MATHEMATICA VOL. 13 (36), 2, 1971, pp. 251–261

ON THE SCHEDULING PROBLEM IN THE CASE OF SEVERAL MACHINES OF THE SAME TYPE

by

L. NÉMETI

Cluj

1. Succesfull results have been obtained in the machine-job scheduling problem only after having taken into account some methods such as: the Branch and Bound technique [3], [4], S.E.P. [1], F. RADÓ 's algorithm for programming with logical conditions [7], [8], all these methods being closely connected.

In [5], [10] this problem was discussed in its following more simple aspect: n jobs are to be performed on m different machines[1]. In two papers [2], [6], a more general case, that of several machines of the same type, was studied, and the authors gave two solving methods.

In this paper we present a third method, which seems to have certain advantages compared with the already known techniques. Paragraph 1 contains the basic notions and definitions, the statement of the problem and the proposed mathematical model. Paragraph 2 indicates the particular form of the Branch and Bound method to be used for the solving of the model. In paragraph 3 there are considered the fundamental problems which appear in the application of the Branch and Bound method and a suitable solving procedure. At last, paragraph 4 contains considerations regarding the proof of the correctness of the proposed method and indications regarding the recommanded strategy of choice.

The jobs to be performed are numbered from 1 to n. These numbers (codes) forme the *set of jobs* $N = \{1, 2, \ldots, n\}$. The types of the machines are also numbered, these numbers forming the *set of types* $M =$

[1] This formulation is more general, than the one — which is widely used — of a technological line in which is assumed:
a) the pieces to be processed pass in the same order through the machines of the line;
b) any machine processes the pieces in the same order (overtaking forbidden).

$= \{1, 2, \ldots, m\}$. For every type $s \in M$ we have given the number μ_s of machines of this type, existing in the workshop. The set N admits the partition

$$N = \bigcup_{s \in M} N_s$$

the subset N_s containing (the codes of) the jobs, which are to be performed on a machine of the type s, according to the technological process, which has been adopted. The work time $t_i > 0$ is given for every job $i \in N$. The technological process also stipulates a sequencing relation between certain pairs of job. The set of these pairs is denoted by $H \subset N \times N$, and $(i, j) \in H$ means, that the job j may earliest begin after a given delay $t_{ij} > 0$, considered from the beginning of the job i (that is, j is an direct technological succesor of i). It is also assumed that: a) a machine performs, at a given moment, at most one job; b) a job is not performed simultaneously on several machines; c) a job, which has already begun, is completely carried out without interruption.

The problem is to find the starting moment x_i of each job $i \in N$ so, that a given economic function be optimal (minimal).

Now let us assume, that the vector $X = (x_i | i \in N)$ has already been determined. Denote by

$$I_i = [x_i, x_i + t_i)$$

the *performing interval* of the job $i \in N$. Let us take a subset $L \subset N_s$, $s \in M$ being a given type. On the x-axis we define the *loading function* with respect to L: $\gamma(x|L)$, the value of which in the point x is equal to the number of the performing intervals I_i, $i \in L$, for which $x \in I_i$ takes place.

These definitions and notations allow us to formulate the mathematical model of our problem in the following way:

Determine the vector $X = (x_i | i \in N)$, which verifies

the *starting conditions*

$$x_i \geqq 0, \quad i \in N \tag{1.1}$$

the *sequencing conditions*

$$x_j - x_i \geqq t_{ij}, \quad (i, j) \in H \tag{2.1}$$

and the *loading conditions*

$$\gamma(x|N_s) \leqq \mu_s, \quad s \in M \tag{3.1}$$

so, that the given economic function

$$z = f(X) \text{ be minimal.} \tag{4.1}$$

We assume in the sequel, that the relations

$$X'' \geqq X' (\equiv x_i'' \geqq x_i', \ \forall i \in N) \Rightarrow f(X'') \geqq f(X') \tag{5.1}$$

and

$$\lim_{x_i \to \infty} z = \infty, \quad i \in N \tag{6.1}$$

take place.

The vector X, which verifies the system $\mathfrak{S}$ of the contraints (1.1), (2.1) and (3.1), is called a *feasible program*. The solution of the problem $\mathfrak{L}$, which consists of the system $\mathfrak{S}$ together with the optimization condition (4.1) is said to be an *optimal program* $\mathfrak{X}$.

