proved also from (6) for m.

By the induction hypothesis, we have

$$(11) \qquad t_{1q}(si) \le t_{1q}(sj) - a_{qj} + \min\{a_{qi}, a_{q+1i}, \ldots, a_{m-1i}\}, \quad 1 \le q \le m-1.$$

In order to prove that (5) holds we only have to prove

$$(12) \qquad t_{1q}(si) \le t_{1q}(sj) - a_{qj} + a_{mi}, \quad 1 \le q \le m.$$

By (6), we have

$$(13) \qquad t_{1q}(sij) \le t_{1q}(sj) + a_{mi}.$$

Then we have

$$t_{1q}(si) + a_{qj} \le t_{1q}(sij) \le t_{1q}(sj) + a_{mi},$$

from this, (12) can be obtained. So (5) holds.

Next, we prove that if (5) holds, then (6) is true. We also proceed by induction. For $m = 2$, from (5), we have

$$(14) \qquad t_{11}(si) \le t_{11}(sj) - a_{1j} + \min\{a_{1i}, a_{2i}\},$$

$$(15) \qquad t_{12}(si) \le t_{12}(sj) - a_{2j} + a_{2i}.$$

(14) is equivalent to $a_{1i} \le a_{2i}$. From this we have

$$t_{11}(sij) + a_{2j} = t_{11}(sj) + a_{1i} + a_{2j} \le t_{12}(sj) + a_{2i};$$

from this and (15) we obtain

$$(16) \qquad t_{12}(sij) = \max\{t_{12}(si), t_{11}(sij)\} + a_{2j} \le t_{12}(sj) + a_{2i}.$$

Since

(17) $$t_{11}(sij) - t_{11}(sj) + t_{12}(sj) = \max\{t_{12}(s)+a_{1i}, t_{11}(sij)\} + a_{2j},$$

using $a_{1i} \leq a_{2i}$, we have

$$t_{12}(s) + a_{2j} + a_{1i} \leq t_{12}(si) + a_{2j}.$$

Substituting this into (17), we obtain

(18) $$t_{11}(sij) - t_{11}(sj) + t_{12}(sj) \leq t_{12}(sij).$$

Combining (18) with (16), we have

$$t_{11}(sij) - t_{11}(sj) \leq t_{12}(sij) - t_{12}(sj) \leq a_{2i},$$

So, (6) holds for m = 2. Now we assume that (6) can be proved from (5) for all integers less than m (m>2), and prove that (6) can be proved also from (5) for m.

Since (5) holds for m, we have

$$t_{1q}(si) \leq t_{1q}(sj) - a_{qj} + \min\{a_{qi}, a_{q+1i}, \ldots, a_{m-1i}\}, \quad q = 1,2,\ldots,m-1.$$

By the induction hypothesis, it can be shown that

(19) $$t_{1q-1}(sij) - t_{1q-1}(sj) \leq t_{1q}(sij) - t_{1q}(sj) \leq a_{q-1}a_{qi}, \quad q = 2,3,\ldots,m-1.$$

In order to prove that (6) holds for m, we only have to prove

(20) $$t_{1m-1}(sij) - t_{1m-1}(sj) \leq t_{1m}(sij) - t_{1m}(sj) \leq a_{mi}.$$

Before this we are going to prove the following inequalities for q = 1,2,...,m by induction:

$$\text{(21)} \qquad t_{1q}(sij) - t_{1q}(sj) \leq a_{mi}.$$

When $q = 1$, since we have $a_{1i} \leq a_{mi}$ from (5), therefore (21) is true. Now we assume that (21) is true for all integers less than r $(1<r\leq m)$, and prove that (21) is also true for r. By the induction hypothesis we have

$$t_{1r-1}(sij) - t_{ir}(sj) + a_{rj} \leq t_{1r-1}(sij) - t_{1r-1}(sj) \leq a_{mi}.$$

