G. M. AVRĂMIȚĂ
Central Institute for
Management and Informatics

Decomposition and Solving of Large-Scale Problems in Production Scheduling

The paper presents a proceeding for solving large-scale problems which can not be processed by the available computing configurations and packages.
The proceeding is based on decomposition of the large-scale problem into subproblems with reduced dimensions, which will be coordonated by sequential processing which takes into consideration a final goal. For the sake of exemplification a production-scheduling dynamic model is presented, which is decomposed into submodels.

1. INTRODUCTION

In the frame of industrial production processes are manipulated a large number of parts and an appreciable number of operations upon raw materials and materials are executed during the development of the technological flow. Successive movement of the raw materials and materials by each working place where one or more operations are executed, the inventory stock between operations, the phases of the technological process in work shops and departments produce a large number of state variables and control variables.

Modelling the dynamic process of production management and planning leads to models of very large size, which can not be solved by the actually, existing computing configuration.

An approach for the development of the model is to have at our disposal: the process chart for the subassemblies the supply schedule, the delivery schedule and the capacity available for each equipment [5].

A dynamic model which presents the evolution of the production process and which allows to obtain an production programme [1], [3], [4] contains:

— modelling the evolution in time of the production system;

$$x(k+1) = Ax(k) + Bu(k) + Cf(k) \tag{1}$$

— modelling the conditions for material continuity of the technological flow;

$$G_1x(k) + D_1u(k) \leqslant DM(k) \tag{2}$$

— modelling the condition that all orders must register within the available capacity;

$$G_2x(k) + D_2u(k) \leqslant DC(k) \tag{3}$$

— one shows the initial state of the system, that is the starting stocks for the planning and scheduling process

$$x(0) = x_0$$

$$x(k) \geqslant 0 \text{ and } u(k) \geqslant 0 \text{ for } k = 1, 2, ..., N \tag{4}$$

— the complete costs for manufacturing and stocking which has to be minimum

$$J = \min_{x,\,u} \left\{ \sum_{k=0}^{N-1} [c_1x(k) + c_2u(k)] \right\}. \tag{5}$$

By solving this model with $x(0)$, $f(k)$, $c(k)$, $k = 1, N$ given, a production programme is obtained under form of control variables and state variables set: $x(0)$, $u(0)$, $x(1)$, $u(1)$, ..., $x(N-1)$, $u(N-1)$, $x(N)$, that represent the manufacturing orders for the scheduled interval as well as the stock evolution in this same interval.

It must be marked that the main element which is taken into account for the production scheduling activity is time, by means of which the dynamics (evolution in time) of the production process is percieved.

The need of representing the dynamics of the production process led to a systemic treatment in the sense of emphasizing the states, the controls and the dynamics of the production system.

The dynamic equations (1) describe the development in time and space of the production process, starting from raw materials and half-finished products, moving through processing operations of parts and subassemblies and finishing with the assembling process, according to the actual technological flow as emphasized by the process of chart. Because the production process develops within given material conditions, existing raw materials and capacities, are imposed upon the dynamics of the system, a series of constrains

(2), (3) which are modelling the real conditions of the production process development.

Starting from the goals set by the activity of production scheduling, which demand the elaboration of a production programme and the achievement of the planned production at term and in corresponding quantities with minimum expenditures, we define the performance function (5) of the system as the minimization of production costs and of unfinished production stocks, since the goals related to the achievement of the quantities to be delivered at planned terms are contained in the dynamic and in the model's constraints.

Having in view the large size of production planning and scheduling problems a decomposition is necessary, due to the limited possibilities offered by the computer. Decomposition itself can be done technologically — by departments, working shops, groups of homogeneous equipments — starting from the process chart, which will allow a reduction in size and therefore solving.

2. A PROCEEDING FOR TECHNOLOGICAL DECOMPOSITION

Consider that we have the dynamic model for a subassembly. To reduce the size of the problem we proceed, as mentioned above, by decomposition of the large problem into subproblems, from the technological point of view, that is by solving subproblems by departments, working shops or equipment group, depending upon the place where operations are executed on parts. Subproblem solving will be made sequentially, starting from the last department, where outputs are known, and step-by-step solve the whole problem, determining the inputs for the last department, inputs which are outputs for the last but one department a.s.o.