2. In order to solv the problem $\mathfrak{L}$ we will use in this paper the Branch and Bound method already mentioned in paragraph 1. It consists in solving a series of programming problems P (F. RADÓ calls them *fundamental problems*). In our case a fundamental problem consists of a system S of inequalities, which is to be verified by the vector X, together with an optimisation condition (4.1) (optimal solution). In order to describe the Branch and Bound method, is useful to introduce the *solving-tree* $F = (U, V)$, where to every node $\alpha \in U$ is associated a fundamental problem P^α, an optimal solution X^α of this and the corresponding value $m^\alpha = f(X^\alpha)$ of the economic function. To each successor β of a node $\alpha \in U$ (i.e. $(\alpha, \beta) \in V$) an other fundamental problem P^β is attached, generated from P^α by well defined rules.

These *generating rules*, as well as the problem P° associated to the root of the tree F being given — we will formulate them immediately — let us consider the terminal nodes of the solving-tree F in an intermediate stage. We choose such a node δ, with the smallest number m^δ. If the solution X^δ verifies the system $\mathfrak{S}$ of the previous paragraph, then it constitutes at the same time an optimal program with respect to the problem $\mathfrak{L}$ and the process is accomplished. Otherwise, new fundamental problems will be generated from P^δ (i.e. new nodes in F, succesors of δ) according to the established generating rules. If $\mathfrak{L}$ is compatible and the generating rules verify certain conditions, called *conditions sof legilimity*, then the process ends in a finite number of steps, and an optimal solution of $\mathfrak{L}$ is obtained.

The fundamental problem P° consists of the relations (1.1), (2.1) and (4.1).

The generation of the new fundamental problems $P_1, P_2, \ldots$, from a given one P is carried out in the following way:

Let the vector $\Xi = (\xi_i | i \in N)$ be the universal minimal (optimal) solution of the problem P. Using this solution, we will determine the functions $\gamma(x|N_s)$, $s \in M$, which we have been defined in the previous para-

graph. If all these functions verify the loading conditions (3.1), then the vector Ξ is a solution of the system $\mathcal{S}$ (i.e. it is a feasible program), and the node of F which belongs to Ξ is called a final node.

In the opposite case we will have types $s \in M$ and subsets $L \subset N_s$ of jobs, each of them with an afferent interval $J = [a, b)$, having the property that for $x \in J$

$$\nu = \gamma(x|L) - \mu_s > 0 \tag{1.2}$$

takes place.

Such a set L is called an *exceeding set* (i.e. exceeding the loading limit), the interval J is the afferent *exceeding* interval and the natural number ν is the corresponding *exceeding degree.*

If this set L contains $\mu_s + \nu$ elements and we have

$$J = \bigcap_{i \in L} I_i \neq \emptyset, \tag{2.2}$$

it is a *normal exceeding set.* The performing intervals, which correspond to any two elements of a normal exceeding set, are never disjoint.

From the normal exceeding sets we will choose one. Let this set be

$$L = \{i_1, i_2, \ldots, i_q\} \subset N_s \tag{3.2}$$

and assume, that the notation of the indices is made in such a way, that for the terminal moment of the jobs $i \in N$

$$\eta_i = \xi_i + t_i,$$

$$\begin{cases} \eta_{i_1} \leqq \eta_{i_2}, & i \in L \\ \eta_{i_2} \leqq \eta_{i_3}, & i \in L \setminus \{i_1\} \end{cases} \tag{4.2}$$

is verified.

Evidently, we have then for the coresponding exceeding interval $J = [a, b)$

$$b = \eta_{i_1}. \tag{5.2}$$

In order to generate new fundamental problems from the problem P, the disjunction

$$(x_{i_1} \geqq \eta_{i_1}) \vee (x_{i_2} \geqq \eta_{i_2}) \vee (x_{i_3} \geqq \eta_{i_3}) \vee \ldots \vee (x_{i_q} \geqq \eta_{i_q}) \tag{6.2}$$

will be formed.