From (5) we have

$$t_{1r}(si) - t_{1r}(sj) + a_{rj} \leq \min\{a_{ri}, a_{r+1i}, \ldots, a_{mi}\} \leq a_{mi}.$$

Using the two above inequalities, we obtain

$$t_{1r}(sij) - t_{1r}(sj) \leq a_{mi}.$$

Then (21) has been proved for $q = 1,2,\ldots,m$. So the first inequality in (20) is true. We want to prove the second inequality in (20). From (21) we have

$$\begin{aligned} t_{1m-1}(sij) - t_{1m-1}(sj) + t_{1m}(sj) \\ &= \max\{t_{1m}(s)-t_{1m-1}(sj), 0\} + a_{mj} + t_{1m-1}(sij) \\ &\leq \max\{t_{1m}(s)+a_{mi}, t_{1m-1}(sij)\} + a_{mj} \\ &\leq t_{1m}(sij), \end{aligned}$$

which proves the second inequality in (20). The proof of the theorem is complete.

For a given partial schedule $s = (s_i,\ldots,s_k)$, when we use the set of conditions (5) to find all nodes corresponding to $(sj\ldots)$ which can be eliminated, we have to calculate $t_{1q}(s\alpha)$, $\forall\alpha \in N\backslash s$, $q = 1,2,\ldots,m$ by

$$t_{1q}(s\alpha) = \max\{t_{1q}(s),t_{1q-1}(s\alpha)\} + a_{q\alpha},\ \forall\alpha \in N\backslash s,\ q = 1,\ldots,m,$$

which requires $O(m(n-k))$ calculates. To check (5) for all i and j, we need $O(m(n-k)^2)$ calculations. If we use the set of conditions (6) to find all nodes of the form (sj...) which can be eliminated, we have to calculate $t_{1q}(s\alpha)$ as above and $t_{1q}(s\alpha\beta)$, $\alpha,\beta \in N\backslash s$, $\alpha \neq \beta$, $q = 1,\ldots,m$ by

$$t_{1q}(s\alpha\beta) = \max\{t_{1q}(s\alpha),t_{1q-1}(s\alpha\beta)\} + a_{q\beta},$$
$$\forall\alpha,\beta \in N\backslash s,\ \alpha \neq \beta,\ q = 1,\ldots,m,$$

which requires $O(m(n-k)^2)$ calculations. To check (6) for all i and j, we also need $(O(m(n-k)^2)$ calculations. Altogether, it seems simpler to use (5) than to use (6).

A simple special case of conditions (5) can be obtained as follows.

COROLLARY 2. *Let* $s = (s_1,s_2,\ldots,s_k)$, $i \neq j$, $\{i,j\} \cap s = \phi$. *If*

(22) $\qquad a_{1i} \le a_{2i} \le \ldots \le a_{mi}$, *and* $a_{ri} \le a_{rj}$, $\quad r = 1,\ldots,m-1,$

we can eliminate all permutations of the form (sj...).

THEOREM 3. *Let* $s = (s_1,s_2,\ldots,s_k)$, $i \neq j$, $\{i,j\} \cap s = \phi$. *If*

(23) $\qquad t_{1q}(si) \le t_{1q}(sj) + \min_{q\le r\le u\le m} \{\sum_{k=r}^{u} (a_{ki}-a_{kj})\},\quad 1 \le q \le m,$

we can eliminate all permutations of the form (sj...).

