Possibilities of Simulation of a Technical Line which is subject to the Balancing Process

Cristina COCULESCU
IT, Statistics and Mathematics Department
Romanian-American University, Bucharest, România

Technical line balancing is a mean of economic efficiency increasing of some industrial activities. Essentially, to balance a technical line means to organize human operator's activity, to establish manufacturing flux, to project the line so to minimize equipments and operators' pause time, having a use of these as good as possible. In this work we'll describe an algorithm of simulation of a technical process, relied on random generation of the main characteristics of working phases, which will be considered in solving balancing problem of a technical line whereon an only model of a product is assembled, operating times are established and line operating rhythm is fixed.
Keywords: *technical phase, workstation, operating time, technical line balancing, simulation.*

1 Introduction

In many cases, experiences on real objects are difficult to make and need a big amount of funds and human resources for make then acceptably.

Nowadays, grace of computing facilities of electronic computers, that are big amount of memory and computing speed, it acts modeling methods improving, complex systems modeling with simulation techniques having a privileged place. These offer us information about the "system" before its real making, through building of mathematic and logic models which show the behavior of real system (or of some features of it) during a long time.

By the meaning of simulation, the „inputs" are generated and considering system states, with suitable algorithms, the "outputs" are established and there is shown the evolution in time of system internal states. However simulation gives under-optimal solutions, simulation is an efficient technique for complex problem study at firm level.

In this work we'll describe an algorithm of simulation of a technical process, relied on random generation of the main characteristics of working phases, which will be considered in solving balancing problem of a technical line whereon an only model of a product is assembled, operating times are established and line operating rhythm is fixed.

2. Simulation of a Technical Line which is Subject to the Balancing Process

2.1. Possibilities of random number generation

For realistic reproduce certain elements of simulated system, and also for solving some numeric problems, it appears the need of existence of some numbers "happily chosen". In simulation programs, generation of random numbers has a relative big part of computer running time.

We'll mark $F = \{1,2,...,n\}$ the manifold of technical process' phases which follow to be simulated and $t = \{t(1), t(2),..., t(n)\}$, the vector of operating times associated to the phases.

The problem of technical process balancing supposes the grouping of phases in workstations so the sum of operating times of the phases which are in a station to be less then or equal to a value R called rhythm. For approached model of balancing problem, R is fixed and is the same for all stations. The phases of technical process must be executed in a certain order in the meaning that certain phases can't have place before others. Order relations between phases reflect technologically, real development of the process and are called relation of precedence. Mathematically, they form an acyclic digraph.

Balancing objective is the reaching of the smallest number of station having prior prop-

erty. The problem of simulation of a technical line that is subject to balancing is complex. There is easy to establish a number of phases and corresponding operating times. Also, there is easy to establish what the direct precedent phases are. Effectively we randomly choose them. But there is important to view whether what is randomly chosen is like a real process. If a phase precedes itself, the model is wrong. If a phase "*i*" must to operate before a phase "*j*" and "*j*" operates before "*i*", there is a mistake. And then, what we do?

Within this context, we purpose to build a function having property that it has equal likelihoods to generate its values. So, we consider the function **rnd** defined like this: for a natural number n, $rnd(n)$ is a natural number between 0 and n-1. If we want a real value, let's say a number between -2 and 3, having two exact digits, we use an expression like $rnd(500)/100-2$.

If maximum number of phases is marked *maxphases* the number of the phases from process will be $rnd(maxphases)+1$. We consider operating time of each phase at most equal to *maxtime* time units, therefore for the phase   *i*, $t(i) = rnd(100*\max time)/100 + 0.01$ (it is computed with at most two exact digits and cannot be zero).

As follows, we consider that arithmetic operations that will be used for generate such function (**rnd**), are with 32 bits that is, every result of an arithmetic operation is considered as the rest of the division of real result by 2^{32}. This value has chosen because it is enough for usual needs and is easy to be used in programming.

There is considered the string $\{x_n\}_{n\geq 0}$ defined like:

$$\begin{cases} x_0 = c \\ x_{n+1} = a*x_n+b \end{cases}, \; for \; n \geq 1 \qquad (8)$$

where *a* and *b* are odd numbers (*a* is a number of shape $a = 4*k+1$).

Such string, especially a and b are prime numbers, has an extreme long cyclicity and is good enough for usual needs. If to its value there is added also that of the standard func-

tion which generates random numbers, from the libraries of several programming languages, and it is set by the time counter and computer's internal clock, the result can be of a big variety. There is also the possibility of combination of two or three strings of mentioned type, their sum being a string having a very long cyclicity and in this case the result being very easy to compute too.

2.2. Acyclicity test of the graph of precedence relations

Let be *maxphases* maximum number of process' phases and *maxtime* maximum operating time of a phase. At the beginning we'll establish the number of phases (*nrphases*) and operating times of phases. For $i = 1, nrphases$

$$t(i) = rnd(\max time) + 0.01$$

The number of phases is randomly generated. After this, phases' names are read. For each phase, we generate then the operating time. Making a precedence relation follows. A precedence relation (shortly, a precedence) can be considered as a couple (f_1, f_2) where $f_1 = rnd(nrphases)$ $f_2 = rnd(nrphases)$ and $f_1 \neq f_2$. That means that f_2 operates before f_1. Clearly, the first care we have is not to have the relation already generated and $f_1 \neq f_2$.