Then a new fundamental problem generated from P consists of the conditions

$$x_i \geqq \xi_{i_1}, \quad i \in N \tag{7.2}$$

$$x_\beta \geqq \xi_\alpha + t_\alpha \tag{8.2}$$

$$x_j - x_i \geqq t_{ij}, \quad (i, j) \in H \tag{9.2}$$

$$z = f(X) : \text{minimal} \tag{10.2}$$

where the relation (8.2) is one of the terms of the disjunction (6.2).

In this way, making use of these terms succesively, we obtain q new fundamental problems $P_1, P_2, \ldots, P_q$ from the problem P. It is characteristic for their optimal solutions $\Xi_1, \Xi_2, \ldots, \Xi_q$ either that the exceeding degree of any of them, with respect to L, is smaller than this exceeding degree of the optimal solution Ξ of the problem P, or that L is no more an exceeding set of the corresponding solution, i.e. that in this case the condition (3.1) is partially verified in the sense, that

$$\gamma(x|L) \leqq \mu_s \tag{11.2}$$

takes place.

If we develop further the solving-tree according to the Branch and Bound method (or to the S.E.P. or to Radó's algorithm), the exceeding cases will be gradually reduced and, afterwards, eliminated, until we come upon solutions, which verify all the conditions (3.1) too. In other words we obtain feasible programs, on which the algorithm decides wether they are optimal or not.

3. Before showing, that the previous generating rules are legitimate, the structure of a fundamental problem and the appropriate solving methods are to be studied. As it results from the foregoing paragraph, such a problem P will have the following form:

$$x_i \geqq a_i, \quad i \in N \tag{1.3}$$

$$x_j - x_i \geqq t_{ij}, \quad (i, j) \in H \tag{2.3}$$

$$z = f(X) : \text{minimal}, \tag{3.3}$$

with given numbers a_i and t_{ij}.

Two any fundamental problems of a given problem $\mathcal{P}$ differ from one another only by the numbers a_i; the sets N and H and the numbers t_{ij} are the same.

Consequently, P represents a problem of potentials [9] and — because of the property (5.1) — it is its universal minimal solution [9], which must be found. In our case this is easily done. A network $G = (N, H)$, is attached to the problem P, where the set of nodes is N and the set of arcs : H. Assign

to every arc $(i, j) \in H$ the lenght t_{ij}. This network is supplemented with a node $i = 0$ and with the arcs $(0, i)$, $i \in N$ of lenght a_i. The new network will be $\bar{G} = (\bar{N}, \bar{H})$ with $\bar{N} = N \cup \{0\}$ and $\bar{H} = H \cup \{(0, i) | i \in N\}$. If the problem $\mathfrak{P}$ has been put correctly, this network has no cycles. Indeed, let us assume, that there would be a cycle $\mathcal{C} = (i_1, i_2, \ldots, i_{p_1}, i_1)$ in $\bar{G}$. By applying the relation (2.3) to $\mathcal{C}$, we obtain the contradiction

$$x_{i_1} - x_{i_2} + x_{i_2} - x_{i_3} + \ldots + x_{i_p} - x_{i_1} = 0 \geqq t_{i_1 i_2} + t_{i_2 i_3} + \ldots + t_{i_p i_1} > 0.$$

Hence, P is incompatible in this case. But, since the topological structure of all networks $\bar{G}$ is the same, then the problem P° (and also $\mathfrak{P}$) is incompatible too.

In a network without cycles, the nodes can be disposed according to levels or ranks [9]. The node $i = 0$, which is the root of $\bar{G}$, will have the level 0, and if a node have the level r, its predecessors have inferior levels and at least one of them will have the level $r - 1$. The determination of the components ξ of the universal minimal solution Ξ of P is done in the increasing order of levels. Thus, if we have determined the value of ξ_i, the numbers ξ_j belonging to the predecessors $j \in P(i)$ of i are all already determined. Here, the set of predecessors of the node i in $\bar{G}$ is denoted by $P(i)$:

$$P(i) = \{(j, i) | (j, i) \in \bar{H}\}, \; i \in N \tag{4.3}$$

We obtain the universal minimal solution Ξ of P by putting

$$\xi_0 = 0 \tag{5.3}$$

and

$$\xi_i = \max \left[a_i, \max_{j \in P(i)} (\xi_j + t_{ji})\right]$$

or shorter

$$\xi_i = \max_{j \in P(i)} (\xi_j + t_{ji}, a_i), \; i \in N. \tag{6.3}$$

A remarkable advantage of our procedure is, that the networks G are the same for all fundamental problems of the problem $\mathfrak{P}$. These fundamental problems differ from one another only by the values a_i.

