SIMULATION MODEL OF A SERIAL PRODUCTION SYSTEM

Dimitri GOLENKO-GINZBURG
Prof, Department of Industrial Engineering and Management (Emeritus),
Ben-Gurion University of the Negev, Beer-Sheva, Israel
& Department of Industrial Engineering and Management,
Ariel University of Samaria, Ariel, Israel

E-mail: dimitri@bgu.ac.il

Doron GREENBERG
Pro PhD, Department of Economics and Business Administration, Faculty of Social Science,
Ariel University of Samaria, Ariel, Israel

E-mail: dorongreen@gmail.com

Nitzan SWID
PhD Candidate, Industrial Engineering and Management Department,
Ariel University of Samaria, Ariel, Israel
& Department of Industrial Engineering and Management,
Bar-Ilan University, Ramat-Gan, Israel

E-mail: nitzansw@effectivy.net

Abstract:
A simulation model describing serial production is outlined. Production process is carried out under random disturbances. The control algorithm of the model is based on the analysis of essential states. Decision-making is based on preference rules. The model can be applied to all types of working shops or sections.

Key words: *serial production; simulation model; preference rules; method of essential states; randomised rules.*

1. INTRODUCTION

Let us consider a simulation model describing serial production at a working shop or section [1,2]. Assume that the shop consists of L groups of equipment, each of which $\ell_{\hbar}$, $\hbar = 1,2,...,L$, having $m_{\hbar}$ machines or units of the same type. During the planning horizon $[T_0, T_{pl}]$, N batches of parts are processed within the shop, each consisting of n_i, $i = 1,2,...,N$, parts of the same type. Directive time limits T_i are set for operating each batch.

An arbitrary part D_i in the n_i-th batch goes through a certain number of operations O_{ij}, $j = 1,2,...,Q_i$, on different groups of equipment, different operations possibly being performed on one and the same group of equipment, $\ell_{j_1} = \ell_{j_2}$.

Each technological operation is characterized by a number for the group of equipment and the duration of the operation (values ℓ_j and t_{ij}). All operations on part D_i are carried out in a definite technological sequence $\{O_{ij}\}$, $j = 1,2,...,Q_i$, which must not be disrupted.

Each group of machines in the $\ell_{\hbar}$-th group of equipment handles a queue of parts waiting to be processed on that group of machines. The queue discipline at moment t is formed by randomized preference rules, i.e., parts are assigned for processing at a frequency in proportion to the value of preference function $F(p_i)$.

It is convenient to assume the preference function equal to a value inversely proportional to the position rank of part p_i, denoting the deadline time required to accomplish processing the part by symbol T_i. The position rank may be then calculated by

$$p_i = \delta\left(T_i - \sum_{j \in A_i} t_{ij} - t\right), \tag{1}$$

where $\delta = \begin{cases} 1 \ when \ T_i - \sum_{j \in A_i} t_{ij} - t \geq 0, \\ 0 \ otherwise. \end{cases}$

Here t denotes the current moment of time, A_i stands for the set of operations on the i-th part still being uncompleted. The preference function is determined by

$$F(p_i) = \frac{n_i}{p_i \cdot \sum_{i \in B} \frac{n_i}{p_i}}, \tag{2}$$

where B denotes the set of parts on queue at moment t when rank position p_i ($i \in B$) is calculated, and n_i stands for the number of parts of the i-th batch unprocessed.

2. THE SIMULATION ALGORITHM

The simulation model's algorithm is based on the following information about each batch of parts stored in a separate Array I:

a) i is the number of the batch of parts;

b) T_i is the directive deadline for processing the batch;

c) n_i is the quantity of parts in the batch;

d) ℓ_j, t_{ij} are the numbers of groups of equipment and the time for performing the operation on the machines of the group (placed in order for the technological processing of the parts), respectively.

A separate Array II provides information on the groups of equipment, which includes the number of the group of equipment ℓ_j and the number of units of equipment m_j.

