

K.II 1130 III 16

# ALGORYTMY

VOL. III • NO. 6 • 1966



INSTYTUT MASZYN MATEMATYCZNYCH

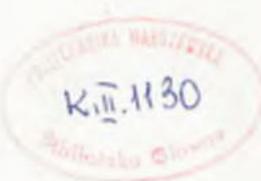
A L G O R Y T M Y  
Vol. III N°6 1966



P R A C E

Instytutu Maszyn Matematycznych

Copyright © 1966 - by Instytut Maszyn Matematycznych, Warszawa  
Poland  
Wszelkie prawa zastrzeżone



K o m i t e t   R e d a k c y j n y

Leon ŁUKASZEWICZ /redaktor/, Antoni MAZURKIEWICZ,  
Tomasz PIETRZYKOWSKI /z-ca redaktora/, Dorota PRAWDZIC,  
Zdzisław WRZESZCZ.

Redaktor działowy: Krzysztof MOSZYŃSKI.  
Sekretarz Redakcji: Witold WUDEL

Adres redakcji: Warszawa, ul.Koszykowa 79, tel.28-37-29

T R E S C  
C O N T E N T S

Metody numeryczne równań różniczkowych  
Numerical methods of differential equations

T. Piwecki, Z. Sokólski	EXPERIMENTAL DETERMINATION OF OVERRELAXATION COEFFICIENTS FOR THE LAPLACE EQUATION IN CYLINDER COORDINATES WITH MIXED BOUNDARY CONDITIONS . . . . .	7
T. Desperat, T.W. Desperat, A. Kiełbasiński	SOME REMARKS ON THE CHOICE OF FACTORS FOR ALTERNATING DIRECTION IMPLICIT METHOD . . . . .	17
J. Olszewski	AN APPLICATION OF NUMERICAL METHODS TO EXAMINE THE PRACTICAL STABILITY OF SOLUTIONS OF ORDINARY DIFFERENTIAL EQUATIONS . . . . .	25
F. Labisch	APPROXIMATION OF THE SOLUTION OF A UNIFORM SHELL DEFORMATION CAUSED BY THE DETONATION OF THE EXPLOSIVE CHARGE OF FINITE THICKNESS	37

Metody optymalizacji  
Methods of optimality

J. Janowski, J. Winkowski	DESCRIPTION OF THE PROGRAM OF CRITICAL PATH'S DETERMINATION . . . . .	55
K. Moszyński, W. Ostalski	OPTYMALIZACJA CIĘCIA TAŚMY . . . . .	71
J. Szmelter, B. Gadek	ROZWIĄZANIE ZADANIA TRANSPORTOWEGO NA MASYNIE MATEMATYCZNEJ . . . . .	83

**Metody statystyczne**  
**Statistical methods**

- R. Zieliński  
GENERATOR LICZB LOSOWYCH O ROZKŁADZIE RÓWNOMIERNYM DLA MASZYNY ZAM-2 . . . . . 103
- R. Zieliński  
STOCHASTYCZNE ALGORYTMY W ZAGADNIENIACH  
OPTYMIZACJI . . . . . 127

CONFERENCE ON THE NUMERICAL SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS  
WITH SPECIAL REFERENCE TO STIFF PROBLEMS

EDWARD L. WATSON  
JOHN D. COLEMAN

Mathematics Department, MSRI, 1979

CONTENTS  
INTRODUCTION  
NUMERICAL METHODS FOR ORDINARY DIFFERENTIAL EQUATIONS  
STIFF PROBLEMS

THEORY AND PRACTICE  
NUMERICAL METHODS FOR STIFF PROBLEMS  
THEORY AND PRACTICE  
NUMERICAL METHODS FOR STIFF PROBLEMS  
THEORY AND PRACTICE

NUMERICAL METHODS FOR ORDINARY DIFFERENTIAL EQUATIONS  
STIFF PROBLEMS  
THEORY AND PRACTICE  
NUMERICAL METHODS FOR STIFF PROBLEMS  
THEORY AND PRACTICE

NUMERICAL METHODS  
OF  
DIFFERENTIAL EQUATIONS



EXPERIMENTAL DETERMINATION OF OVERRELAXATION COEFFICIENTS FOR THE LAPLACE EQUATION IN CYLINDER COORDINATES WITH MIXED BOUNDARY CONDITIONS

by Teofil PIWECKI  
Zdzisław SOKÓLSKI

Received November 13th, 1965

The authors discuss an improvement of the relaxation method for the solution of Laplace's equation /1/ with boundary conditions /1a/. The residuals were successively removed by an overrelaxation process.