We also notice, that if the problem P° is compatible, then $\mathfrak{P}$ also has a solution and viceversa. In this case all fundamental problems, which appear along the algorithm, are compatible.

4. In order to obtain correct results by the Branch and Bound method, the rules used for generating new fundamental problems have to satisfy certain conditions of legitimity (formulated e.g. for S.E.P. in [1] and [2]). We will consider two of these conditions, since their verification by the suggested generating rules is not abvious. We will prove the following propositions.

Proposition 1. *Let P be a fundamental problem, which has the vector Ξ as universal minimal solution. Assume, that Ξ possesses an exceeding set L with the exceeding degree $\nu > 0$. Let X be another solution of P, in which the set L has an exceeding degree $\nu' < \nu$, or in which L is no more an exceeding set ($\nu' \leqslant 0$), the relation (11.2) being verified. Then the vector X verifies the disjunction (6.2).*

Proof. Let ξ_i, $i \in N$ be the component of Ξ and x_i, $i \in N$ the components of X. From the definition of the universal minimal solution results

$$\xi_i \leqq x_i, \; i \in N. \tag{1.4}$$

There are at least one pair of jobs α, $\beta \subset L$ for which

$$x_\alpha \geqq x_\beta + t_\beta$$

takes place, otherwise the set L would constitute for X also an exceeding set with $\nu' \geqq \nu$. Hence, according to (1.4) we have

$$x_\alpha \geqq \xi_\beta + t_\beta. \tag{2.4}$$

If $\alpha = i_1$ (and therefore $\beta \neq i_1$)
then (2.4) means

$$x_\alpha = x_{i_1} \geqq x_\beta + t_\beta.$$

We obtain from (4,2)

$$x_\beta + t_\beta \geqq \eta_\beta \geqq \eta_{i_1}$$

Therefore

$$x_\alpha \geqq \eta_{i_1} \text{ (the first term in (6.2)).}$$

If $\alpha \neq i_1$,
then (2.4) can be written

$$x_\alpha \geqq \eta_\beta$$

and by (4.2) we have

$$\eta_\beta \geqq \eta_{i_1}, \text{ hence}$$

$x_\alpha \geqq \eta_{i_1}$ (one of the terms of (6.2), excepting the first), The disjunction (6.2) is verified.

The proposition 1 ascertains, that by branching in the solving-tree (by generating new fundamental problems) the optimal solution $\mathfrak{x}$ of problem $\mathfrak{X}$ cannot be omitted.

Proposition 2. *Let us consider in a solving-tree a path from the root to any given node containing r arcs. Denote by $P^{(r)}$ the fundamental problem attached to this node and by $\Xi^{(r)} = (\xi_i^{(r)} | i \in N)$ its universal minimal solution.*

Put

$$q^{(r)} = \sum_{i \in N} \xi_i^{(r)}.$$

Then

$$\lim_{r \to \infty} q^{(r)} = \infty \tag{*}$$

takes place.

This statement includes the case, when the considered path cannot contain arcs in a number grater than a certain given number R, because it ends sooner with a final node of the solving tree.

Since the number of jobs is finite, the relation (*) supposes, that the „moving" of a job (this notion will be explained farther down) during the branching — according to disjunction (6.2) — can be indefinitely repeated. This situation can be met actually, as it results from the following example (see figure). Three jobs are considered: 1, 2, 3, which are performed on the same machine. There is no technological sequencing connection between

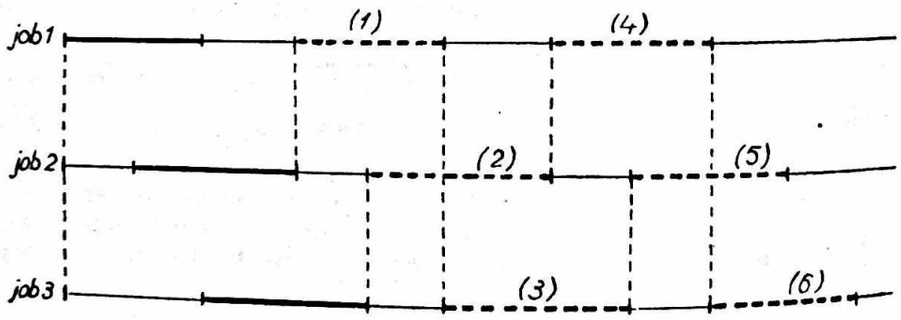


Fig. 1.