After that, total operating time and maximum operating time for a phase (the biggest operating time of generated phases) are computed. This computing is very important because if in balancing problem, a rhythm is smaller than maximum working time for a phase, the problem is wrong.

After that, we must decide whether precedence relation we put in is correct. It not, it will be cancelled in the sense of cancellation of its value in precedence matrix. There are two ways to do the test. First, it is necessary to clarify what means that precedence (f_1, f_2) is correct, that is, if f_2 can precede f_1. The relation is correct if and only if no phase precede itself.

If we have precedence of type (f_1, f_2) and (f_2, f_1), between indirect predecessors of

f_1 is also f_1 as predecessor of f_2 and we reach up-mentioned contradiction. **Acyclicity test of precedence relations** reduces to establish if no phase precede itself. A checking method consists of the establishing of all predecessors of each phase. For this, we purpose the following algorithm:

Algorithm 1

1. We consider F_1 the manifold of f predecessors. Let $k = 1$.

2. $k = k + 1$. We consider F_k the manifold of direct predecessors of the phases from F_{k-1}. If $F_k \neq F_{k-1}$ we repeat step 2.

3. If $f \in F_k$, the precedence is not correct. Otherwise it is correct.

Another method supposes weight computing of all technical process phases. The weight of a phase is the sum of its time with times of all its direct or indirect predecessors. Therefore, if a phase has no predecessor, its weight is equal to its working time. If the phase has predecessors, there is computed the sum of the times of precedent phases with computed weights. To have an acyclic graph of precedence relations, we must compute the weight of all phases and reciprocally, if there is a phase whose weight can't be computed, the graph is cyclic.

Simulation Algorithm

As follows, we'll describe an algorithm of simulation of a technical process, in the meaning of random generation of the main characteristics of working phases that will be considered for solving balancing problem of a technical line whereon an only model of a product is assembled, operating times are established and line operating rhythm is fixed. The algorithm is the fruit of authors' research in the wide and very complex domain of the problematic of technical line balancing [1].

Algorithm 2

1. There is randomly generated the number of phases and for each phase, working time is generated.

2. Matrix *PRED* of precedence relations is initialized by zeros. This matrix has a number of lines equal to the number of phases and 20 columns.

3. $nrfail = 0$;

4. A triplet (f_1, f_2, p) is generated where $f_1 = rnd(nrphases)$, $f_2 = rnd(nrphases)$, $p = rnd(20)$. If $f_2 \neq f_1$ we make $PRED(f_1, p) = f_2$ in the meaning that the phase f_2 precedes f_1. If $f_1 = f_2$ or $PRED(f_1, f_2) \neq 0$, we repeat step 4. .

5. There is checked with algorithm 1 if the matrix *PRED* corresponds to an acyclic graph of precedence relations,

6. If the graph is cyclic, $PRED(f_1, p) = 0$ and if $nrfail < 10000$ go to step 4. Else STOP.

7. If the graph is acyclic, go to step 3.

3. Conclusions

Balancing using in industrial practice, can be facilitated by making a program which additional to a more efficient implementation of selected algorithms must offer the possibility of a comparative analysis of the methods used for solving balancing problem approached concerning the quality of given solutions. Because this action demands the using of a relative big number of problem instance, we considered necessary the making of possibility of random generation of these models, being known that their gaining from real technical processes is difficult.

The idea we started from, was that to build a function which to have equal likelihoods of generation its values. For this, we combined standard random number generator of C programming language, with the values of random string (8) of natural parameters *a, b, c*. By helping of gained function (*rnd*) there was realized a simulation model of real technical process in the meaning of random generation of the main characteristics of working phases that will be considered in solving balancing problem of a technical line whereon an only model of a product is assembled, operating times are established and line operating rhythm is fixed.

Simulation model and its informatics implementation have been then integrated in a program, proving very useful within the process

of testing performances of used methods for solving the approached balancing problem [1].

General rules from technical line simulation process shown in this work, can be applied for simulation other models, from other domain, especially of economic ones.

References
1.Coculescu, C. (2005), *Modeling and Simulation of Economic Processes. Concepts, Models and Balancing Algorithms*, Juristic Universe Editing Press, ISBN 973-8446-44-9, Bucharest
2.Filip, F.G., Bărbat, B. (1999), *Industrial Informatics – New Paradigms and Applications*, Technical Editing Press, ISBN 973-31-1324-7, Bucharest
3.Hoza, M.G. (2006), *XXIst Century enterprise. Intelligent Enterprise*, Economic Editing Press, ISBN 973-5904462, Bucharest
4.James, A.J. (2000), *Next Generation Manufacturing. Methods and Techniques*, Wiley, ISBN 978-0471360063
5.Kosanke, K., Jochem, R., Nell, J., Bas, A.O. (2002), *Enterprise Inter-and Intra-Organizational Integration*, Springer, ISBN 978-1402072772
6.Swamidass, P.M. (2002), *Innovations in Competitive Manufacturing*, AMACOM, ISBN 978-0814471401
7.Vernadat, F. (1996), *Enterprise Modeling and Integration*: *Principles and Applications*, Springer, ISBN 978-0412605505