PROOF. Let $w' = (sjR_1iR_2)$ be a permutation of form (sj...), and let $w = (siR_1jR_2)$. Suppose that $\ell(w)$ is a feasible line of matrix A(w). Clearly, there exist integers q,r and u with $1 \le q \le r \le u \le m$, such that the feasible sum t_ℓ corresponding to $\ell(w)$ satisfies

(24) $\qquad t_\ell \le t_{1q}(si) + t_{qr}(R_1) + t_{ru}(j) + t_{um}(R_2).$

Since,

$$t_{ru}(j) + \min_{q\le r\le u\le m} \{\sum_{k=r}^{u} (a_{ki}-a_{kj})\} \le t_{ru}(i),$$

if conditions (23) hold, we have

$$t_\ell \le t_{1q}(sj) + t_{qr}(R_1) + t_{ru}(i) + t_{um}(R_2) \le t_{1m}(w').$$

By Lemma 1, we have $t_{1m}(w) \le t_{1m}(w')$. □

We give a simple special case of condition (23) as follows.

COROLLARY 3. *Let* $s = (s_1,s_2,\ldots,s_k)$, $i \ne j$, $\{i,j\} \cap s = \phi$. *For* $1 \le q \le m$, *let* $Q_q = \{k \mid a_{ki} \le a_{kj},\ q \le k \le m\}$. *If*

(25) $$t_{1q}(si) \le t_{1q}(sj) + \min\{\sum_{k\in Q_q} (a_{ki}-a_{kj}), (a_{ui}-a_{uj}), q\le u\le m\}, \quad 1 \le q \le m,$$

we can eliminate all permutations of the form (sj...). (*If* $Q_q = \phi$, *define* $\Sigma_\phi(a_{ki}-a_{kj}) = +\infty$.)

2. LOWER BOUND

Branch-and-bound algorithms are commonly used for solving permutation flow-shop sequencing problems. For a given partial schedule $s = (s_1,\ldots,s_k)$, we want to compute a lower bound on the value of all possible completions $s\bar{s}$ of s, where $\bar{s}$ is a permutation of all unscheduled jobs. Several formulae to compute a lower bound have been presented in the literature, for example, the "machine-based bound" [13,19], etc. In [9] Lageweg, Lenstra and Rinnooy Kan give a general bounding scheme, that generates most previously known bounds and leads to some new bounds. Computational experiments show that the "two-machine bounds" developed in [9] are superior to previous bounds in solving permutation flow-shop problems.

We propose another two-machine bound. For a given partial schedule $s = (s_1,s_2,\ldots,s_k)$, let σ be the index set of unscheduled jobs, i.e.

$\sigma = N\backslash s$. Arranging all jobs belonging to σ in the sequence $(j_1, j_2, \ldots, j_{n-k})$ by applying Johnson's rule to a_{pj}, $\forall j \in \sigma$, we define

$$R_1(p) = (j_2, j_3, \ldots, j_{n-k}, j_1)$$

$$R_i(p) = (j_1, \ldots, j_{i-1}, j_{i+1}, \ldots, j_{n-k}, j_i), \quad i = 2, \ldots, n-k-1,$$

$$R_{n-k}(p) = (j_1, j_2, \ldots, j_{n-k}),$$

and

$$b_p = t_{1p}(s) + \min_{1\le i\le n-k} \{t_{pp+1}(R_i(p)) + \sum_{r=p+2}^{m} a_{rj_i}\}, \quad p = 1, \ldots, m-2,$$

$$(26) \qquad b_{m-1} = t_{1m-1}(s) + t_{m-1m}(R_{n-k}(m-1)),$$

$$b_m = t_{1m}(s) + \sum_{j\in\sigma} a_{mj}.$$

Suppose that $(j'_1, \ldots, j'_{n-k})$ is any permutation of jobs in σ, we have $j'_{n-k} = j_i$ for some index $j_i \in \sigma$. Obviously, we have $t_{pp+1}(j'_1, \ldots, j'_{n-k}) \geq t_{pp+1}(R_i(p))$, $p = 1, \ldots, m-1$. From Lemma 1 the completion time of the sequence $(sj'_1, \ldots, j'_{n-k})$ satisfies the following inequalities:

$$t(sj'_1, \ldots, j'_{n-k}) \geq t_{1p}(s) + t_{pp+1}(j'_1, \ldots, j'_{n-k}) + \sum_{r=p+2}^{m} a_{rj_i} \geq$$