the jobs. On the figure, a scheduled job is reprezented by a segment (of the lenght of the job) on a straight line, which is the „prop" of the corresponding job (Gantt diagram). Let us assume, that at a stage of calculations the three jobs have been scheduled as it is shown by the three segments drawn with thick, continous lines.

If we repeatedly apply the generation rules using permanently the first term of disjunction (6.2), we obtain the situation shown by the figure. The numbers in brackets indicate the number of order to the repetition of the disjunction. We can see, that at the third, the sixth repetition, we came periodically upon the initial situation, only the segments are „moved" rightward.

After these remarks, we pass to the

proof. We notice first, taking into account the generating rules, that at the universal minimal solution of a fundamental problem, for the beginning ξ_i of any job i

either $\xi_i = 0$

or $\xi_i = \xi_j + t_j$

or $\xi_i = \xi_j + t_{ji}$

take place, where j is another job. Hence we have

$$\xi_i^{(r)} = \sum_{j \in N} k_j^{(r)}(i) t_j + \sum_{(j,h) \in H} k_{jh}^{(r)}(i) t_{jh}, \quad i \in N,$$

where $k_j^{(r)}(i)$ and $k_{jh}^{(r)}(i)$ are nonnegative integers. We obtain

$$q^{(r)} = \sum_{j \in N} k_j^{(r)} t_j + \sum_{(j,h) \in H} k_{jh}^{(r)} t_{jh}, \tag{**}$$

where $k_j^{(r)}$ and $k_{jh}^{(r)}$ are also nonnegative integers out of which one at least is different from zero.

Another remark is the following: let $P^{(r+1)}$ be a fundamental problem generated from $P^{(r)}$. Then

$$q^{(r+1)} > q^{(r)}. \tag{***}$$

Now, let us assume, that the numbers $k_j^{(r)}$ and $k_{jh}^{(r)}$ would admit a common superior bound χ, independent of r. Then, evidently, at most R different numbers $q^{(r)}$, $r = 1, 2, \ldots$ could exist, where R depends on χ and s (s: the number of elements from $N \cup H$). But if $r \geqq R$, then it results from (***), that at least one number $k_j^{(r)}$ or $k_{jh}^{(r)}$ bigger than χ must exist. Hence, these numbers are not bounded, if r can increase to the infinite. Then the relation (*) results from this fact, if we take into account the formula (**).

It results from proposition 2, taking into account the hypothesis (6.1), that the number $m^{(r)}$ attached to a fundamental problem $P^{(r)}$ (we recall that $m^{(r)} = f(\Xi^{(r)})$ is the value of the economic function for the optimal solution of the fundamental problem $P^{(r)}$) is increasing beyond any limit

with the lenght r of the path from $P^{(0)}$ to $P^{(r)}$. So, if problem $\mathfrak{P}$ has a solution, this will be obtained by our algorithm in a finite number of steps, because the branches of the solving-tree, where the value $m^{(r)}$ increase over $min\ f(X) = f(\mathfrak{X})$, will be left out by the algorithm.

The above proves the legitimity of our generating rules.

As at regards the choice of the exceeding set L out of the existing ones, this is arbitrary. In order to diminish the amount of calculations, which are needed, it is appropiate to use the following rule of selection.

Let us denote by $\mathfrak{L} = \{L\}$ the class of the normal exceeding sets L, which belong to the universal minimal solution Ξ of the considered fundamental problem P. An exceeding interval $J = [a, b)$ (2.3) and an exceeding degree ν (1.2) belong to each set $L \in \mathfrak{L}$. The smallest number b will be fixed. The exceeding sets, to which this number corresponds, form the subclass $\bar{\mathfrak{L}} \subset \mathfrak{L}$. This class $\bar{\mathfrak{L}}$ admits a partition

$$\bar{\mathfrak{L}} = \bigcup_{s \in M} \bar{\mathfrak{L}}_s$$

with the property

$$L \in \bar{\mathfrak{L}}_s \Leftrightarrow L \subset N_s.$$

We will choose a type $s \in M$, for which the subclass $\bar{\mathfrak{L}}_s$ is not void. Let $\bar{L}$ be the biggest set among the elements of $\bar{\mathfrak{L}}_s$. Then

$$L \in \bar{\mathfrak{L}}_s \Rightarrow L \subset \bar{L} (\subset N_s).$$

The elements of $\bar{L} = \{i_1, i_2, \ldots\}$ will be numbered according to the relation (4.2).