The algorithm simulates advancement of parts from operation to operation, as well as processing of parts; in particular, it simulates the corresponding changes of information about the part. This includes i, the number of the batch; f_i, the number of the part in the batch; O_{ij}, the number of the routine operation on the part; and $t_{i,j-1}$, the termination moment of the preceding operation.

By storing and processing this information, we can simulate the individual processing of each part in the batch and obtain the total characteristics necessary for simulating the system as a whole.

We will employ the following symbols for recording the flow chart of the simulation model's algorithm: A denotes calculating blocks, F - blocks for simulating random variables, T - blocks for transforming and processing information; L - blocks for checking logical conditions, K is a counter, and Z is the block for terminating computation and providing final results of the simulation. Symbol A^m means that upon the block's completing the procedure, we must unconditionally proceed to block m. Also, $L^{m,n}$ testifies that a check of logical conditions is required, and depending on its results, go to either block m or n. In all other cases, when the corresponding index in the upper right part of the block is absent, proceed to the next block of the algorithm. The index in the lower right part of each block designates its ordinal number in the logical structure of the algorithm.

This is the flow chart of the simulation model's algorithm:

$T_1 \quad T_2 \quad L_3^{1,4} \quad T_4 \quad T_5 \quad T_6 \quad A_7 \quad L_8^{6,9} \quad T_9 \quad F_{10} \quad L_{11}^{9,12} \quad L_{12}^{5,13} \quad T_{13} \quad T_{14} \quad T_{15} \quad L_{16}^{17,18} \quad T_{17} \quad L_{18}^{15,19}$
$T_{19} \quad F_{20} \quad L_{21}^{19,22} \quad T_{22} \quad A_{23} \quad L_{24}^{22,25} \quad T_{25}^{26,27} \quad A_{26}^{23} \quad T_{27} \quad L_{28}^{22,27} \quad L_{29}^{13,30} \quad L_{30}^{31,32} \quad A_{31}^{30} \quad A_{32} \quad L_{33}^{44,34}$
$A_{34} \quad L_{35}^{38,36} \quad L_{36}^{37,33} \quad A_{37}^{44} \quad A_{38} \quad L_{39}^{40,42} \quad T_{40} \quad L_{41}^{45,43} \quad T_{42} \quad T_{43} \quad L_{44}^{46,32} \quad K_{45} \quad L_{46}^{47,4} \quad A_{47} \quad K_{48}$
$L_{49}^{50,1} \quad A_{50} \quad Z_{51}$.

The main idea of the simulation model's algorithm boils down to a combination of the method of essential states [1,2] with the simulating-at-a-constant-speed technique [1,2]. Let us briefly outline the algorithm blocks' functioning in greater detail:

Blocks 1-3 (in cycle) form array M_I of "accompanying cells" for the initial information on each batch. From the array obtained, Block 4 forms a queue to the L groups of equipment. The address of the destination of the accompanying cell in the queue is determined on the basis of relation $r = k + 2i \cdot \ell_j$, where k is the address of an arbitrary memory cell of the computer.

When the array of accompanying cells is completely formed, Block 7 determines the value of the corresponding priority coefficient for each batch. The calculation takes into account all the parts of the batch, except those already processed.

Blocks 9-11 normalize values $F(p_i)$ in a way to comply with the conditions for normalizing and determining the probability within the given bounds: $0 \le F(p_i) \le 1$, $\sum_{i \in B} F(p_i) = 1$.

Blocks 5, 6, 8 and 12 form array M_{II} in cycle, occupying memory cells similarly to array M_I.

Before beginning to simulate the loading of the equipment by means of a random numbers generator, Block 14 singles out the number of machines assigned to the L groups of equipment. In order to reflect the work of the machines, a special array of memory cells M_{III} is assigned for this purpose.

Block 16 reveals unoccupied machines, while Block 17 memorizes their amount. Block 20 engages the random numbers generator as many times as the number of unoccupied machines. A machine is regarded unoccupied if condition

$$\sum_{i} F(p_i) < \xi_{fM} \le \sum_{i+1} F(p_i) \qquad (3)$$

holds, where ξ_{fM} are random independent values uniformly distributed in interval $[0,1]$, their quantity being equal to the number of unoccupied machines.