The values "p" of corrections were obtained from  $p = F_0/H_i$ ;  $F_0$  being the residual,  $H_i$  - the overrelaxation coefficient. It was shown that the use of three different coefficients:  $H_1$  - for the nodes on the axis of symmetry,  $H_2$  - for the upper boundary, and  $H$  - for all other nodes is advantageous.

The paper presents experimental results of investigations on the possibility of shortening the computation time needed to obtain the solution of Laplace's equation with given boundary conditions by the relaxation method.

By applying the overrelaxation process the time for obtaining the solution is significantly shortened. Overrelaxation consists in applying greater corrections at separate nodes than it is needed to change the residual at the node to zero. Some investigations on the determination of the so-called overrelaxation coefficients are given below.

Examinations \*) have been made using as an example Laplace's equation in a cylindrical region with a rotational symme-

\*) Examinations have been made in April, 1964 by the authors, at the Computing Centre P.G. using the ZAM-2 Beta digital computer.

try in relation to the axis of cylinder. The equation in cylindrical coordinates is, in this case, written

$$\frac{\partial^2 \varphi}{\partial r^2} + \frac{1}{r} \frac{\partial \varphi}{\partial r} + \frac{\partial^2 \varphi}{\partial z^2} = 0$$

/1/

with boundary conditions /fig. 1/

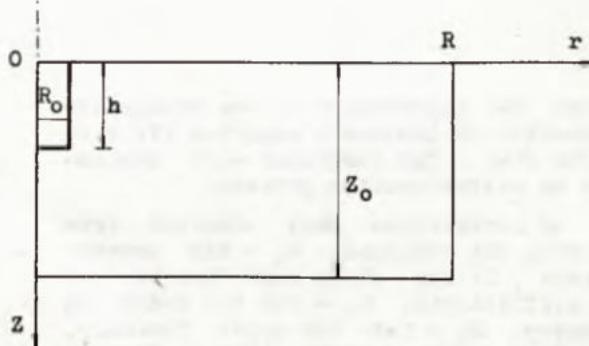


Fig. 1.

$\varphi = \text{const}_1$  for  $r = R_o$   $0 \leq z \leq h$  and for  $z = h$   $0 \leq r \leq R_o$

$\varphi = \text{const}_2$  for  $r = R$   $0 \leq z \leq z_o$

$\frac{\partial \varphi}{\partial n} = 0$  for  $z = 0$  and  $R_o < r < R$ ; for  $z = z_o$  /1a/  
and  $0 < r < R$  and for  $r = 0$  and  $h < z \leq z_o$ .

This equation describes the flow of water to an incomplete artesian well in horizontal aquifer.

Under given boundary conditions the below difference equations correspond to equation /1/:

- for internal region:

$$\varphi_1 + \varphi_3 + \frac{2n+1}{2n} \varphi_2 + \frac{2n-1}{2n} \varphi_4 - 4 \varphi_0 = 0, \quad /2a/$$

- for upper boundary:

$$2\varphi_3 + \frac{2n+1}{2n}\varphi_2 + \frac{2n-1}{2n}\varphi_4 - 4\varphi_0 = 0 , \quad /2b/$$

- for lower boundary:

$$2\varphi_1 + \frac{2n+1}{2n}\varphi_2 + \frac{2n-1}{2n}\varphi_4 - 4\varphi_0 = 0 , \quad /2c/$$

- for axis of symmetry ( $r = 0$ ):

$$\varphi_1 + \varphi_3 + 4\varphi_2 - 6\varphi_0 = 0 . \quad /2d/$$

In the above formulae  $\varphi_0, \varphi_1, \varphi_2, \varphi_3, \varphi_4$ , denote the values of the function  $\varphi$  in points 0, 1, 2, 3, 4,  $n = r_0/h$  respectively, where  $r_0$  determines the abscissa of the central point "0" /fig. 2/,  $h$  is the mesh size.