For branching, we will choose a subset $L \subset \bar{L}$, which contains at least the jobs i_1 and i_2, and which has an exceeding degree $\nu = 1$, i.e. L will consists of $q = \mu_s + 1$ elements.

The set L can be obtained with the following procedure:

For each type $s \in M$, there has to be made the following calculations with respect to the operations $i \in N_s$ (we assume that the considered fundamental problem has been solved, and therefore the starting moments x_i of the operations $i \in N$ are known).

1. The operations which have to be performed on the type s will be ordered according to the non-decreasing end-moments $y_i = x_i + t_i$. The ordered set N_s is denoted by $\bar{N}_s (\bar{N}_s = (i_s^1, i_s^2, \ldots))$.

2. We denote by $R_s^p \subset N_s$ the subset, which contains only the elements i_s^r, $r \geqq p$ of $\bar{N}_s$ for which $I_{i_s^r} \cap I_{i_s^p} \neq \emptyset$. $\bar{R}_s^p$ is the set R_s^p, ordered according the same principle as $\bar{N}_s$.

There has to be find the least number $\bar{p}$ for which $\bar{R}_s^{\bar{p}}$ contains at least $\mu_s + 1$ elements.

If such an $\bar{p}$ does not exist, we note this fact $L_s = \emptyset$ and pass to an other s. If $L_s = \emptyset$, $\forall s \in M$, then the considered fundamental problem gives at the same time an admissible solution of the studied problem.

3. If there exists such a $\bar{p}$, we retain the first $\mu_s + 1$ elements of $\bar{R}_s^{\bar{p}}$; they form the set L_s, which is a normal exceeding set of the type s with an exceeding interval

$$I_s = [a_s, b_s).$$

4. After examining all the types $s \in M$, there has to be retained the set L_s with the least b_s and this will be used for the branching of the studied fundamental problem.

REFERENCES

[1] Bertier P., Roy B., *Une procédure de résolution pour une classe de problèmes pouvant avoir un caractère combinatoire.* Bull. du Centre Intern. de calcul, Rome, **4**, 1, 19–28 (1965).

[2] Descamps R., Chevignon P., *Algorithmes d'optimisation pour une classe générale de problèmes d'ordonnancement avec limitations de ressources.* METRA **V**, 4, 603–618 (1966).

[3] Ignall E., Schrage L., *Application of the Branch and Bound Technique to some Flow-Shop Scheduling Problems.* JORSA **13**, 3, 400–412 (1965).

[4] Little B. D., Murty K. G., Sweeny D. W., Karel G., *An Algorithm for the Travelling Salesman Problem.* JORSA **11**, 972–982 (1963).

[5] Németi L., *Das Reihenfolgeproblem in der Fertigungsprogrammierung und Linearplanung mit logischen Bedingungen.* Mathematica (Cluj), **6**, 1, 87–99 (1964).

[6] Németi L., Radó F., *Sur la programmation temporelle de la fabrication.* Mathematica (Cluj), **8**, 1, 109–111 (1966).

[7] Radó F., *Programare liniară cu condiții logice.* Comunic. Acad. R.P.R., **13**, 1039–1042 (1963).

[8] Radó F., *Un algorithme pour résoudre certaine problèmes de programmation mathématique.* Mathematica (Cluj), **6**, 1, 105–116 (1964).

[9] Roy B., *Cheminement et connexité dans les graphes.* Application aux problèmes d'ordonnancement. METRA, Série spéciale no. 1, 1962.

[10] Roy B., Sussmann B., *Les problèmes d'ordonnancement avec contraintes disjonctives.* SEMA, Direction scientifique, Rapport de recherche No. 9, 1964.

Received, 7 IX, 1970.