Having to solve the optimal control problem (1)—(5) for the model in figure 1, the procedure will be the following: we solve successively the subproblems for work shop painting, work shop magnets, work shop surface protection, work shop mechanical presses and turning shop. That is:

$$x^n(k+1) = A^n x^n(k) + B^n u^n(k) + c^n f^n(k)$$

$$G_1^n x^n(k) + D_2^n u^n(k) \leqslant DM^n(k)$$

$$G_2^n x^n(k) + D_1^n u^n(k) \leqslant DC^n(k)$$

$$x^n(0) = x_0^n$$

$$x^n(k) \geqslant 0 \text{ and } u^n(k) \geqslant 0 \text{ for } k = 1, 2, \ldots, N$$

$$J^n = \min_{\substack{x^n,\ u^n \\ x^n,\ u^n}} \left\{ \sum_{K=0}^{N-1} [c_1^n x^n(k) + c_2^n u^n(k)] \right\}$$

for the last work shop (painting) a.s.o. up to the first work shop turning):

$$x^1(k+1) = A^1 x^1(k) + B^1 u^1(k) + C^1 f^1(k)$$

$$G_1^1 x^1(k) + D_1^1 u^1(k) \leqslant DM^1(k)$$

$$G_2^1 x^1(k) + D_2^1 u^1(k) \leqslant DC^1(k)$$

$$x^1(0) = x_0^1$$

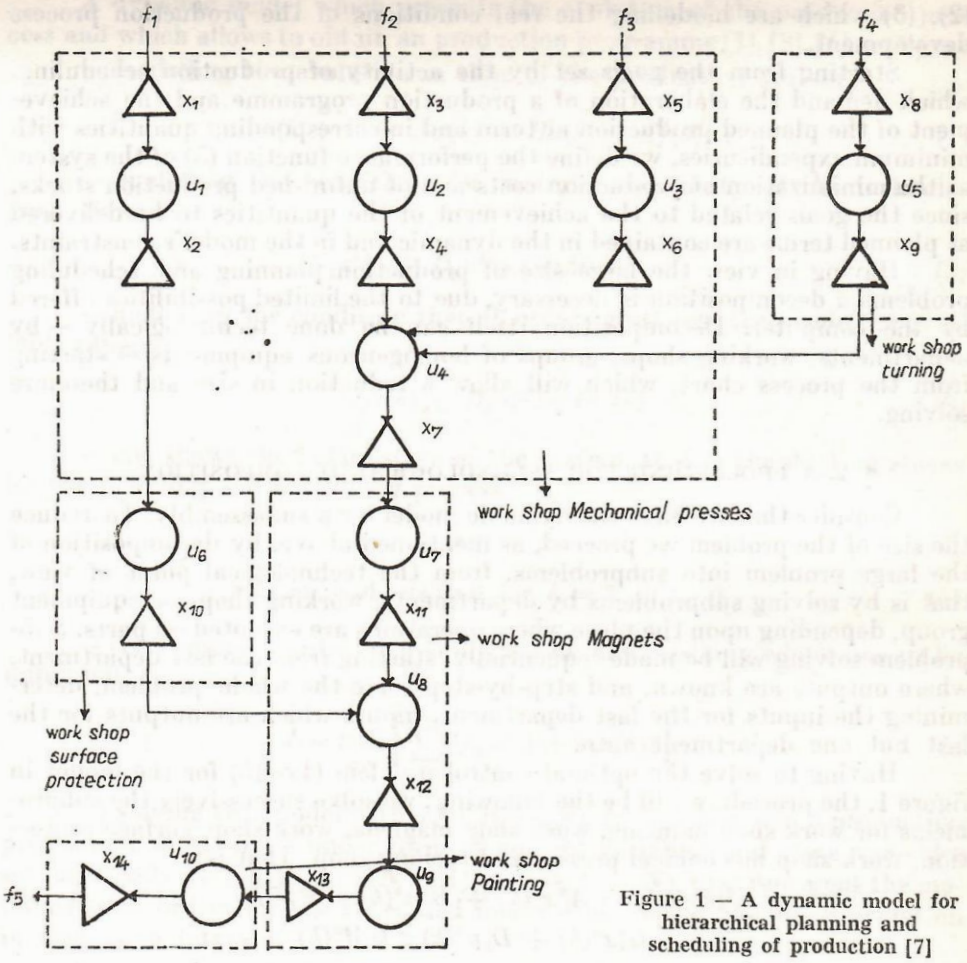

Figure 1 — A dynamic model for hierarchical planning and scheduling of production [7]

$$x^1(k) \geqslant 0 \text{ and } u^1(k) \geqslant 0 \text{ for } k = 1, 2, ..., N$$

$$J^1 = \min_{x^1,\, u^1} \left\{ \sum_{K=0}^{N-1} [c_1^1 x^1(k) + c_2^1 u^1(k)] \right\}.$$

The linking between submodels is done through predetermined inputs, which for the preceeding submodel become outputs a.s.o.