Block 23 evaluates $\sum_{i \in B} F(p_i)$, and Block 25 checks compliance with (3) for each random variable. If (3) holds, Block 27 memorizes the address of the batch sent to the machine and calculates the time value of the unproductive idleness of the parts in that batch.

Blocks 30 and 31 dispatch the values of the duration of processing each operation fed in, to unoccupied cells of the memory array M_{III}. After "loading" the machines, Block 32 changes the time counter by value Δt, and Block 34 subtracts the contents of the time counter from the operation processing duration. Block 34 is guided by Block 33 comprising the counter of the number of loaded machines. If the processing is accomplished, the contents of one or more cells $\langle \alpha \rangle$ of array M_{III} is equaled zero.

The analysis of array M_{III} is controlled by Block 35. When $\langle \alpha \rangle = 0$, we proceed to Block 38, which memorizes the time when the operation has finished processing. When $\langle \alpha \rangle > 0$, Block 36 applies to Block 33 in order to continue checking the contents of other array cells. When $\langle \alpha \rangle < 0$, Block 37 registers the machine's idle time.

Block 39 checks whether all the parts in the batch have been processed, Block 40 removes the batch from the queue and changes the operation number in the corresponding "accompanying cell" if all the parts in the given group of equipment are processed.

Block 41 checks whether any batch has finished processing. If a batch has not been completely processed, Block 42 changes the number of the part, and Block 43 reports that there is a released machine. Block 44 checks whether there are unoccupied machines. When H, return to Block 32 to continue simulating of processing the parts at the regarded operation. If the batch has been fully processed, Block 45 increases the number of processed batches by one.

Block 46 checks whether all the batches have been fully processed. If not, return to Block 4 to simulate processing of the remaining batches. When all the batches have been fully processed, Block 47 memorizes the total time for processing all the batches, summing up the values of non-productive idleness of the parts in all batches, and determines the values of idleness for all the machines.

Block 48 keeps the number of the iterative simulation cycle implemented, and Blocks 50-51 calculate the histogram, and print out the results of the simulation.

3. THE MODEL

It can be well-recognized that regarding simulation of large-scale serial (mass) production, it is characterized by the fact that assembly sections consume parts uniformly, while processing of parts is carried out in batches.

This is how the formalized flow chart of materials can be presented for such production. Assembly is ensured by sets of items and parts in special stores or bunkers, which are kept supplied by intermediate machine shops. No batch of parts is fed into production before the level of parts ready for assembly reaches a certain fixed value, called the order point. In turn, the machine shop, where the processing is to begin, places orders in the factory stores for the appropriate raw and semi-manufactured materials. The purpose of production we describe here is to ensure the assembly of parts needed with a given reliability at the minimum production expenditures, whose basic components are cost of equipment and of raw and semi-manufactured material reserves.

The simulation model on Fig. 1 is based on an analysis of essential states, such as the moment of the routine order for any batch of parts, the moment when processing begins, when transferring from one operation to another, when completing the processing, moment when equipment goes out of commission and is restarted, as well as beginning of the shift, month, or year.

The random parameters of the simulation model are:

1) the duration of non-stop work by machines and the time for repairing them;
2) the number of workers;
3) the number of discarded parts; and
4) time when there is lack of semi-manufactured or raw materials necessary for feeding a batch of parts into production.

The time for processing parts per operation is considered a deterministic value.

The simulation model is adaptive in that the order points can be corrected if the frequency at which the planned production program being not carried out for period $\left[0, T_{pl}\right]$

goes beyond the bounds of the planning horizon. After this, the entire simulation cycle is repeated upon setting the realization anew. There can also be corrections of preference rules Q when there are queues of batches of parts for the machines.

