ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1992
Seria: MECHANIKA z. 108 Nr kol. 1161

International Conference on
COMPUTER INTEGRATED MANUFACTURING
Internationale Konferenz über
RECHNERINTEGRIERTE FERTIGUNGSSYSTEME
Zakopane, March 24-27 1992

YANG DEHONG

The Department of Information Engineering
Beijing Metallurgical Management Cadres Institute
Beijing , P.R.China

APPLICATION OF MULTI-STAGE PROGRAMMING METHOD TO SCHEDULING PROBLEMS IN MACHINE MANUFACTURING PROCESS WITH COMPLEX SYSTEMS

Summary. In this paper,the scheduling problem at complex systems is formulated .The problem has been solved with aid of multistage programming method. The algorithms of solving the problem is given.

1. Introduction

The scheduling problem is very important in controlling of industrial processes.The system with a complex structure consists of machines with input-store and output-store , and objects which will be serviced at the machines . Each object has a lot of operations need to be done at different machines. Each object can be put into any input-store and taken out from any output-store. The problem is to find a optimum schedule . It is very difficult problem .In this paper , the scheduling problem at complex system has been solved with aid of multistage programming method .

The multistage programming method is proposed by professor Franciszek Marecki,Silesian Technical University, Gliwice,Poland. He succeeded in applying the method to solve the scheduling problems at the systems with typical structure and many other problems.The idea of this method was introduced in detail in his doctoral dissertation.

2. The multistage programming method

The multistage programming method is one of ways to solve combinatorial problems.It includes four basic concepts:1) state of decision process . 2) value of state . 3) state generation procedure.4)unperspective state elimination.A simple introduction of the concepts will be given in following .

2.1 State of decision process

For any multistage process , there are some feasible decisions can be made at each stage . So there will be a lot of decision ways.The state is a concept describing the decision process. From one state we can get the information that which stage the process reaches and what decisions have been made. After one decision has been made,the state will be changed . So from initial state we will get many state sequences . A state sequence is called a trajectory . The final state of each trajectory gives out a feasible solution of the problem.

In the process we discuss in this paper the stages are denoted by

$$0,1,2,\dots,e-1,e,e+1,\dots,E.$$

At e-th stage, propose there are L_e states, and we denote these states with

$$p^{e,1},p^{e,2},\dots,p^{e,l},\dots,p^{e,L_e}$$

In general, the state is defined as a vector or matrix according to the problem that will be solved.

2.2 The value of state

The value of state is a function of state $p^{e,l}$, and corresponds to the optimum criterion. We denote the value of state with $v^{e,l}$.

$$V^{e,l}=Z(p^{e,l})$$

where: Z(.) - a function dependents on the optimization criterion.

If the problem is with only one optimum criterion, the value of state can be defined as a scalar . If the problem is with the multi criterion , the value of state can be defined as a vector. Through comparing two state values , we can determine which state is better than another one.

2.3 The state generation procedure

To get a feasible solution of the problem,a feasible trajectory

$$p^{0,1},\dots,p^{e-1,k},p^{e,l},\dots,p^{E,l}$$

should be generated stage by stage . A state generation procedure is a mathematical formula that generates a new state $p^{e,l}$ from last state $p^{e-1,k}$.

2.4 The unperspective states elimination

The state p is unperspective one if the calculation of the optimum solution from it is impossible . For elimination of the unperspective states, the following three rules are often used.

1) the exhausting rule.
2) the domination rule.
3) the fathoming rule.

The exhausting rules eliminate the states which have not possibility to generate any feasible solution from it.
The domination rules eliminate the states from which the optimum final state can not be generated.
The fathoming rules eliminate the states which do not allow to find the final state better than current best one .