This decomposition method being based upon technological was called "technological decomposition" or "horizontal decomposition".

3. AN EXEMPLE OF TECHNOLOGICAL DECOMPOSITION (HORIZONTAL)

In the following we will show a didactic dynamic model, that is of small size, which we will solve globally, then decomposie and coordinate, starting from the requirements of the modelled system.

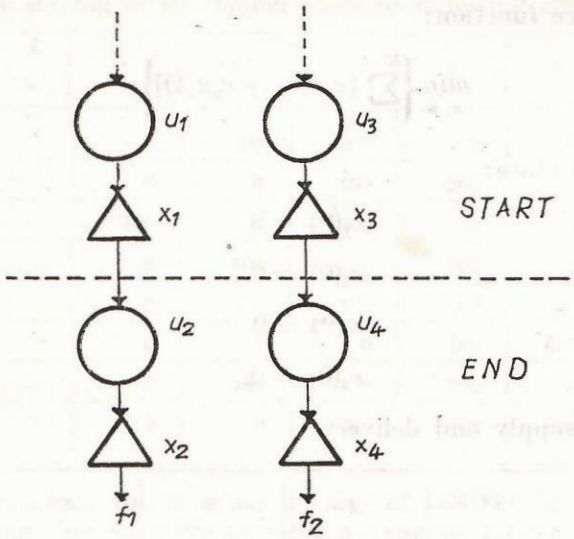

Figure 2 — Dynamic model for production management and planning

Dynamic equations:

$$x_1(k+1) = x_1(k) + u_1(k) - u_2(k)$$

$$x_2(k+1) = x_2(k) + u_2(k) - f_1(k)$$

$$x_3(k+1) = x_3(k) + u_3(k) - u_4(k)$$

$$x_4(k+1) = x_4(k) + u_4(k) - f_2(k).$$

Capacity constraints:

$$u_1(k) \leqslant d_{M1}(k)$$

$$u_2(k) + u_4(k) \leqslant d_{M3}(k)$$

$$u_3(k) \leqslant d_{M2}(k).$$

Continuity constraints for the technological process (succession constraints):

$$x_1(k) \geqslant u_2(k)$$

$$x_2(k) \geqslant f_1(k)$$

$$x_3(k) \geqslant u_4(k)$$

$$x_4(k) \geqslant f_2(k).$$

Performance function:

$$\min_{x,\ u}\left\{\sum_{K=0}^{N-1}[c_1x(k)+c_2u(k)]\right\}.$$

The initial state:

$$x_1(0)=0$$
$$x_2(0)=0$$
$$x_3(0)=0$$
$$x_4(0)=0.$$

Capacity, supply and delivery:

d, f \ k	1	2	3	4	5
M1		400	200		
M2	400		200		
M3		400	400	400	
f1					600
f2					600

Looking at the dynamic model shown above, we see that it has as outputs two finished products and that the solution is unique, therefore the solution polyhedron is pointlike.

We have solved the dynamic model using various coefficients, depending upon the choosen performance function. Thus:

a)
$$\min_{x}\left\{\sum_{k=0}^{N-1}[c_1x(k)]\right\}$$
$$c_{11}=15$$
$$c_{12}=50$$
$$c_{13}=40$$
$$c_{14}=75.$$

The solving of the example shown in figure 2 was accomplished on a FELIX C-256 computer by means of the PROGEN package (a Romanian extension of the OPALINE package).

Aggregate solving of the model leads to following results:

x, u \ k	1	2	3	4	5	6
u_1	0	400	200	0	0	
u_2	0	0	400	200	0	
u_3	400	0	200	0	0	
u_4	0	400	0	200	0	
x_1	0	0	400	200	0	0
x_2	0	0	0	400	600	0
x_3	0	400	0	200	0	0
x_4^+	0	0	400	400	600	0

The performance function has a value of 188,000. M. U.*)

Considering that we have to solve a dynamic model of large size, we have decomposed the dynamic model in figure 2 from the point of view of the place where the technological process develops, i.e. by production departments. As stated above, decomposition can be made by departments, work shops, homogeneous equipment groups etc.

Thus we obtain:

— the submodel END

— the submodel START.

