Franciszek MARECKI

Department of Computer Science Systems
Technical University of Łódź
Bielsko Biała Branch

BUFFER STORE OF LINE - TYPE MODELLING

Summary: The paper presents the mathematical model of a buffer store. The store consist of automatic lines of FIFO type. Loading and unloading control consist in the choice of the line.

1. Introduction

Buffer stores of various types occur in the computer integrated production [3], [4]. The process of storing is modelled by means of arithmetical-logical equations of state [1]. These equations allow to construct computer simulator program for control purposes [2]. The paper presents mathematical model of buffer store. The store consist of type FIFO lines and of control points: loading and unloading ones.

2. Problem formulating

Let's consider the system composed of two machines with buffer stores between them. The machines serve object of various types. During the change of the type of the served object the machine expects the change of the tool. The expectation time of the machine depends on the type of the two objects following each other.

From the point of view of minimalization of the total expectation time of the machine - the optimum sequence of serving the objects may be marked. The optimum sequences of serving objects for every machine are different in the general case. For this reason compromising solutions are marked.

The buffer store between machines plays important part in the optimization of the production schedule. This store allows to minimize times of expectation of the second machine. The object that exist in the store and minimizes the expectation time, is subjected to the service in the second machine.

Buffer stores have limited capasity. Moreover, various constructions of these stores are used, e.g. linear (FIFO), pile (LIFO), round robin etc.

A model of the multilinear buffer store will be presented in the paper. The store consist of many independent lines with the common loading point and with the common unloading point.

Let's assume that the objects, that have been served with the optimum sequence for the first machine, are delivered to the store. These objects will be placed in the chosen store line.

The objects are moved to the furthest free position in the buffer store lines. The object from the last position of the line may be delivered to the unloading point - and further on, to the service in the second machine.

Moments and types of the objects occuring in the loading point are random. The stream of object in the loading point determines extortion of the system. The moments of taking object from the unloading point to the second machine may be treated analogically.

The objects may be introduced into the store on the first position of the chosen line. The objects may be taken out from the last position of the store of the chosen line in the analogical way.

We assume that all the times of changes of tools in the second machine are known before starting the service of the next object. The problem of control a buffer store consist in defining:

- the line where the next object should be unloaded
- the line where the next object should be loaded.

The minimization of the global shutdown time of the second machine is accepted as the criterion of control.

3. Mathematical model

Let's assume that the buffer store consist of M lines, and every line contains N positions.

We accept the following designations:

m - number of lines, (m=1,...,M)

n - number of positions, (n=1,...,N)

Let's assume that there are I object types in the served system. The matrix of shutdown times for the change of the tool is given:

$$\mathbf{T} = \left[t_{i,j}\right] \qquad \begin{matrix} i = 1,\dots,I \\ i = 1,\dots,J \end{matrix} \tag{1}$$

where: $t_{i,j}$ - the time of the machine shutdown after the service of the i-th object and before the service of the j-th object.

The cycle "c", after which the object moves one position further within the buffer store line, is given. We accept that during the "c" cycle the object may be taken out:

- from the loading point to the first position of the chosen line,
- from the last position of the chosen line to the unloading point,

A buffer store works in cycles. Let's denote by: k - the cycle number, (k=1,...,K).

We accept the following priorities of cycle k:

I). The object is unloaded if the unloading point in k - 1 cycle was empty and if the object was on the last position of the chosen line.

II). The object is moved to the following position if this position has been free in the cycle k-1

III). The object is loaded if it was in the loading point in the cycle k - 1, and when the first position of the chosen line was free.

Definition 1:

The state of the buffer store after cycle k is the following matrix

$$\mathbf{Z}^k = \left[z_{m,n}\right] \qquad \begin{matrix} m = 1,\ldots,M \\ n = 1,\ldots,N \end{matrix} \tag{2}$$

where

$$z^k_{m,n} = \begin{cases} i & \text{if there is the object of type i in k cycle on n position of m line} \\ 0 & \text{if the n position of the m line is empty in the k cycle} \end{cases} \tag{2.a}$$

We also accept:
- the state of loading point

$$x^k = \begin{cases} i & \text{if the object of type i exists in the loading point in this cycle} \\ 0 & \text{if the loading point was empty in this cycle} \end{cases} \tag{3}$$

- the state of unloading point

$$y^k = \begin{cases} i & \text{if the object of type i exists in the unloading point in cycle k} \\ 0 & \text{if the unloading point was empty in this cycle} \end{cases} \tag{4}$$

The initial state of the system is defined by:

x^0, y^0, $\mathbf{Z}^0$

and the final state is defined by:

x^k, y^k, $\mathbf{Z}^k$

The number of cycles results from the time of the work of the system.

Definition 2:

The following vector is the control in cycle k

$$\mathbf{U}^k = \left[u_l^k\right] \qquad l = 1,2 \tag{5}$$

where

$$u^k_1 = \begin{cases} m & \text{if the object in cycle k should be loaded into line m} \\ 0 & \text{in the oposite case} \end{cases} \tag{5.a}$$

and

$$u^k_2 = \begin{cases} m & \text{if the object in cycle k should be unloaded from line m} \\ 0 & \text{in the oposite case} \end{cases} \tag{5.b}$$

The admissible control must fulfil the following condition:

$$(u_1^k = m) \Rightarrow \left[(z_{m,1}^{k-1} = 0) \wedge (x^{k-1} > 0)\right] \tag{6.a}$$

and

$$(u_2^k = m) \Rightarrow \left[(z_{m,1}^{k-1} > 0) \wedge (y^{k-1} = 0)\right] \tag{6.b}$$

Moreover we assume that

$$\left[(x^{k-1} > 0) \wedge (u_1^k > 0)\right] \Rightarrow (x^k = 0) \tag{7.a}$$

and

$$\left[(y^{k-1} = 0) \wedge (u_2^k > 0)\right] \Rightarrow (y^k > 0) \tag{7.b}$$

The record of the state is kept at the end of the cycle.

