COWLES FOUNDATION DISCUSSION PAPER NO. 73

Note: Cowles Foundation Discussion Papers are preliminary materials circulated privately to stimulate private discussion and critical comment. References in publications to Discussion Papers (other than mere acknowledgment by a writer that he has access to such unpublished material) should be cleared with the author to protect the tentative character of these papers.

On the Job Shop Scheduling Problem*

Alan S. Manne

May 8, 1959

* Research undertaken by the Cowles Commission for Research in Economics under Contract Nonr-358(01), NR 047-066 with the Office of Naval Research.

On the Job Shop Scheduling Problem*

Abstract

This is a proposal for the application of discrete linear programming to the typical job shop scheduling problem - one that involves both sequencing restrictions and also non-interference constraints for individual pieces of equipment. Thus far, no attempt has been made to establish the computational feasibility of the approach in the case of large-scale realistic problems. This formulation seems, however, to involve considerably fewer variables than two other recent proposals [1, 4], and on these grounds may be worth some computer experimentation.

Statement of the problem

It will be assumed that this sequencing problem involves the performance of n tasks - each task being defined in such a way as to require the services of a single machine for an integral number of time units. (For concreteness, we may refer to the unit of time as a "day".) The scheduling problem consists of drawing up a plan for time-phasing the individual jobs so as to satisfy: (a) sequencing requirements - e.g., the children must be washed before they are dried, and (b) equipment interference problems - e.g., the one-year-old and the three-year-old cannot occupy the bathtub at the same time. (All parents will devoutly hope that each of these tasks can be performed in less than a day.)

---

* This paper was stimulated by reading an unpublished manuscript by E. H. Bowman [1]. Bowman's ideas have led directly to the model formulated here.

The integer-valued unknowns x_j are to indicate the day on which task j is to be begum. $(x_j = 0, 1, \ldots T)^*$ Just as in Selmer Johnson's formulation

---

* For purposes that will shortly become evident, it will be convenient to suppose that we have sufficient a priori knowledge to be able to select a (large) integer T that will constitute a redundant upper bound upon the unknowns x_j .

---

[3], the schedule is to be drawn up so as to minimize the "make-span," i.e., the elapsed calendar time for the performance of all jobs - subject, of course, to the constraints upon sequencing and machine interference, and also subject to any delivery date requirements on individual items.

Non-interference restrictions

Suppose that jobs j and k require a_j and a_k consecutive days respectively. Then if they are to be prevented from occupying the same machine at the same time, we must required that one of the two must precede the other by sufficient time so that the first one can be completed before the second is begun:

$$\text{either (1.a)} \quad x_j - x_k \geq a_k$$
$$\text{or else (1.b)} \quad x_k - x_j \geq a_j \ .$$

In order to convert this condition into a linear inequality in integer unknowns, it will be convenient to define a new integer-valued variable y_{jk} , and to write down the following restrictions:

$$(2) \qquad 0 \leq y_{jk} \leq 1$$

$$(3) \qquad (T + a_k)y_{jk} + (x_j - x_k) \geq a_k$$

$$(4) \qquad (T + a_j)(1-y_{jk}) + (x_k - x_j) \geq a_j$$

Condition (2) ensures that y_{jk} equals either zero or else unity. We already know that $|x_j - x_k| \leq T$. The effect of conditions (3) and (4) may therefore be summarized as follows:

$$\text{If } (x_j - x_k) \begin{Bmatrix} \geq 1 \\ = 0 \\ \leq -1 \end{Bmatrix}, \text{ condition (3) implies}$$

$$\text{that } y_{jk} = \begin{Bmatrix} 0,1 \\ 1 \\ 1 \end{Bmatrix}, \text{ and by (4), } y_{jk} = \begin{Bmatrix} 0 \\ 0 \\ 0,1 \end{Bmatrix}.$$

Hence, if $(x_j - x_k) = 0$, there is no value that can be assigned to y_{jk} so as to satisfy both (3) and (4). If, on the other hand, $(x_j - x_k) \neq 0$, y_{jk} will be set at a value of either zero or unity, depending upon which job is to precede the other. Equations (3) and (4) then ensure that the one job will be initiated in sufficient time to be completed before the beginning of the second one. Note that with the classical form of linear programming, it would have been impossible to specify such an either-or condition as (1). This non-interference restriction leads directly to a non-convex set of restraints upon the unknowns. It is little wonder that Gomory's discovery of integer programming [2] has led to a revival of interest in the machine interference problem.