3. The scheduling problem at complex systems

Consider M machines, and each machine is with a input-store and a output-store . N objects should be serviced at machines . Each object has a lot of operations should be done.After one operation of a object has been finished , the object can be taken out from the output-store and be put into another input-store that next operation should be done at its machine . The system is shown on the figure 1. Where :

B_m^I - the input-store of machine A_m . A_m - the m-th machine

B_m^O - the output-store of machine A_m. M - the number of machine

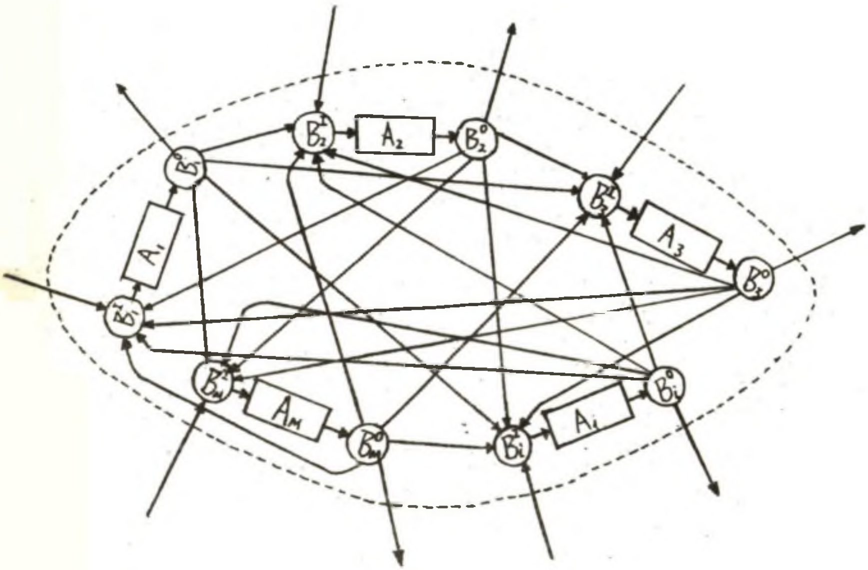

Figure 1. The system with complex structure

Any object can be put into system at any input-store and be taken out of system from any output-store . So the system is a complex system . In this paper the problem is discussed depending on following assumption .

3.1 Assumptions

-the system with the unlimited stores and independent objects.
-the M machines A_m, m=1,2,...,M.

-the N objects w_n, n=1,2,...,N.

-at any moment in each machine, only one object can be serviced.
-every object can be serviced in each machine only one time.
-all operations , which should be done to each object w_n , are known, and can be denoted by w_{n_1}, w_{n_2}, ..., $w_{n_{k_n}}$.

k_n-is the number of the operations of object w_n.

-the feasible technological routes , that each object w_n passes through some machines , are known , and can be described by the matrix

$$U_n=[u^n_{k,m}]_{K_n \times M}$$

$$n=1,2,3,...,N$$
$$k=1,2,3,...,K_n$$
$$m=1,2,3,...,M$$

where

$$u^n_{k,m}=\begin{cases} 1 & \text{if the operation } w_{n_k} \text{ can be serviced at } A_m \\ 0 & \text{otherwise} \end{cases}$$

-the sequencing relations among operations of object w_n are known , and it can be described by the matrix

$$S_n=[s^n_{i,j}]_{K_n \times K_n}$$

$$n=1,2,3,...,N$$
$$i=1,2,3,...,K_n$$
$$j=1,2,3,...,K_n$$

where

$$s_{i,j}=\begin{cases} 1 & \text{if operation } w_{n_i} \text{ must be finished before } w_{n_j} \\ 0 & \text{otherwise} \end{cases}$$

-each object can be put into any input-store and be taken out from any output-store .
-the service time that i-th operation of object w_n needs at machine A_m is known as $v_{n_i,m}$.

-the earliest moment φ_{n_i} at which the i-th operation of object w_n can be start is given.

-the latest moment ψ_n at which the all operations of object w_n should be finished is given .

-the shutdown time $\tau^{m}_{n1_i,n2_j}$ between the i-th operation of object w_{n1} and j-th operation of object w_{n2} is given .