The submodel END will be described as follows:

Dynamic equations:

$$x_2(k+1) = x_2(k) + u_2(k) - f_1(k)$$
$$x_4(k+1) = x_4(k) + u_4(k) - f_2(k).$$

Capacity constraints:

$$u_2(k) + u_4(k) \leqslant d_{M3}(k)$$

Continuity constraints for the technological process (succession constraints):

$$x_2(k) \geqslant f_1(k)$$
$$x_4(k) \geqslant f_2(k).$$

Performance function:

$$\min_{x,\,u} \left\{ \sum_{k=0}^{N-1} [c_1 x(k) + c_2 u(k)] \right\}.$$

The initial state:

$$x_2(0) = 0$$
$$x_4(0) = 0.$$

*) Monetary Unities

Capacity, supply and delivery:

d, f \ k	1	2	3	4	5
$M3$		400	400	400	
f_1					600
f_2					600

The submodel START will be described as follows:
Dynamic equations:

$$x_1(k+1) = x_1(k) + u_1(k) - u_2(k)$$
$$x_3(k+1) = x_3(k) + u_3(k) - u_4(k).$$

Capacity constraints:

$$u_1(k) \leqslant d_{M1}(k)$$
$$u_3(k) \leqslant d_{M2}(k).$$

Continuity constraints for the technological process (succession constraints):

$$x_1(k) \geqslant u_2(k)$$
$$x_3(k) \geqslant u_4(k).$$

Performance function:

$$\min_{x,\,u} \left\{ \sum_{k=0}^{N-1} [c_1 x(k) + c_2 u(k)] \right\}.$$

The initial state:

$$x_1(0) = 0$$
$$x_3(0) = 0.$$

Capacity, supply and delivery:

d, f \ k	1	2	3	4	5
$M1$		400	200		
$M2$	400		200		
f_3		400	200		
f_4			200	400	

Deliveries f_3 and f_4 from the submodel START appear through change of u_2 and u_4:

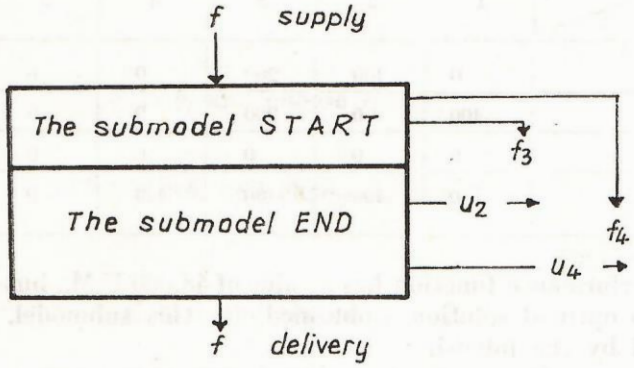

Fig. 3 Hierarchical dynamic model for production management and planning

For submodel END we had in view the performance function:

$$\min_x \left\{ \sum_{k=0}^{N-1} [c_1 x(k)] \right\}$$

$$c_{12} = 50$$

$$c_{14} = 75$$

which leads to following results:

x, u \ k	1	2	3	4	5	6
u_2	0	400	200			
u_4	0	0	200	400		
x_2	0	0	400	600	600	0
x_4	0	0	0	200	600	0

The performance function has a value of 140,000. M. U.

For the submodel START we had in view the performance function:

$$\min_x \left\{ \sum_{k=0}^{N-1} [c_1 x(k)] \right\}$$

$$c_{11} = 15$$

$$c_{13} = 40$$

which leads to following results:

x, u \ k	1	2	3	4	5	6
U_1	0	400	200	0	0	
U_3	400	0	200	0	0	
x_1	0	0	0	0	0	0
x_3	0	400	400	400	0	0

The performance function has a value of 48,000 U.M., but we must mention that no optimal solution is obtained, for this submodel. This situation is generated by the following:

— the system has an unique optimal solution;

— the method used for decomposition allows to obtain only suboptimal solutions for the entire system and optimal solutions for each subsystem.