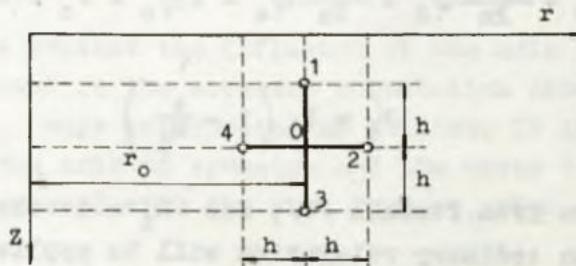


Fig. 2.

As it is well known the relaxation method consists - in accepting the values of the function  $\varphi$  at all nodes of the integration region: on the boundary known - according to boundary conditions, and arbitrary at the remaining nodes, and in a correction of the latter which lasts until equations /2/ are satisfied with the required accuracy in the entire region.

As the assumed values of the function  $\varphi$  generally do not satisfy equations /2/, their right hand sides do not equal zero but a

certain residual  $F_0$ . The relaxation consists just in reducing residuals  $F_0$  by means of adding an appropriate correction  $p$  to the value of the function  $\varphi_0$ .

We determine the correction  $p$  as the quotient

$$p = \frac{F_0}{H_1} , \quad /3/$$

where for the internal nodes  $F_0 = \varphi_1 + \varphi_3 + \frac{2n+1}{2n}\varphi_2 + \frac{2n-1}{2n}\varphi_4 - 4\varphi_0$ , and  $H_1$  defines a certain coefficient. After adding the correction  $p$  to  $\varphi_0$  one obtains the corrected value of the function

$$\varphi'_0 = \varphi_0 + p$$

and the new residual in the central point will be equal to

$$F'_0 = \varphi_1 + \varphi_3 + \frac{2n+1}{2n}\varphi_2 + \frac{2n-1}{2n}\varphi_4 - 4\varphi'_0 = F_0 - 4\frac{F_0}{H_1} = F_0 \left(1 - \frac{4}{H_1}\right)$$

$$F'_0 = F_0 \left(1 - \frac{4}{H_1}\right) . \quad /4/$$

As is seen from formula /4/, for  $H_1 = 4$  the residual will be zero, thus an ordinary relaxation will be applied. For  $H_1 < 4$  the correction will be greater than it is needed to change the residual to zero, thus overrelaxation will be applied. For  $H_1 > 4$  the correction will be smaller than it is needed to change the residual to zero, thus underrelaxation will be applied.

This paper aims at the determination of such values  $H_1$  that are necessary to obtain the shortest time needed for the solution.

While investigating, it appeared that under the given boundary conditions, the speed of the convergence of corrected values of the function to the proper solution was various at different points of the region. This process was the slowest on the axis of symmetry and on the upper bound. It suggests the idea of applying dif-

ferent coefficients  $H_1$  and  $H_2$  for points on the axis of symmetry and on the upper bound. It was accepted  $H_0 = H$  for the internal region  $H_1 = \gamma_1 \cdot H$  for the axis of symmetry, and  $H_2 = \gamma_2 \cdot H$ , for the upper bound.

It is easy to notice that on the axis of symmetry in the case of ordinary relaxation  $\gamma_1 = 3/2$ . Indeed, the equivalent of formula /4/ for the axis of symmetry takes the form

$$F'_0 = F_0 \left( 1 - \frac{3}{H_1} \right), \quad /5/$$

thus, after substituting  $H_1 = 3/2 H_0$  one obtains formula /4/ and the formula for the correction

$$p = \frac{F_0}{H_1}. \quad /6/$$

Similarly it can be stated that in the case of ordinary relaxation  $H_2 = H (\gamma_2 = 1)$ .

In order to make evident the influence of the axis of symmetry and of the upper bound in the accepted computation scheme, coefficients  $\gamma_1$  and  $\gamma_2$  were interpreted as follows. If in the entire region, including the axis of symmetry and the upper bound, overrelaxation should proceed with the same "power", then

$$H_1 = \frac{3}{2} H \quad H_2 = H,$$

where:  $H$  is the overrelaxation coefficient for the remaining integration region. If, however, on the axis of symmetry and on the upper bound the overrelaxation to be used is of a different "power", then  $H_1 = 3/2\alpha_1 H$ ;  $H_2 = \gamma_2 H$  and  $\alpha_1 \neq 1$  and  $\gamma_2 \neq 1$ .