-the number of the object α_m , which have been serviced as the last one in the machine A_m (before the scheduling period which is considered) is given .

3.2. The schedule optimization criterion

The goal of the scheduling is to find the optimum schedule. The optimization criterion that is considered in this paper is minimization of the maximum operation ending tardiness

$$Q = \max_{1 \leq n \leq N} (t_{L,n} - w_n) \rightarrow \min \qquad (1)$$

The $t_{L,n}$ is the moment at which the latest operation of n-th object has been finished .

4. Number-group-matrix and number-group-function

To define the state of this problem , we have to introduce two new concept.

Def.1 if the elements of a matrix are not scalar , but are number-groups,we call this matrix as number-group-matrix.

Def.2 if (a,b) is a number-group , G(.) can be defined as a number-group-function as follow

$$G(a,b)=b$$

and F(.) can be defined as a number-group-function as follow

$$F(a,b)=a$$

5. The algorithm

The algorithm depends on the multistage programming method.

5.1. The state

Def.3 The state $p^{e,l}$ is a number-group-matrix

$$P^{e,l} = [P^{e,l}_{i,j}]_{N \times M} \qquad (2)$$

$$i=1,2,3,...,N$$
$$j=1,2,3,...,M$$

For the matrix , the number of rows corresponds the number of objects , and the number of columns corresponds the number of machines.

The entries of the matrix are determining in the following way

$$p^{e,l}_{i,j}=\begin{cases}(k,t_{i,j})\text{ at stage } \eta, \eta\le e, \text{ if operation } w_i \text{ is finished at moment } t_{i,j} \text{ at } A_j{}^{i}{}_{k} \\ (0,0) \quad \text{otherwise}\end{cases}$$

The elements of the initial state $p^{0,1}$ are equal to (0,0) . From the final state $p^{E,l}$ the feasible schedule can be read out.

5.2. The state value

The state value function which is defined as following corresponds to the optimization criterion (1).

Def.4 The state value is the scalar, which can be find from the formula

$$V^{e,l}=\max_{n\in I^{e,l}}\left[\max_j(t_{n,j})-\psi_n\right] \tag{3}$$

where

$$I^{e,l}=\{n : \sum_{j=1}^{M} R(p^{e,l}_{n,j})=K_n\} \tag{4}$$

and

$$R(P^{e,l}_{i,j})=\begin{cases}1 & \text{if } p^{e,l}_{i,j}\ne(0,0) \\ 0 & \text{otherwise}\end{cases} \tag{5}$$

The optimum global state is determined from the condition

$$\min_{1\le l\le L_E}(V^{E,l}=V^{E,l^{o}})\Rightarrow(P^{E,l^{o}}=P^{o})$$

5.3 The state generation procedures

For the scheduling problem discussed in this paper , the state generation procedure has the form

$$\forall\forall_{n\,n_k}\forall_m\left\{(p^{e-1,l}_{n,m}=0)\wedge(u^{n}_{k,m}=1)\wedge(\forall_i(s^{n}_{i,k}=1)\Rightarrow\exists_{1\le j\le M}F(p^{e-1,l}_{n,j})=i)\right\}\Rightarrow$$

$$\Rightarrow(p^{e,l}=p^{e-1,l}+\delta p) \tag{6}$$

The elements of the number-group-matrix $\delta P=[\delta p_{i,j}]_{N\times M}$ are defined as following

$$\delta p_{i,j}=\begin{cases}(k,t_{n,m}) & \text{if } (i=n)\wedge(j=m) \\ (0,0) & \text{otherwise}\end{cases} \tag{7}$$

The moment $t_{n,m}$ is defined by formula as following

$$t_{n,m}=\max(\varphi_{n_k,m},\ \tau^{m}_{h_m,n_k}+T^{e-1,l}_{m})+v_{n_k,m} \tag{8}$$

Where: $T_m^{e-1,l}$ -the time at which the latest operation has been finished in state $p^{e-1,l}$ at machine A_m.

h_m -the number of the last operation serviced at machine A_m according to the state $p^{e-1,l}$.