To obtain a solution also for submodel START, we must introduce tolerances [2]:

$$[b_i,\ B_i] \quad B_i > b_i \quad \text{for } i = \overline{1, m}$$

$$[B_i,\ b_i] \quad B_i < b_i \quad \text{for } i = \overline{m+1, p}.$$

The following system is obtained:

$$\begin{cases} \sum_{j=1}^{n} a_{ij} x_j \leqslant b_i \quad \text{for } i = \overline{1, m} \\ \sum_{\substack{j=1 \\ x_j \geqslant 0}}^{n} a_{ij} x_j \geqslant b_i \quad \text{for } i = \overline{m+1, p}. \end{cases}$$

which leads to the vector

$$\begin{bmatrix} c_1 \\ \vdots \\ c_m \\ c_{m+1} \\ \vdots \\ c_p \end{bmatrix}$$

which has to meet the conditions:

— the system $b_i \leftrightarrow c_i$ must be compatible;

— c_i must be as close as possible to b_i.

Solving we obtain:

$$c_i^* \leqslant b_i \text{ and choose } c_i = b_i$$

for $i = \overline{1, m}$

$$c_i^* > b_i \text{ and choose } c_i = c_i^*$$

and

$$c_i^* \geqslant b_i \text{ and choose } c_i = b_i$$

for $i = \overline{m+1, p}$

$$c_i^* < b_i \text{ and choose } c_i = c_i^*.$$

It can be demonstrated that there is an equivalence between:

$$\begin{cases} f_i(x) = \dfrac{B_i - \sum\limits_{j=1}^{n} a_{ij}x_j}{B_i - b_i} \text{ for } i = \overline{1, m} \\ f_i(x) = \dfrac{\sum\limits_{j=1}^{n} a_{ij}x_j - B_i}{b_i - B_i} \text{ for } i = \overline{m+1, p} \\ \max\limits_{x_i \geqslant 0} \quad \min\limits_{i=\overline{1, p}} \quad f_i(x) \end{cases}$$

and

$$\begin{cases} x_{n+1} = max \\ x_{n+1} - x_{n+2} \leqslant f_i(x) \quad \text{for } i = \overline{1, p}. \end{cases}$$

We notice that adding up the performance functions of the two sub-models, the sum is equal to that obtained for the aggregate model. Applying tolerances for production planning will lead to a violation of constraints (capacity, productivity, materials, specific consumption etc.) and therefore to a "dilution" of the global solution.

b) A similar result is obtained when we have the following situation:

$$\min_{x,\, u} \left\{ \sum_{k=0}^{N-1} [c_1 x(k) + c_2 u(k)] \right\}$$

$$\begin{aligned} c_{11} &= 15 & c_{21} &= 10 \\ c_{12} &= 50 & c_{22} &= 35 \\ c_{13} &= 40 & c_{23} &= 5 \\ c_{14} &= 75 & c_{24} &= 35. \end{aligned}$$

Solving the aggregate model leads to an optimal solution, the obtained value for the performance function being 239,000 M.U.

Solving the submodels we obtain for END an optimal solution, with a value for the performance function of 182,000 M.U.; for START the obtained value for the performance function is 57,000 M.U. Also in this case it is necessary to apply tolerances in production planning and programming.

c) Not such a good result as for a) and b) is obtained if:

$$\min_{x,\,u}\left\{\sum_{k=0}^{N-1}[c_1x(k) + c_2u(k)]\right\}$$

$$c_{11} = 35 \qquad c_{21} = 10$$

$$c_{12} = 50 \qquad c_{22} = 15$$

$$c_{13} = 40 \qquad c_{23} = 5$$

$$c_{14} = 75 \qquad c_{24} = 35.$$

The sum of the performance functions for the two submodels (170,000 + 57,000 = 227,000 M.U.) is smaller than that obtained solving the aggregate model (239,000 M.U.).

d) Solving the model for the case:

$$\min_{u}\left\{\sum_{k=0}^{N-1}[c_2u(k)]\right\}$$

$$c_{21} = 10$$

$$c_{22} = 35$$

$$c_{23} = 5$$

$$c_{24} = 35$$

we obtain equal values for the total value of the performance functions for the two submodels (42,000 + 9,000 = 51,000 M.U.) and for the aggregate model 51,000 U.M.

e) A similar situation is obtained also for:

$$\min_{u}\left\{\sum_{k=0}^{N-1}[c_2u(k)]\right\}$$

$$c_{21} = 10$$

$$c_{22} = 15$$

$$c_{23} = 5$$

$$c_{24} = 35$$

where the value of the performance function for the aggregate model (39,000 M.U.) is equal to the total performance function of the submodels (30,000 + 9,000 = 39,000 M.U.)

From the models presented from a) to e) it results that having to solve a large size problem, which has only one solution, that is the optimal solution, by applying horizontal decomposition, we will find that one or more submodels have no solution, and that for the latter it is necessary to apply tolerances. Application of tolerances to these submodels will influence negatively the global optimum, the solution offered being suboptimal.