Sequencing restrictions

Once the non-interference stipulations have been written down, the remainder of the formulation becomes virtually automatic. If job j is to precede k, this means that job k is to be performed at least a_j days later than j. The integer programming condition becomes:

$$(5.a) \qquad x_j + a_j \leq x_k$$

"Weak" precedence relationships may be written in an analogous fashion. For example, in order to specify that both jobs i and j precede k, but that there is no precedence restriction affecting the performance of i and j, we would have:

$$(5.b) \qquad x_i + a_i \leq x_k$$

$$x_j + a_j \leq x_k$$

Still another possibility might be that there be a delay of exactly Θ_{jk} days between the performance of jobs j and k. Such a restriction would be indicated by:

$$(5.c) \qquad x_j + a_j + \Theta_{jk} = x_k$$

Specific delivery requirements

It may happen that the shop is committed to the delivery of an individual job no later than a specified date. If task j is the last task which the shop is to perform upon the item, and if the item is to be available on day d_j, this form of requirement may be written:

$$(6) \qquad x_j + a_j \leq d_j$$

Overall delivery requirements

Following S. Johnson [3], we shall employ as our minimand the "make-span" or total calendar time needed for the performance of all prospective jobs. If this calendar time is denoted by $\underline{t}$, the problem now consists of the minimization of $\underline{t}$ with respect to the non-negative integers x_j and y_{jk}, subject to constraints (2)-(6), and also subject to:

$$(7) \qquad x_j + a_j \leq t \qquad (j = 1, \ldots n)$$

For the economist who is thoroughly conditioned to look askance at any other minimand than dollar costs, it is difficult to become reconciled to the adoption of Johnson's criterion, the minimization of $\underline{t}$, the make-span. In defending this choice of minimand, however, it should be pointed out that $\underline{t}$ is significantly correlated with dollar costs. In minimizing $\underline{t}$ we also obtain the following cost and profit benefits: (a) a lowered amount of inventory tied up in work-in-process, (b) a shorter average customer delay time for all items, and (c) a lowered amount of idle time incurred prior to the performance of all currently booked jobs - i.e., a greater capacity to take on additional work in the event that new orders materialize. Since all of these factors work in the same direction, calendar time appears to be a reasonable proxy variable for economic cost. The job sequence that serves to minimize the make-span ought to be one that also scores quite well on the criterion of dollar costs.

Computational aspects

Not counting any of the slack variables nor the minimand $\underline{t}$, the number of unknowns here is equal to the total number of the x_j plus the y_{jk}. If,

then, there are $\underline{n}$ tasks and if also there are $\underline{m}$ possible conflicting pairs of machine assignments, the total number of unknowns would come to $n + m$. For example, with 5 machines and with 10 tasks to be performed on each machine, we would have $n = 50$, and $m = \frac{(5)(10)(10-1)}{2} = 225$. The total number of integer-valued unknowns x_j and y_{jk} would come, therefore to 275 - an impressive computational load but by no means an impossible one.

One of the most promising avenues to be explored here would be a reduction in the number of the unknowns y_{jk}. These unknowns are involved only in connection with the machine interference equations (2)-(4), and amny of these restrictions must turn out to be redundant in any real-world problem. Whenever the shop is one in which one machine typically precedes another in the processing sequence, then the precedence restrictions (5) together with the non-interference restrictions connected with the most heavily loaded machine will automatically tend to make the non-interference restrictions redundant in the case of the less heavily loaded one. This conjecture could easily be tested by experimentation on small-scale problems.

References

1. Bowman, E. H., "The Schedule Sequencing Problem," March 19, 1959, unpublished manuscript.

2. Gomory, R., "Outline of an Algorithm for Integer Solutions to Linear Programs," Bulletin of the American Mathematical Society, September 1958.

3. Johnson, S. M., "Optimal Two and Three Stage Production Schedules and Setup Times Included," Naval Research Logistics Quarterly, 1954.

4. Wagner, H., "An Integer Linear Programming Model for Machine Scheduling," Technical Report No. 66, Contract N6onr-25133, Stanford University, December 24, 1958.