Fig. 3 shows the integration region which is here accepted as the basis to determine the optimal values of overrelaxation coefficients for equation /1/. The region has been covered with a quadratic network consisting of 10 columns and 9 rows. There were 77 network nodes in which the value of the function  $\varphi$  had been searched.

The accepted region was not very large because a lot of examples were to be computed. The correction of function  $\varphi$  in the network nodes was carried out through columns, starting with the zero column from the top downward /see figure 3/. The same initial values of the function were accepted in all network nodes, as shown in fig. 3. The number of rotations around the entire region, needed to obtain the solution with the required accuracy  $\varepsilon$ , has been accepted as basic estimation of the efficacy of overrelaxation coefficients.

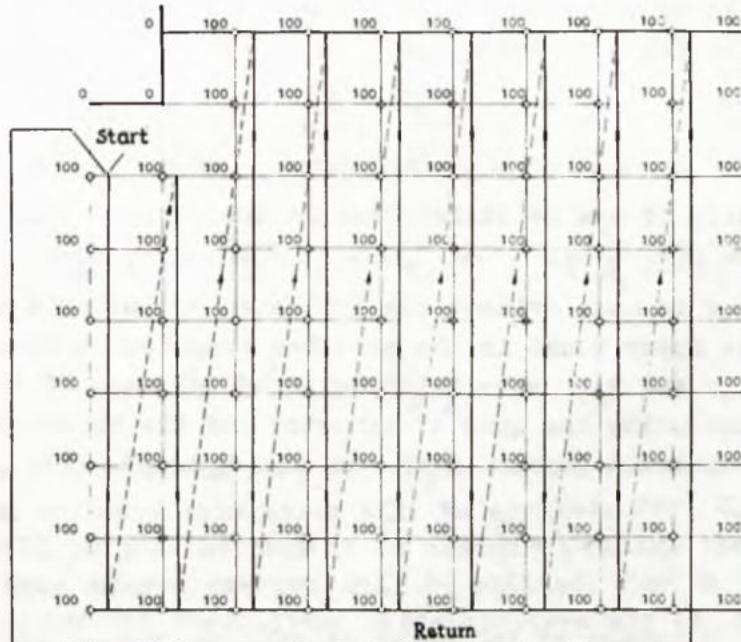


Fig. 3.

Besides, for each case, the computation time and the number of points /nodes/ in which the function was corrected /i.e.  $|F_0| > \varepsilon$ / and uncorrected /i.e.  $|F_0| \leq \varepsilon$ /, were determined.

The results are presented in diagrams in figs 4, 5, 6. The number of rotations LOB has been put in on the ordinate axis and the appropriate overrelaxation coefficients  $H$  /fig. 4/,  $\alpha_1$  /fig. 5/ and  $\eta_2$  /fig. 6/ - on the abscissa axis.

