STICHTING
MATHEMATISCH CENTRUM
2e BOERHAAVESTRAAT 49
AMSTERDAM

AFDELING MATHEMATISCHE STATISTIEK

Solution of certain assignment-scheduling problems with discrete linear programming

by

Jac.M. Anthonisse

February 1967

Summary

n jobs must be assigned to M machines and the jobs assigned to the same machine must be scheduled, in such a way that the total cost of performing all jobs is minimized.
The method proposed here is applicable if the machines are independent. The solution of the scheduling-assignment problem is found by solving a discrete linear programming problem with p+q constraints, where p and q denote the number of different machines and jobs respectively. The number of variables depends on the number of jobs that might be run subsequently on a machine.

§1.

Consider the following assignment-scheduling problem:
M machines, labeled $m_1, \dots, m_M$, are available. Machine m_i may be used for t_{m_i} time units, against cost c_{m_i} per time unit.
Each of n jobs, labeled $j_1, \dots, j_n$, must be performed on one of the machines. Job j_k takes $u_{j_k m_i}$ time units on machine m_i. If, on machine m_i, job j_k is to be followed by job j_l it takes $v_{j_k j_l m_i}$ time units to adjust the machine.
Before starting a first job or after completing a last job on a machine no adjustment is necessary.
Assign the jobs to machines and schedule the jobs assigned to same machine, in such a way that the total cost of performing all jobs is minimized.

§2.

Assignment-scheduling problems of the type given above may be formulated as a discrete linear programming problem, but often the number of variables and the number of constraints are of a magnitude preventing actual solution.

This is mainly caused by the fact that a great number of scheduling problems must be solved implicitly.

The results of solving the assignment-scheduling problem is that to each machine at most 1 ordered subset from the set of jobs is assigned, where the following conditions are satisfied:

1. the union of the assigned subsets is the complete set of jobs,
2. subsets assigned to different machines are disjunct,
3. the ordered subset assigned to machine m_i can be completed in at most t_{m_i} time units.

From these conditions it is clear that the assignment-scheduling problem can be solved in the following 2 steps:

step 1. generate, for each machine, the ordered subsets from the set of jobs, which may be completed within the available time,

step 2. assign, under conditions 1. and 2., at most 1 ordered subset to each machine, minimizing the total cost.

§3.

Now a zero-one linear programming problem will be given that is equivalent with step 2.

Define a JOB as an ordered subset of jobs that may be run, in the given order, on a machine. The JOB's associated with machine m_i (by step 1) may be labeled $J_{N_{i-1}+1}, \ldots, J_{N_i}$, where $N_0 = 0$.

Define further:

$N = N_M$, C_j = the cost of performing the JOB labeled J_j ($j = 1, \ldots, N$), the $M \times N$ matrix A with elements

$$a_{ij} = \begin{cases} 1 \text{ if } N_{i-1}+1 \leq j \leq N_i \\ 0 \text{ otherwise,} \end{cases}$$

the $n \times N$ matrix B with elements

$$b_{kj} = \begin{cases} 1 \text{ if job } j_k \text{ is an element of JOB } J_j \\ 0 \text{ otherwise.} \end{cases}$$

The linear program

$$\text{minimize} \quad \sum_{j=1}^{N} C_j x_j$$

under the conditions

$$\sum_{j=1}^{N} a_{ij} x_j \leq 1 \qquad (1 = 1, \ldots, M) \qquad (1)$$

$$\sum_{j=1}^{N} b_{kj} x_j = 1 \qquad (k = 1, \ldots, n) \qquad (2)$$

$$x_j = 0 \text{ or } 1 \qquad (j = 1, \ldots, N) \qquad (3)$$

is equivalent with step 2.

The interpretation of $x_j = 0$ is : do not perform JOB J_j, $x_j = 1$ must be interpreted as: perform JOB J_j or: assign this ordered subset of jobs to the associated machine.

By the definition of A restrictions (1) and (3) have the effect that at most 1 JOB is assigned to each machine. Restrictions (2) and (3) ensure that exactly 1 JOB containing job j_k will be performed.

§4.

Although the method is presented with the help of a simple example is it clear that the method is applicable to any assignment-scheduling problem in which the machines are independent.

Of special interest are the problems with a stochastical character, in step 1 all JOB's can be excluded which would, with probability $\geq \alpha$, take more than t_{m_i} the time units on machine m_i.

§5.

If identical jobs and/or identical machines are present the assignment-scheduling problem is easily formulated as a general integer linear programming problem.

Suppose $n = \sum f_k$ and $M = \sum F_i$, where f_k and F_i denote the number of jobs of type j_k and the number of machines of type m_i respectively.

In this case it is sufficient to generate the JOB's for each type of machine, and step 2 is equivalent with the following problem:

minimize $\sum_{j=1}^{N} C_j x_j$

under the restrictions

$$\sum_{j=1}^{N} a_{ij} x_j \leq F_i \qquad (i = 1, \ldots, p)$$

$$\sum_{j=1}^{N} b'_{kj} x_j = f_k \qquad (k = 1, \ldots, q)$$

x_j = a non-negative integer $(j = 1, \ldots, N)$

where p and q are the number of different machines and jobs respective-y, b'_{kj} = the number of times a job of type j_k occurs in JOB J_j, and the other symbols have been defined above.

§6.

In the example given above, as well as in other applications, the generating of all JOB's is simplified by the fact that it is impossible to obtain a JOB by adding a job to an ordered subset of jobs that itself is not a JOB.

This leads to the following procedure to generate all JOB's for a particular machine.

1. determine the JOB's containing only 1 job, let V_1 denote this set of JOB's, i:=1,
2. determine the ordered subsets containing i + 1 jobs, such that the first i jobs are a JOB,
3. select the ordered subsets containg i + 1 jobs that are JOB's, denote this set of JOB's by V_{i+1},
4. if V_{i+1} is empty all JOB's have been generated, otherwise i:=i+1 and proceed at 2.