The time $T_r^{e-1,l}$ is defined from the formula

$$T_m^{e-1,l} = \max_{1 \le i \le N} G(p_{i,m}^{e-1,l}) \tag{9}$$

The number h_m is determined from the formula

$$h_m = \begin{cases} i_r & \text{if } (t_{i,m} = T_m^{e-1,l} > 0) \wedge (F(p_{i,m}^{e-1,l}) = r) \\ \alpha_m & \text{otherwise} \end{cases} \tag{10}$$

5.4. Unprespective state elimination

Exhausting rule:

Theorem 1. The state p is exhausted if the condition is held

$$\exists_{n_k} (\forall_j F(p_{n,j}) \ne k) \wedge (\forall_m (u_{k,m}^n = 1) \wedge (p_{n,m} \ne 0)) \tag{11}$$

Domination rule:

Theorem 2. The state p is dominated by the state $p^{e,l}$ if the condition is held

$$\forall_{1 \le n \le N} \forall_{1 \le m \le M} \{(p_{n,m}^{e,l} = 0) \Leftrightarrow (p_{n,m} = 0)\} \wedge \{\forall_{1 \le n_k \le K_n} (\exists_j F(p_{n,j}^{e,l}) = k \Leftrightarrow \exists_i F(p_{n,i}) = k)\}$$

$$\wedge \{T_m^{e,l} \le T_m\} \wedge \{V^{e,l} \le V\} \tag{12}$$

Fathoming rule:

Theorem 3. The state p is fathomed if the condition is held

$$V^c \le V \tag{13}$$

where V^c is the value of the current best final state .

6. Final remarks

In the paper the multistage programming was applied to scheduling problem at complex systems . A feasible algorithm is given. If the system is with more complex structure, the problem will be more difficult than the one in this paper.The application of multistage programming method to the other problems are presented in others paper.

I would like to thank professor Franciszek Marecki , who as my professor gave me many good suggestion in my working.

REFERENCES

[1] Marecki F. : Draft of the theory of discrete multistage industrial processes . Dissertation , Silesian Technical University , Gliwice , Poland 1985 .
[2] Marecki F. : Application of a multistage programming method to scheduling at a system with parallel structure . Beijing Metallurgical Management Cadres Institute Papers . 1991.1 .
[3] Marecki F. : Assembly line balancing problem.Wuhan Iron and Steel University Scientific Papers. 1988. 6.
[4] Garfinkel R.S., Nemhauser G.L. : Integer Programming , John Wiley and sons , New York 1972 .
[5] Bellman R. , Dreyfus S. E. : Applied Dynamic Programming Princeton University Press , 1962.

ANWENDUNG DER MULTIETAPPIGEN PROGRAMMIERUNGSMETHODIK FÜR DAS PROBLEM DER MARMONOGRAMMIERUNG IN DEN KOMPLEXEN SYSTEMEN

Kurzfassung

Im referat wurde das Problem der Harmonogrammirung von Prozessen mit komplexen Stauerungssystemen besprochen. Zur Lösung dieses Problems wurde eine multietappige Programmirungsmethodik ausgenutzt und entsprechender Algorithmus vorgeschlagen.

ZASTOSOWANIE METODY PROGRAMOWANIA WIELOETAPOWEGO DO POBLEMOW PLANOWANIA W PROCESIE WYTWARZANIA W SYSTEMACH KOMPLEKSOWYCH.

Streszczenie

W pracy przedstawiono problem planowania w kompleksowych systemach.Problem został rozwiązany metodą programowania wieloetapowego. Podano również algorytm rozwiązania.

Wpłyneło do redakcji w styczniu 1992 r. Recenzent: Jan Kosmol