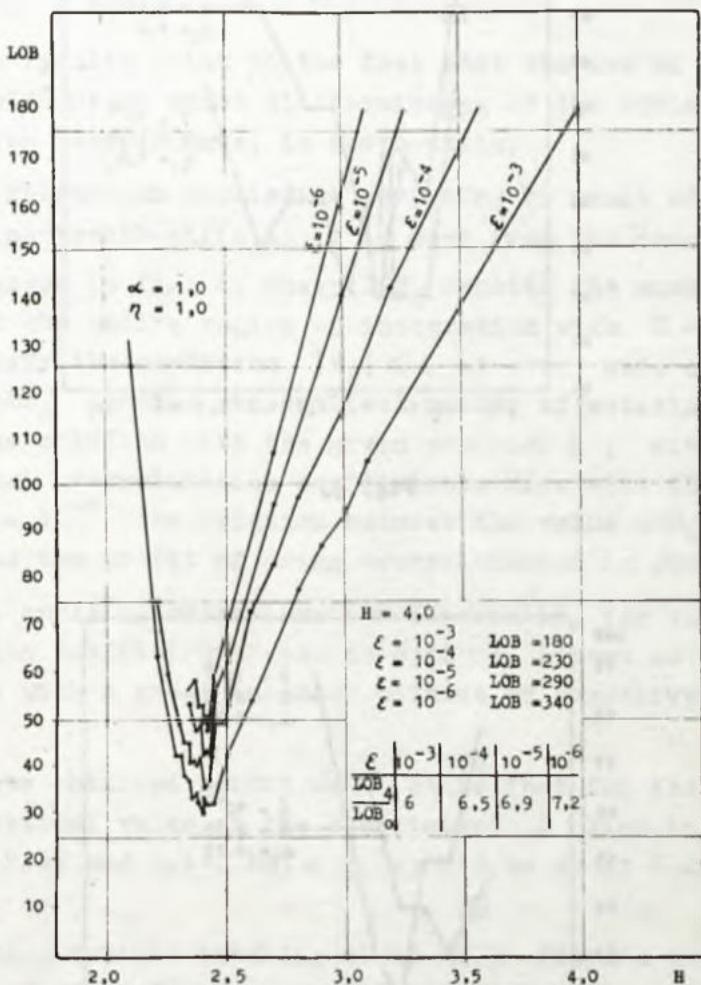


Fig. 4.

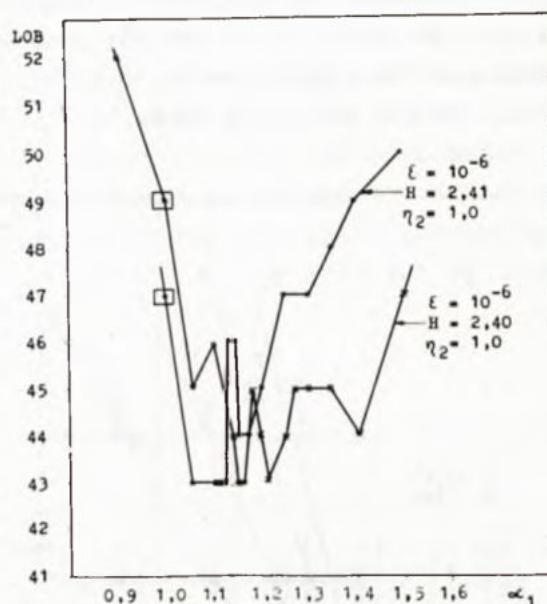


Fig. 5.

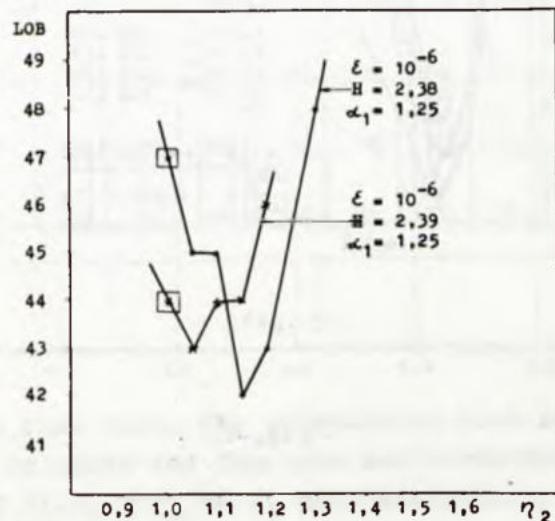


Fig. 6.

It results from diagrams that the application of various overrelaxation coefficients on the axis of symmetry and on the upper bound and inside the region permits an additional shortening of the computation time as compared with the time needed to obtain the solution when applying an equal overrelaxation coefficient in the whole region.

The above results point to the fact that the use of overrelaxation, even without an exact distinguishing of the optimal value of overrelaxation coefficients, is worth-while.

Ordinary relaxation applied to computing by means of a digital computer is not worth-while as it is seen from the compendium  $LOB_4/LOB_0$  given in fig. 4, where  $LOB_4$  denotes the number of rotations around the entire region of integration with  $H = 4$ , necessary to satisfy the condition  $|F_0| \leq \varepsilon$  at every node of the network, and  $LOB_0$  denotes the smallest number of rotations needed to obtain the solution with the given accuracy  $\varepsilon$ , with the use of the optimal overrelaxation coefficient. E.g. with the assumed accuracy  $\varepsilon = 10^{-6}$  the relation between the value  $LOB_4$  and  $LOB_0$  is 7.2 - thus the profit of using overrelaxation is obvious.