$$t_{1p}(s) + t_{pp+1}(R_i(p)) + \sum_{r=p+2}^{m} a_{rj_i}, \qquad 1 \le p \le m-2,$$

$$t(sj'_1, \ldots, j'_{n-k}) \geq t_{1m-1}(s) + t_{m-1m}(R_{n-k}(m-1)),$$

$$t(sj'_1, \ldots, j'_{n-k}) \geq t_{1m}(s) + \sum_{j\in\sigma} a_{mj}.$$

We deduce that $t(sj'_1, \ldots, j'_{n-k}) \geq b_p$, $p = 1, \ldots, m$. We thus obtain a lower bound

$$B(s) = \max_{1\le p\le m} \{b_p\}.$$

For calculating the above lower bound, at the root node of the search tree we calculate $\Sigma_{r=p+2}^{m} a_{rj}$ and obtain the optimal job order by applying

Johnson's rule to a_{pj}, a_{p+1j} for $1 \le p \le m-1$ in $O(mnlogn)$ steps; for any subset of unscheduled jobs σ, the optimal order $R_{n-k}(p)$ has been determined at the root node, and we calculate the lower bound $B(s)$ in $O(m(n-k)^2)$ steps.

When we look for an optimal sequence of a flow-shop scheduling problem, we can consider both the original problem and the inverse flow-shop problem in which the processing times a_{kj} and a_{m-k+1j} are interchanged for all jobs j and all machines k. In that case nodes of the form $\{(s,\ldots,s')\}$ will occur in the search tree, where s and s' are two given mutually disjoint fixed partial schedules (either s or s' may be empty). In [14] Potts presents an "adaptive branching rule" for such nodes $\{(s,\ldots,s')\}$ and gives a lower bound $B(s,s')$. Computational results indicate that his algorithm is more efficient.

As an analogue to (26), we propose a lower bound on all possible completions of $(s,\ldots,s')$. Let $\sigma = N\backslash(s \cup s')$. Arranging all jobs belonging to σ in the sequence $(j_1,j_2,\ldots,j_h)$ by applying Johnson's rule to a_{pj}, a_{p+1j}, $\forall j \in \sigma$, we define

$$V_1(p) = (j_2,j_3,\ldots,j_h,j_1),$$

$$V_i(p) = (j_1,\ldots,j_{i-1},j_{i+1},\ldots,j_h,j_i),\quad i = 2,\ldots,h-1,$$

$$V_h(p) = (j_1,j_2,\ldots,j_{h-1},j_h),$$

and

$$b'_p = t_{1p}(s) + \min_{1\le i\le h} \{t_{pp+1}(V_i(p)) - a_{p+1j_i} + t_{p+1m}(j_i s')\},$$
$$p = 1,\ldots,m-2,$$