In situations in which we have a large size problem, where besides the optimal solution, there exists also suboptimal solutions (therefore the problem has several solutions), each solved subproblems will have an optimal solution, but the sum of performance functions of the submodels will lead to a suboptimal solution (for the entire problem).

4· THE VERTICAL DECOMPOSITION METHOD

Refering still to the model presented in figure 2, we mention that another interesting means of solving large size problems is to consider such a model as a graph.

In solving the problem we start from the known outputs of the system. First the longest branch of the graph is solved (considering branches as verticals — hence the name "vertical decomposition"). Thus the longest branch of graph is solved. In the situation in which there are several such branches (equally long), starting from the known outputs of the system, we chose that branch which has the gratest number of ties (nodes) with the rest of the system. When these are several equal branches (with the same number of ties), and all have outputs, we will solve the first, then the secound a.s.o until the whole system is solved.

In this way optimal solution for each submodel can be obtained solutions which summed up are equal to the aggregate optimal solution.

5. CONCLUSIONS

In solving large size systems it is necessary to decompose and co-ordinate, because only in this way can be solved the problems which confront production planning and programming in large plants. We consider that the way in which penalty coefficients in performance functions are ascentained, see horizontal decomposition method is of great importance and that this may lead to a situation in which the sum of partial optima for the submodels is equal to the aggregate optimum (the importance of penalty coefficients for the performance function can be noticed in 3 a), b), d) and e)).

We consider that solving large size system through decomposition and co-ordonation must have in view the obtention of an aggregate optimum, because only this would allow to obtain a high degree of economic efficiency. The horizontal decomposition method allows to solve large scale problems, but it leads to optimal solutions only for each subproblem, and only to a suboptimal solution for the entire problem.

REFERENCES

1. CONSTANTINESCU P., NEGOIȚĂ C. V.: *Informational Systems as Models of Managerial and Managed Systems.* Editura Tehnică, Bucharest, 1975.
2. NEGOIȚĂ C. V., SULARIA M.: *Fuzzy Programming and Tolerances in Planning.* Studii și cercetări de calcul economic și cibernetică economică, no. 1, 1976.
3. NEGOIȚĂ C. V.: *Automatic Control Theory with Applications to Industry.* Curs A.S.E., Facultatea de Cibernetică Economică, 1973.
4. FOMIN P.: *Modelling and Management of Production Processes.* Paper presented at the second Symposium of Informatics and Management, Cluj-Napoca, March, 1975.
5. ANDREI N. C., FULGA I. ST. et al.: *Production Management Model.* Paper presented at the second session of scientific-technical communications Automatic and Information Systems in Industry, Institutul Politehnic București, București, 27—29 April, 1977.
6. STĂNCIULESCU F., AVRAMIȚĂ G.: *Experimental Results Concerning an Algorithm for Control of Large-Systems by Decomposition and Dynamic Co-ordination.* Paper presented at the Symposium of Romanian Academy, December, 1976.
7. STĂNCIULESCU F., AVRAMIȚĂ G.: *Models and Algorithms for Hierarchical Management of Production Processes.* Paper presented at the III-rd Symposium for Informatics and Management, 11—14 May, 1977, Cluj-Napoca.

DECOMPOSITION ET RESOLUTION DES PROBLEMES DE GRANDES DIMENSIONS DANS LA PROGRAMMATION DE LA PRODUCTION

On présente une solution des problèmes de grandes dimensions, ne pouvant pas être résolus à l'aide des configurations de calcul et des paquets de programmes actuels. La résolution se fait par décomposition des problèmes de grandes dimensions en sous-problèmes de dimensions réduites, coordinés par leur résolution séquentielle, en fonction du but final.

En tant qu'illustration on présente un modèle dynamique de programmation de la production, décomposé en sous-modèles.

РАЗЛОЖЕНИЕ И РАЗРЕШЕНИЕ ЗАДАЧ БОЛЬШИХ РАЗМЕРОВ В ПЛАНИРОВАНИИ ПРОИЗВОДСТВА

В настоящей работе представлен способ разрешения задач больших размеров, которые не могут быть решены при существующих ЭВМ и программах для них. Разрешение этих задач осуществляется при разложении задач больших размеров в подзадачи меньших размеров, скоординированных при их последовательном разрешении, в зависимости от конечной цели.

В качестве примера приведена динамическая модель программирования производства, разложенная в подмодели.

10. XI. 1977