SEQUENCING n JOBS ON TWO MACHINES WITH ARBITRARY TIME LAGS; ALTERNATE PROOF AND DISCUSSION OF GENERAL CASE

S. M. Johnson

P-1526

October 27, 1958

The RAND Corporation
1700 MAIN ST. • SANTA MONICA • CALIFORNIA

SUMMARY

This note presents an alternate proof of a result of L. G. Mitten, solving the problem of sequencing n jobs through two machines with arbitrary time lags when the job sequences are the same for both machines. The more difficult general problem is also discussed and partially solved.

SEQUENCING n JOBS ON TWO MACHINES WITH ARBITRARY TIME LAGS; ALTERNATE PROOF AND DISCUSSION OF GENERAL CASE

S. M. Johnson

Introduction

In [2], L. G. Mitten solved the problem of sequencing n jobs on two machines with arbitrary time lags while minimizing total elapsed time, when it is assumed that the job sequences on both machines are the same. He mentioned that the general problem would sometimes involve different job sequences on the two machines and hence would be quite difficult.

In the present note an alternate proof of Mitten's result is shown to follow as a corollary of a special three-stage problem solved in [1] when properly interpreted. Also Mitten's general case is discussed and partially solved.

An Alternate Derivation of Mitten's Result

In [2], L. G. Mitten solved the following problem. Each of n jobs must be run first on machine I and then on machine II. Running times (including any set-up or tear-down times) for the i-th job are A_i on I and B_i on II. Let $D_i \geq 0$ be the arbitrary time lag associated with item i, such that job i is not started on II sooner than D_i time units after it was started on I, nor finished on II sooner than D_i time units after it was finished on I. A rule is given for determining the sequence in which the jobs are to be run on the machines, using the same sequence for both machines, in order to minimize the time

between the start of the first job on machine I and the completion of the last job on machine II.

Mitten's solution to this problem is as follows. Sequence those jobs whose $A_i < B_i$ in order of increasing value of D_i, all such jobs coming before jobs whose $A_i \geq B_i$, which in turn are sequenced in order of decreasing value of D_i.

We proceed as follows.

Write D_i, the arbitrary time lag for the i–th job lot, as

$$D_i = M_i + \min(A_i, B_i) \tag{1}$$

where $M_i \geq -\min(A_i, B_i)$.

If $M_i \geq 0$, we can interpret M_i as the processing time of the i–th job lot on all intermediate stages or non–bottleneck machines.

If $M_i < 0$, then we can interpret this case (following Mitten) as one involving lap–phasing, starting an item on the next stage before its entire job lot is available.

Then Mitten's problem can be interpreted as a restricted three–stage problem, which was treated in [1].

The general three–stage time matrix discussed in [1], in terms of present notation, is

A_1	A_2				A_n
M_1	M_2				M_n
B_1	B_2				B_m

For this sequence $S_0 = (1, 2, \ldots, n)$ the total time to process all the jobs is

$$(2) \qquad T(S_0) = \max_{1 \leq u \leq v \leq n} \left(\sum_{i=1}^{u} A_i + \sum_{i=u}^{v} M_i + \sum_{i=v}^{n} B_i \right) .$$

We wish to permute the columns of the time matrix to find the minimum $T(S)$.

This three-stage problem was solved in [1] only for a special set of assumptions leading to the conclusion that $u = v$ in (2). In the present problem the same restriction $u = v$ holds since the total time for a given sequence, $S = (1, 2, 3, \ldots, n)$, is

$$(3) \qquad T(S) = \max_{1 \leq u \leq n} \left\{ \sum_{i=1}^{u} A_i + M_i + \sum_{i=u}^{n} B_i \right\} ,$$

that is, there is no bottleneck on the intermediate stage. Then the solution in [1] was shown to reduce to an equivalent two-stage problem with an optimal given by the transitive rule: Item i precedes item j if

$$(4) \qquad \min(A_i + M_i, B_j + M_j) < \min(A_j + M_j, B_i + M_i)$$

with ties ordered either way. In [1] it was shown that this led to an easy method of scheduling:

Find the smallest number of the set of $2n$ numbers $(A_i + M_i, B_i + M_i)$, $i = 1, 2, \ldots, n$. If it is an $A_i + M_i$, place that item i first; if it is a $B_i + M_i$, place that item last. Then repeat on the reduced set of items until all are

ordered.

Mitten's rule gives an alternate interpretation of (4).

If $A_i < B_i$, $B_j < A_j$, item i precedes item j from (4).

If $A_i < B_i$, $A_j < B_i$, then (4) implies item i precedes item j if $A_i + M_i < A_j + M_j$, that is, if $D_i < D_j$.

Similarly, if $B_i > A_i$, $B_j > A_j$, item i precedes item j if $D_j < D_i$.

The two rules are equivalent (except possibly for conventions concerning ties). In any case, both give the same total time.

Note that Mitten's rule also applies to the standard two-stage problem in [1] and is perhaps easier to remember than the rule given in [1].