Moreover, applying overrelaxation especially, for values of the overrelaxation coefficients near to optimal, allows us to carry out computations with a great accuracy without an excessive waste of time.

The results obtained permit us to state that for the considered region the optimal value of the coefficient  $H$  lies in the interval between 2.37 and 2.41, but  $\alpha_1$  should be about 1.25, and  $\gamma_2$  - about 1.15.

Undoubtedly, further investigations will afford a more exact determination of optimal values of all three coefficients, and especially permit us to state if the given values are also optimal for regions of integration with other proportions and dimensions.

Data gained during the execution of the presented investigations show that the process of relaxation itself may be still more accelerated if realized in a digital computer by means of applying changes of the values of coefficients  $H, \alpha_1, \gamma_2$  in time.

References

- [1] ALLEN; D.N. De G.: Relaxation methods, New York 1954.
- [2] COLLATZ, L.: Metody numeryczne rozwiązywania równań różniczkowych, PWN, Warszawa 1960.
- [3] BECKENBACH, E.F.: Nowoczesna matematyka dla inżynierów, PWN, Warszawa 1962.

SOME REMARKS ON THE CHOICE OF FACTORS  
FOR ALTERNATING DIRECTION IMPLICIT  
METHOD

by Teresa DESPERAT  
Tadeusz Wiesław DESPERAT  
Andrzej KIELBASIŃSKI

Received April 2, 1966

The paper deals with practical aspects of a choice of the alternating direction method parameters for Poisson's equation on a rectangle. A modification of Peaceman-Rachford algorithm is proposed, doubling a convergence speed. Experimental results illustrate discussed problems.

The paper [1] contains a description of the iterative method for solving a set of difference equations which approximate a mixed boundary value problem for Laplace's equation on a square. The basic concept of this method is to choose a set of real parameters  $\alpha_1, \alpha_2, \dots, \alpha_p$ , for which the rational function

$$\omega(\lambda) = \prod_{i=1}^p \frac{1 - \alpha_i \lambda}{1 + \alpha_i \lambda} \quad /1/$$

gives a good approximation to zero in the interval  $[\varepsilon, 1 - \varepsilon]$ , for some  $\varepsilon \in (0, 0.5)$ . We shall call integer  $p$  the degree of the function  $\omega(\lambda)$ .

Authors [1] propose the following algorithm:

let  $\varphi$  be any number from  $(0, 1)$ .

We choose the first parameter  $\alpha_1$  so that the function

$$\omega_1(\lambda) = \frac{1 - \alpha_1 \lambda}{1 + \alpha_1 \lambda}$$

takes the value  $\varphi$  at the point  $\lambda_0 = \varepsilon$ .

Then we determine the point  $\lambda = \lambda_1$ , at which the function  $\omega_1(\lambda)$  achieves the value  $-\varphi$ .

We find  $\alpha_2$  so that the function

$$\omega_2(\lambda) = \frac{1 - \alpha_2 \lambda}{1 + \alpha_2 \lambda}$$

takes the value  $\varphi$  at the point  $\lambda_1$ , and so on.

We continue this process till  $\lambda_j < 1 - \varepsilon$ .

It is easily seen that

$$\alpha_1 = \frac{1 - \varphi}{1 + \varphi} \frac{1}{\varepsilon} \quad /2/$$

$$\alpha_k = \left( \frac{1 - \varphi}{1 + \varphi} \right)^2 \alpha_{k-1} \quad k = 2, 3, \dots$$

and  $p = k$  when

$$\alpha_k \leq \frac{1 + \varphi}{1 - \varphi} \frac{1}{1 - \varepsilon} \quad *$$

The function  $\omega(\lambda)$  thus obtained approximates zero in the interval  $[\varepsilon, 1 - \varepsilon]$  with an error not exceeding  $\varphi$  /in the sense of the uniform approximation/.

The above described construction shows /see [1] /, that, in the case of a mesh with  $N^2$  nodal points, the number of arithmetic operations, necessary in the ADI method for error reduction by a definite factor, is proportional to  $N^2 \log N$ , while the optimal overrelaxation methods /D. Young, S. Frankel, J. von Neumann/ require a number of operations proportional to  $N^3$ . Due to this fact there is considerable interest in the ADI method.