$$b'_{m-1} = t_{1m-1}(s) + t_{m-1m}(V_h(m-1)) + t_{mm}(s'),$$

$$b'_m = t_{1m}(s) + \sum_{j\in\sigma} a_{mj} + t_{mm}(s').$$

We thus obtain a lower bound

$$B'(s,s') = \max_{1\le p\le m} \{b'_p\}.$$

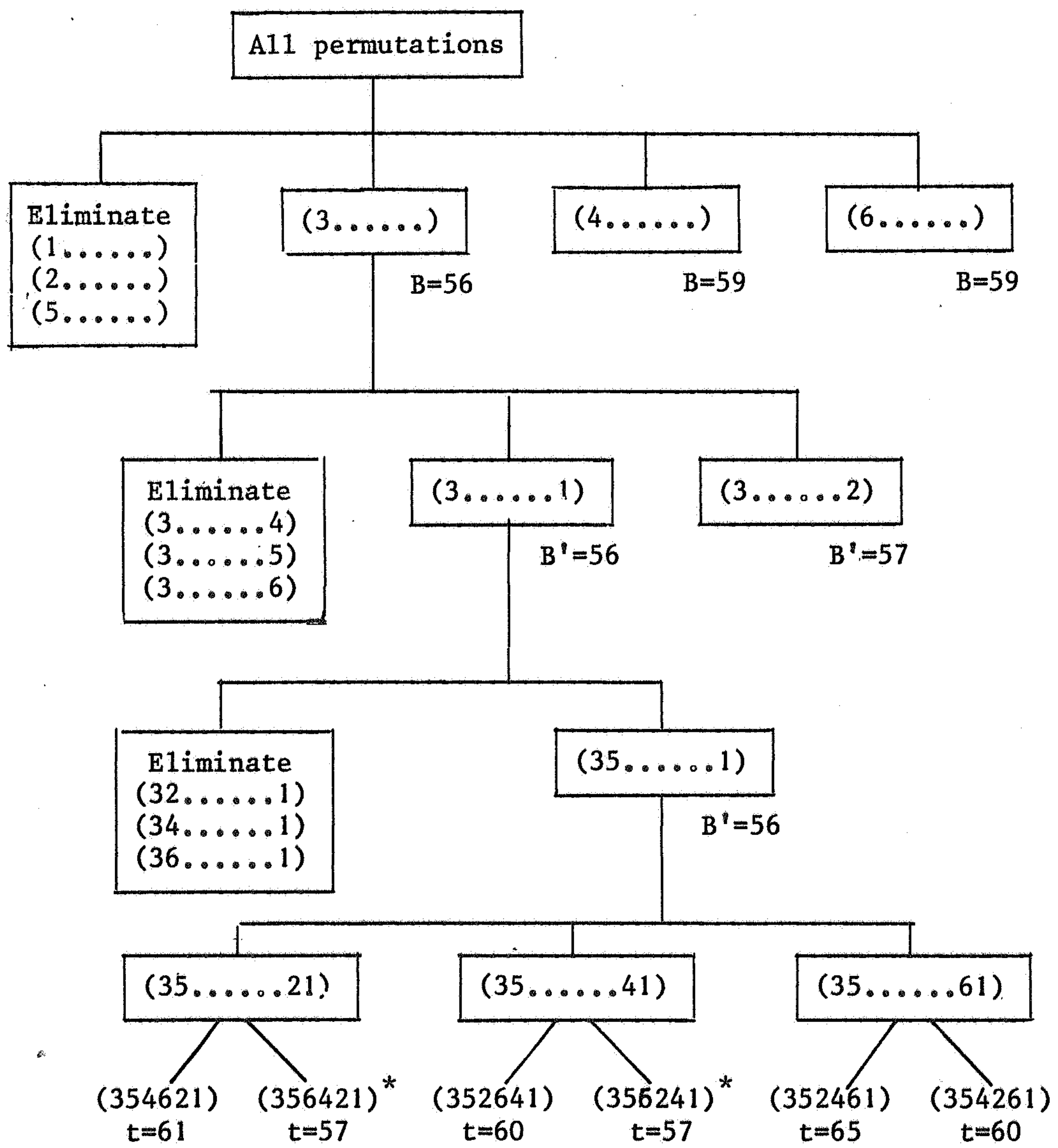
If s' is empty, we have $b'_p = b_p$, $p = 1,\ldots,m$, and $B'(s,\phi) = B(s)$.

Incorporating elimination criteria (5), (22), (23), (25) and lower bounds B(s), B'(s,s'), using the adaptive branching rule from [14] and a heuristic procedure to produce an initial upper bound (for example, the procedure in [5]), we obtain an algorithm for the flow-shop sequencing problem.

EXAMPLE (Lomnicki (1965)).

	J_1	J_2	J_3	J_4	J_5	J_6
M_1	6	12	4	3	6	2
M_2	7	2	6	11	8	14
M_3	3	3	8	7	10	12

Upper bound = 59.