4. Equations of state

The buffer store state changes after every cycle. These changes depend on the accepted control.

Equations of state have the general form:

$$x^k = f_x(x^{k-1}, \mathbf{Z}^{k-1}, u_1^k, d_x) \qquad (8)$$

$$y^k = f_y(y^{k-1}, \mathbf{Z}^{k-1}, u_1^k, d_y) \qquad (9)$$

$$z^k = f_z(x^{k-1}, y^{k-1}, \mathbf{Z}^{k-1}, u_1^k, u_2^k) \qquad (10)$$

where: d_y, d_x- exterior interference

The state of the unloading point is marked in the following way:

$$y^k = \begin{cases} y^{k-1}, & \text{if} \quad (y^{k-1} > 0) \wedge (d_y = 0) \\ z_{m,N}^{k-1}, & \text{if} \quad (y^{k-1} = 0) \wedge (u_2^k = m) \\ 0, & \text{if} \ (y^{k-1} > 0) \wedge (d_y = 1) \vee (y^{k-1} = 0) \wedge (u_2^k = 0) \end{cases} \qquad (11)$$

The state of the store is marked in the following way:

$$\underset{1 \le m \le M}{\forall} z_{m,N}^k = \begin{cases} z_{m,N-1}^{k-1}, & \text{if} \quad (z_{m,N}^{k-1} = 0) \\ z_{m,N}^{k-1}, & \text{if} \quad (z_{m,N}^{k-1} > 0) \wedge (u_2^k = 0) \\ 0, & \text{if} \quad (z_{m,N}^{k-1} > 0) \wedge (u_2^k = m) \end{cases} \qquad (12)$$

- the positions from 2 to N - 1:

$$z_{m,n}^k = \begin{cases} z_{m,n-1}^{k-1}, & \text{if} \quad (z_{m,n}^{k-1} = 0) \\ z_{m,n}^{k-1}, & \text{if} \quad (z_{m,n+1}^{k-1} > 0) \wedge (z_{m,n}^{k-1} > 0) \\ 0, & \text{if} \quad (z_{m,n}^{k-1} > 0) \wedge (z_{m,n+1}^{k-1} = 0) \end{cases} \qquad (13)$$

- the first position

$$z_{m,1}^k = \begin{cases} x^{k-1}, & \text{if} \quad (z_{m,1}^{k-1} = 0) \wedge (u_1^k = m) \\ z_{m,1}^{k-1}, & \text{if} \quad (z_{m,2}^{k-1} > 0) \\ 0, & \text{if} \ (z_{m,1}^{k-1} = 0) \wedge (u_1^k = 0) \vee (z_{m,1}^{k-1} > 0) \wedge (z_{m,2}^{k-1} = 0) \end{cases} \qquad (14)$$

The state of the loading point is marked in the following way:

$$x^k = \begin{cases} d_x, & \text{if} \quad (x^{k-1} = 0) \\ x^{k-1}, & \text{if} \quad (x^{k-1} > 0) \wedge (u_1^k = 0) \\ 0, & \text{if} \quad (u_1^k > 0) \end{cases} \tag{15}$$

Equations of state allow to mark the state of the store in the following cycles from k=1 to k=K

5. Final remarks

Equations of state allow to create a computer simulator program of control a buffer store. In the simulator, control: $u_1{}^k$ and $u_2{}^k$ should be defined for every cycle k. The total time of the change of tools in the machine is accepted as the criterion of control. Therefore the following may be added up:

$$q = \sum_{k=1}^{k=K} q^k \rightarrow \min \tag{16}$$

where: q^k - time of the machine expectation in the cycle k
Components q^k are marked in the following way:

$$q^k = \begin{cases} t_{i,j}, & \text{if} \quad (x^k = j) \wedge (x^l = i) \text{ and for } 1 < r < k \\ & \qquad (u_2^r = 0) \\ 0, & \text{in oposite case} \end{cases}$$

Heuristic control rules may be applied for minimizing index (16).

6. References

[1] Rasztabiga D., Marecki F. : MODELLING OF A ROLLING-MILL LINE, International Conference on: "Computer Integrated Manufacturing", Zeszyty Naukowe Politechniki Śląskiej, seria: MECHANIKA z. 108, Gliwice 1992, ss. 323-334.

[2] Marecki F., Ptasznik K. : SYMULATOR STEROWANIA LINIĄ MONTAŻOWĄ, Zeszyty Naukowe Pol..Śl. seria: AUTOMATYKA z.101, Gliwice 1990, str. 181-191.

[3] Czarnota J. :EKSPERCKI SYSTEM STEROWANIA MAGAZYNEM WYSOKIEGO SKŁADOWANIA, II Krajowa Konferencja Naukowa nt.

"Inżynieria Wiedzy i Systemy Ekspertowe", ISiTS Politechniki Wrocławskiej, Wrocław 1993, tom II str. 493-501.
[4] Marecki F. : EKSPERCKIE SYSTEMY STEROWANIA PROCESAMI DYSKRETNYMI, II Krajowa Konferencja nt. : "Inżynieria Wiedzy i Systemy Ekspertowe", ISiTS Politechniki Wrocławskiej, Wrocław 1993, tom II, str.376-381.

Revised by: Jan Kaluski