According to D. Young /see [2] p. 281/ one should take  $\varphi = \sqrt{2} - 1$  and repeat the complete step of ADI method until the required accuracy of solution is obtained. This idea was applied in practical procedures /see [3] p. 55/. Meanwhile it is easily seen, that the optimal value  $\varphi$  found by D. Young, optimizes only the bound of

the error, but not the error itself. It is sufficient to tabulate densely enough the function constructed according to the rule of Peaceman and Rachford (2), in order to check that the optimal value  $\rho$  differs considerably from  $\sqrt{2} - 1$ .

To illustrate this, suppose that  $N = 20$ ,  $\varepsilon = \sin^2 \frac{\pi}{2(N+1)}$ ,  $\rho = \sqrt{2}-1$ . Then the average convergence rate is 0.277, whereas for  $\rho = 0.1$  this rate is 0.444. More detailed observation of the values of function (1), constructed according to (2) and tabulated in the interval  $[\varepsilon, 1-\varepsilon]$ , shows that the function (1) takes a value of the same order of magnitude as  $\rho$  at the point  $\lambda = \varepsilon$  and sometimes at the end point  $\lambda = 1-\varepsilon$ , while the values of this function for inner points of the interval are of the order  $\rho^2$ . This fact also may be easily deduced from the construction of the function (1), (2).

We now attempt some modification of the process (2).

We choose  $\alpha_1$  so that the function

$$\omega_1(\lambda) = \frac{1 - \alpha_1 \lambda}{1 + \alpha_1 \lambda}$$

takes the value  $\rho^2$  at the point  $\lambda_0 = \varepsilon$ , thus

$$\alpha_1 = \frac{1 - \rho^2}{1 + \rho^2} \frac{1}{\varepsilon} \quad /3.1/$$

We take just such a value of the last parameter  $\alpha_p$ , that the function

$$\omega_p(\lambda) = \frac{1 - \alpha_p \lambda}{1 + \alpha_p \lambda}$$

takes the value  $-\rho^2$  at the point  $1 - \varepsilon$ . We therefore require

$$\alpha_p = \alpha^* = \frac{1 + \rho^2}{1 - \rho^2} \frac{1}{1 - \varepsilon} \quad /3.2/$$

The values  $\alpha_k /k = 2, 3, \dots, p-1/$  are computed in the same way as in algorithm (2):

$$\alpha_k = \left( \frac{1 - \rho}{1 + \rho} \right)^2 \alpha_{k-1} \quad /3.3/$$

till the inequality  $\alpha_k > \alpha^*$  holds. /In this way we determine the value  $p/$ .

An experimental comparison was made of the function constructed according to the modified algorithm (3) with the functions constructed according to the primary algorithm (2). The results confirm our hypothesis, that it is possible to obtain in this way the function, which approximates zero in  $[\varepsilon, 1-\varepsilon]$  with accuracy of the order  $\rho^2$ , and has almost the same degree  $p$  as a function constructed in algorithm (2).

An advantage of the proposed algorithm is that it doubles the convergence speed of the method.

It is easily seen that we obtain the best bound for the error when the equality

$$\left( \frac{1 - \rho}{1 + \rho} \right)^2 \alpha_{p-1} = \alpha^* \quad /4/$$

holds, i.e. when the value of the last parameter  $\alpha_p$  is not "artificially" imposed, but belongs to the recursively defined sequence  $\{\alpha_k\}$ . Hence, a construction described in algorithm (3) can be slightly improved, not by choosing the degree  $p$  of the function  $\omega(\lambda)$  for given  $\varepsilon$  and  $p$ , but rather by determining the value  $\rho$  for given  $\varepsilon$  and  $p$  in such a way that the last factor of the rational function  $\omega(\lambda)$ :

$$\frac{1 - \alpha_p \lambda}{1 + \alpha_p \lambda}$$

takes the value  $-\rho^2$  at the point  $\lambda = 1-\varepsilon$ . In order to find we obtain from (3) and (4) the following equation:

$$\frac{1+\rho^2}{1-\rho^2} \frac{1-\epsilon}{1-\rho} = \frac{1-\rho^2}{1+\rho^2} \left( \frac{1-\rho}{1+\rho} \right)^{2(p-1)},$$

where  $\epsilon$  and  $p$  are given numbers. Denoting  $\epsilon/(1-\epsilon)$  by  $z$  we have:

$$\varphi(\rho) = \left( \frac{1-\rho}{1+\rho} \right)^{2p-1} \left( \frac{1-\rho^2}{1+\rho^2} \right)^2 - z = 0. \quad /5/$$