Optimal sequences (356421), (356241). Minimum completion time 57.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge useful discussions with B.J. Lageweg and J.K. Lenstra.

REFERENCES

[1] ASHOUR, S., *A Branch-and-Bound Algorithm for Flow-Shop Scheduling Problems*, AIIE Trans. 2, 172-176 (1970).

[2] BAKER, K.R., *A Comparative Study of Flow-Shop Algorithms*, Oper. Res. 23, 62-73 (1975).

[3] BESTIWICK, P.F. & N.A.J. HASTINGS, *A New Bound for Machine Scheduling*, Oper. Res. Quart. 27, 479-487 (1976).

[4] BROWN, A.P.G. & Z.A. LOMNICKI, *Some applications of the Branch-and-Bound Algorithm to the Machine Scheduling Problem*, Oper. Res. Quart. 17, 173-186 (1966).

[5] CAMPBELL, H.G., R.A. DUDEK & M.L. SMITH, *A Heuristic Algorithm for the* n *Job*, m *Machine Sequencing Problem*, Management Sci. 16, B 630-637 (1970).

[6] DANNENBRING, D.G., *An Evaluation of Flow Shop Sequencing Heuristics*, Management Sci. 23, 1174-1182 (1977).

[7] IGNALL, E. & L. SCHRAGE, *Application of the Branch-and-Bound Technique to some Flow-Shop Scheduling Problems*, Oper. Res. 13, 400-412 (1965).

[8] JOHNSON, S.M., *Optimal Two-and Three-Stage Production Schedules with Setup Times Included*, Naval Res. Logist. Quart. 1, 61-68 (1954).

[9] LAGEWEG, B.J., J.K. LENSTRA & A.H.G. RINNOOY KAN, *A General Bounding Scheme for the Permutation Flow-Shop Problem*, Oper. Res. 26, 53-67 (1978).

[10] LENSTRA, J.K., *Sequencing by Enumerative Methods*, Mathematical Centre Tract 69, Mathematisch Centrum, Amsterdam, 1977.

[11] LOMNICKI, Z.A., *A Branch-and-Bound Algorithm for the Exact Solution of the Three-Machine Scheduling Problem*, Oper. Res. Quart. 16, 89-100 (1965).

[12] McMAHON, G.B., *Optimal Production Schedules for Flow Shops*, Can. Oper. Res. Soc. J. 7, 141-151 (1969).

[13] McMAHON, G.B. & P.G. BURTON, *Flow-Shop Scheduling with the Branch-and-Bound Method*, Oper. Res. 15, 473-481 (1967).

[14] POTTS, C.N., *An Adaptive Branching Rule for the Permutation Flow-Shop Problem*, European J. Oper. Res. 5, 19-25 (1980).

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

[16] SMITH, R.D. & R.A. DUDEK, *A General Algorithm for the Solution of the* n-*Job,* m-*Machine Sequencing Problem of the Flow Shop*, Oper. Res. 15, 71-82 (1967).

[17] SZWARC, W., *Elimination Methods in the* mxn *Sequencing Problem*, Naval Res. Logist. Quart. 18, 295-305 (1971).

[18] SZWARC, W., *Optimal Elimination Methods in the* mxn *Sequencing Problem*, Oper. Res. 21, 1250-1259 (1973).

[19] SZWARC, W., *Mathematical Aspects of the* 3xn *Jobshop Sequencing Problem*, Naval Res. Logist. Quart. 21, 145-153 (1974).

[20] YUE, M. & J. HAN, *On the* mxn *Flow-Shop Scheduling Problem*, Scientia Sinica, 5, 462-470 (1975).

[21] YUE, M., *On the* n *Job,* m *Machine Sequencing Problem of Flow-Shop*, in Operational Research 75 (K.B. Haley, ed.), North-Holland Publishing Company, 1976, 179-200.