The General Case Allowing Different Job Sequences

In the problem not treated by Mitten, where different job sequences are allowed for the two bottleneck machines, the following remarks will reduce the problem in most cases down to a relatively small list of sequences whose total times can be compared and the optimal sequence found.

One can easily show that for every sequence S_I on machine I there is an optimal sequence on machine II given by processing items in order of their availability times on II. For suppose the job sequence on II is not in the same order as the sequence $\{t_i\}$ where t_i is the availability time for item i on II for a given fixed sequence S_I on I. If items i and j of S_{II} are not in the same order as t_i and t_j, then they both

must have started on II at a time after $\max(t_i, t_j)$. Thus we can interchange consecutive items i and j on II without loss of time. By successive interchanges, starting from the left, of consecutive pairs of those items which are out of order we can reorder S_{II} to match up with the sequence $\{t_i\}$ without loss of time. Then start each item on machine II as soon as possible. This reduces the problem from $(n!)^2$ cases to n! cases.

But then use symmetrical arguments (reversing the time scale) to find an optimal sequence on I for a given sequence on II. Repeating this process we eventually find a pair of mutually optimal sequences (S_I, S_{II}). However, and this is the real difficulty, there may be many such pairs of sequences which satisfy this necessary condition for over-all optimality.

Nevertheless, the above technique leads to a proof of the following useful result.

Theorem.

A necessary condition for a reversal of order of consecutive items i, j on I to j, i on II in a pair of mutually optimal sequences (S_I, S_{II}) is that

$$M_i > M_j + \max(A_j, B_j). \tag{5}$$

This is also a sufficient condition provided item i is not reversed with its preceding item on I and item j is not reversed with its following item on II.

Since $t_i > t_j$, $M_i > A_j + M_j$. Symmetrically, $t_i' > t_j'$ or $M_i > B_j + M_j$, giving (5).

Now for simplicity assume there is only one M_i satisfying (5) for several items j, k, ℓ, say. The question of which item should be interchanged with item i seems to be too hard to answer by any simple decision rule. We propose to try each case and compare. Then if we try reversing i with j where

$$(5') \qquad M_j + A_j + B_j > M_i > M_j + \max(A_j, B_j),$$

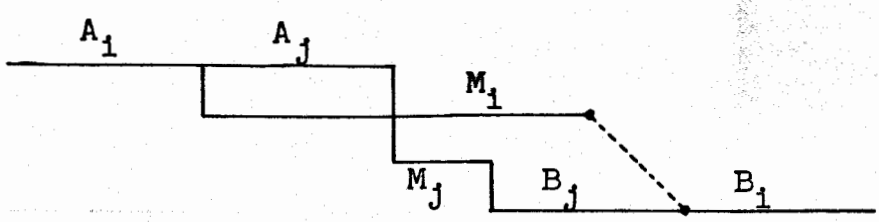
the two consecutive columns for items i and j in the time matrix

A_i	A_j
M_i	M_j
B_i	B_j

can be replaced by a single column

$A_i + A_j$
M_j
$B_j + B_i$

corresponding to single fictitious items, as far as computing total time is concerned. To see this consider the corresponding Gantt chart:



Here M_j is the analogous time lag between the finish of the job pair (i, j) on I and the start of the job pair (j, i) on II.

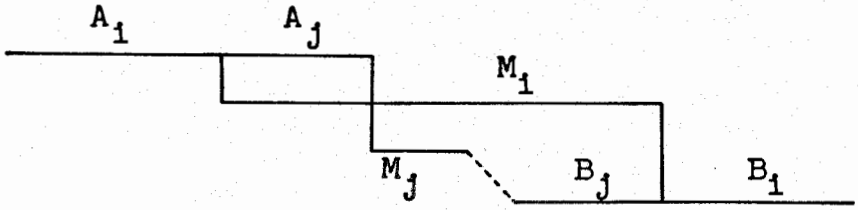
Similarly, if

(5") $$M_i > M_j + A_j + B_j,$$

the new column is

$A_i + A_j$
$M_i - A_j - B_j$
$B_j + B_i$

since from the Gantt chart below



the analogous delay time is $M_i - A_j - B_j$.

Thus we have replaced the reversed pair of items by a single fictitious item and now are left with a problem of the type solved by Mitten. All that is required is to insert this "item" into its proper place in the sequence given by Mitten's rule.

Next we try each possible reversal with item i, and compute the total time for such optimally ordered sequences and compare. Note it may still be best to have no reversal for item i even though (5) is satisfied for item i and some

item j since they may not be adjacent in the optimal sequencing.

In general, there may be several pairs of items satisfying (5) or even combinations of 3 or more items calling for permutations of order from S_I to S_{II}. If there are not too many possible cases, each of these can be worked out and compared.

Further analysis yields some dominance rules concerning which items should be interchanged but the results are too special to be of very much value.

REFERENCES

1. Johnson, S. M., "Optimal Two- and Three-stage Production Schedules with Setup Times Included," Naval Research Logistics Quarterly, Vol. 1, No. 1, March, 1954, pp. 61-68.

2. Mitten, L. G., "Sequencing n Jobs on Two Machines with Arbitrary Time Lags," Management Science, to be published.