4. APPLYING PREFERENCE RULES

Unlike the preference rules considered in [1,2], which are used mainly in small-scale serial and serial production systems, preference rules in large-scale serial production are represented in the form

$$Q = \varphi\{t, t_s, k, k_f, \vec{\eta}, \vec{\eta}_\ell, t_i, t_w, t'_w, c, p, n\} , \quad (4)$$

where:

- t is the current time;
- t_s is the order moment;
- k is the number of operations to be performed;
- k_f is the number of operations completed by moment t;
- $\vec{\eta}$ is a vector, each i-th coordinate of which designates the coefficient of loading groups of equipment on which the i-th operation on the batch considered is carried out;
- $\vec{\eta}_\ell$ is the vector of coefficients of loading equipment for operations uncompleted by moment t;
- t_i, t_w, t'_w are the duration for processing a batch of parts at the i-th operation, the total processing time for all operations, and the total processing time for operations uncompleted by moment t;
- c is the cost of raw and semi-manufactured materials;
- p is the given reliability of supplying the assembly with ready parts of a given type; and
- n is the number of parts in the batch being processed.

$Q_1 = t - t_s$ is one of the simplest preference rules. It indicates the degree to which a batch of parts is behind the order point.

If we exclude the duration of processing parts in operations already completed from rule Q_1, we obtain rule $Q_2 = t'_w - t_w + t - t_s$, which characterizes the total idleness of the batch of parts in the course of operations done by moment t.

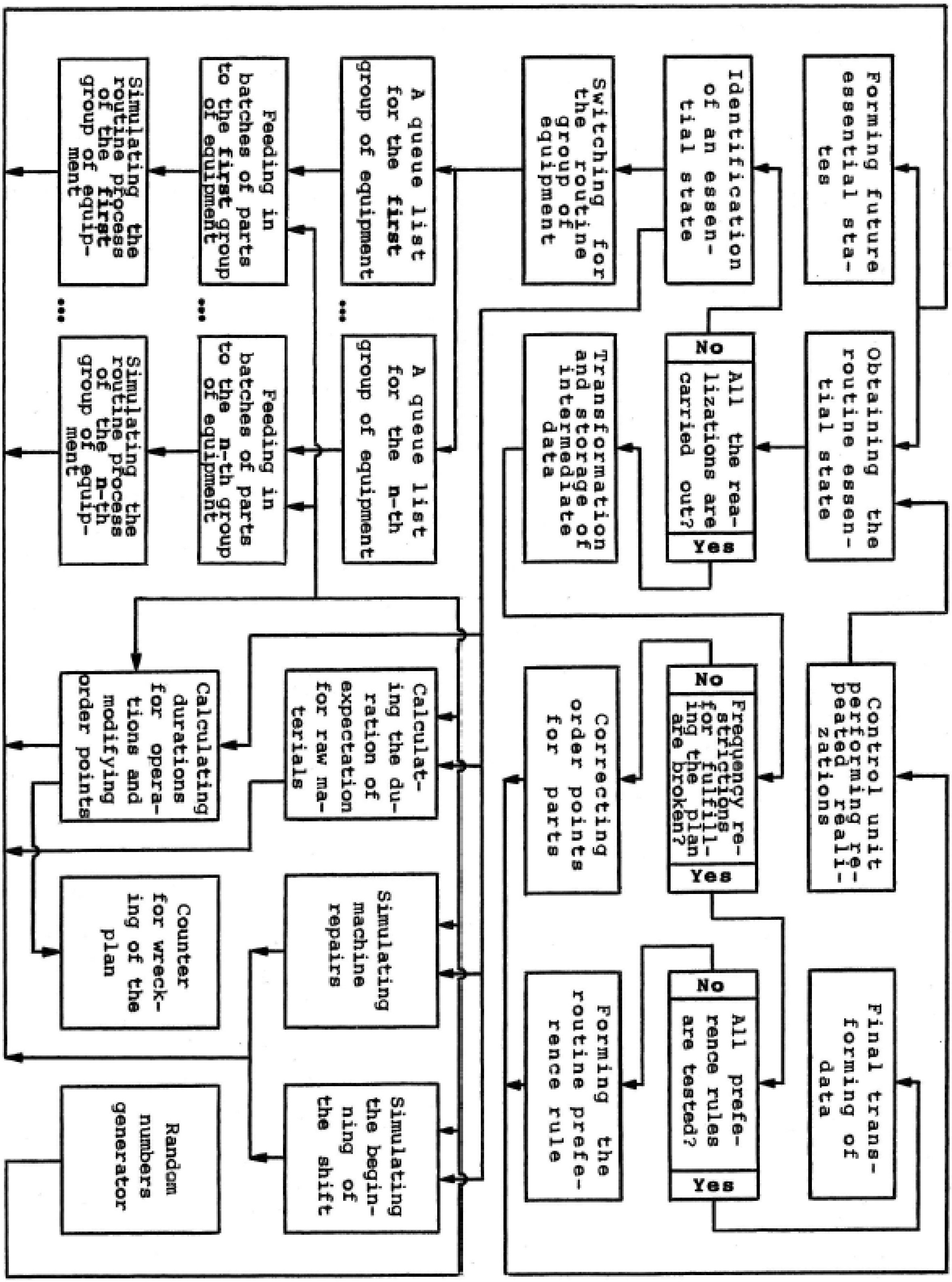

Figure 1. Flow-chart of the simulation model for large-scale production type

If we take into account the possibility of parts being idle in subsequent operations, preference should be granted to batches of parts designated to go through a large number of operations before processing is finished. In other words, the preference rule must forecast any idleness of the parts in future. In the simplest of cases, these considerations lead us to rules $Q_3 = Q_1 \cdot k/(k_f + 1)$ and $Q_4 = Q_2 \cdot k/(k_f + 1)$. A more accurate idleness forecast should account not only for the number of uncompleted operations, but also for coefficients of loading the respective groups of technological equipment. Examples of such rules are $Q_5 = Q_1 \cdot f_1(\vec{\eta})$; $Q_6 = Q_2 \cdot f_1(\vec{\eta})$; $Q_7 = Q_1 \cdot f_2(\vec{\eta})$; $Q_8 = Q_2 \cdot f_2(\vec{\eta})$, where

$$f_1(\vec{\eta}) = \frac{\sum_{i=1}^{k} \eta_i^2}{\sum_{i=1}^{k_f+1} \eta_i^2}, \qquad f_2(\vec{\eta}) = \frac{\sum_{i=1}^{k} \frac{1}{1-\eta_i}}{\sum_{i=1}^{k_f+1} \frac{1}{1-\eta_i}}, \tag{5}$$

It is natural to assume that other conditions being equal, we should prefer more expensive parts, as well as parts for which the given reliability of supply for assembly is higher. These considerations bring us to the following rules:

$$\begin{aligned} &Q_9 = Q_1(c + 3t_w); \quad Q_{12} = Q_1(1-p)^{-1}; \\ &Q_{10} = Q_5(c + 3t_w); \quad Q_{13} = Q_9(1-p)^{-1}; \\ &Q_{11} = Q_8(c + 3t_w); \quad Q_{14} = Q_{10}(1-p)^{-1}. \end{aligned} \tag{6}$$

It is natural to assume that other conditions being equal, we should prefer more expensive parts, as well as parts for which the given reliability of supply for assembly is higher. These considerations bring us to the following rules:

$$p_j = \frac{t - t_s^j}{\sum_{j \in \Im} (t - t_s)}. \tag{7}$$

The simulation model makes it possible to test all the listed rules and find the most efficient of them.

To conclude this paper it should be noted that optimization units do not enter into the simulation models for derail and large-scale serial types of production described above. These optimization units should be considered apart.

REFERENCES

1. Golenko-Ginzburg, D. **Hierarchical Control Models of Man-Machine Production Systems, Vol. 1: Fundamentals**, Lorman, Mississippi: Science Book Publishing House, 2012
2. Golenko-Ginzburg, D. and Kats, V. **Priority rules in job-shop scheduling**, Proceedings of the 14th Israel Congress on Advanced Technologies in Engineering, Management and Production, Tel-Aviv, June 7-9, 1994, pp. 41E-47E