It is easily seen that the root  $\rho^*$  of this equation lies between  $\rho = \rho_1$  and  $\rho = \rho_2$ , where  $\rho_1$  is a root of the equation

$$\left( \frac{1-\rho}{1+\rho} \right)^{2p} = z$$

and  $\rho_2$  is a root of the equation

$$\left( \frac{1-\rho}{1+\rho} \right)^{2(p-1)} = z.$$

Now  $\rho_1 = \frac{1-u}{1+u}$  where  $u = z^{2p}$  and  $\rho_2 = \frac{1-v}{1+v}$  where  $v = z^{2(p-1)}$ . In order to evaluate numerically  $\rho^*$ , a bisection method or any other method can be used.

The choice of the optimal value  $\rho = \rho^*$  for given  $\epsilon$  and  $p$  being known, the following question may be put: for which  $p$  does the process (3) give the most rapid error reequation? This problem is not solved in a theoretical way. We obtain the following experimental results:

Table 1.

N	$p_{opt}$	$\rho^*_{opt}$	convergence rate
10	6	0.185	0.756
20	7	0.206	0.589
30	8	0.204	0.522
45	8	0.230	0.467
55	9	0.214	0.444
65	10	0.201	0.426

However, the maxima of the convergence rate, which define the value  $p_{opt}$  are not always "sharp". Sometimes a small change of  $p$  gives a distinct change of convergence rate. For example, if  $N = 20$ ,  $p = 5$  the convergence rate is 0.748 while for  $p = 4$  it is 0.731.

The here described improvement of the method of Peaceman and Rachford obviously does not match a construction of an optimal rational function of the form (1) for the interval  $[\epsilon, 1-\epsilon]$ . Such a construction is described in [4], [5].

Table 2 contains some exemplary maximal deviations for the optimal function  $\omega_{opt}(\lambda)$ , constructed according to Wachspress, and for the function  $\omega(\lambda)$  constructed according to the algorithm (3).

Table 2.

$N$	$p$	$\max  \omega_{opt}(\lambda) $	$\max  \omega(\lambda) $
10	8	$1.11_{30} - 3$	$2.42_{10} - 3$
15	8	$2.84_{10} - 3$	$5.53_{10} - 3$
20	8	$4.91_{10} - 3$	$8.96_{10} - 3$
20	16	$1.20_{10} - 5$	$1.10_{10} - 4$
25	8	$7.09_{10} - 3$	$1.23_{10} - 2$
25	16	$2.51_{10} - 5$	$2.04_{10} - 4$
45	8	$1.57_{10} - 2$	$2.39_{10} - 2$
45	16	$1.23_{10} - 4$	$7.11_{10} - 4$
45	32	$7.83_{10} - 9$	$1.14_{10} - 6$

We also studied a more general form of algorithm (3) in which we assumed

$$\alpha_1 = \frac{1 - m\varphi^2}{1 + m\varphi^2} \frac{1}{\epsilon}$$

$$\alpha^* = \frac{1 + M\varphi^2}{1 - M\varphi^2} \frac{1}{1-\epsilon} ,$$

where  $m$  and  $M$  were some positive parameters. We found experimentally that if  $m$  and  $M$  are greater than unity, then the improvement obtained is insignificant. For example, if  $N = 30$ ,  $p = 8$ ,  $m = M = 1$  the error of approximation is  $1.54_{10} - 2$  while for  $M = m = 1.5$  it is  $1.46_{10} - 2$ . It seems that any essential improvement of this method can't be expected by fitting an appropriate  $m$  or  $M$ . The advantage of the proposed algorithm is a considerable simplicity of the realization.

All numerical experiments described in this paper have been computed on the computer G I E R at ZON /Computing Centre/ of Warsaw University.

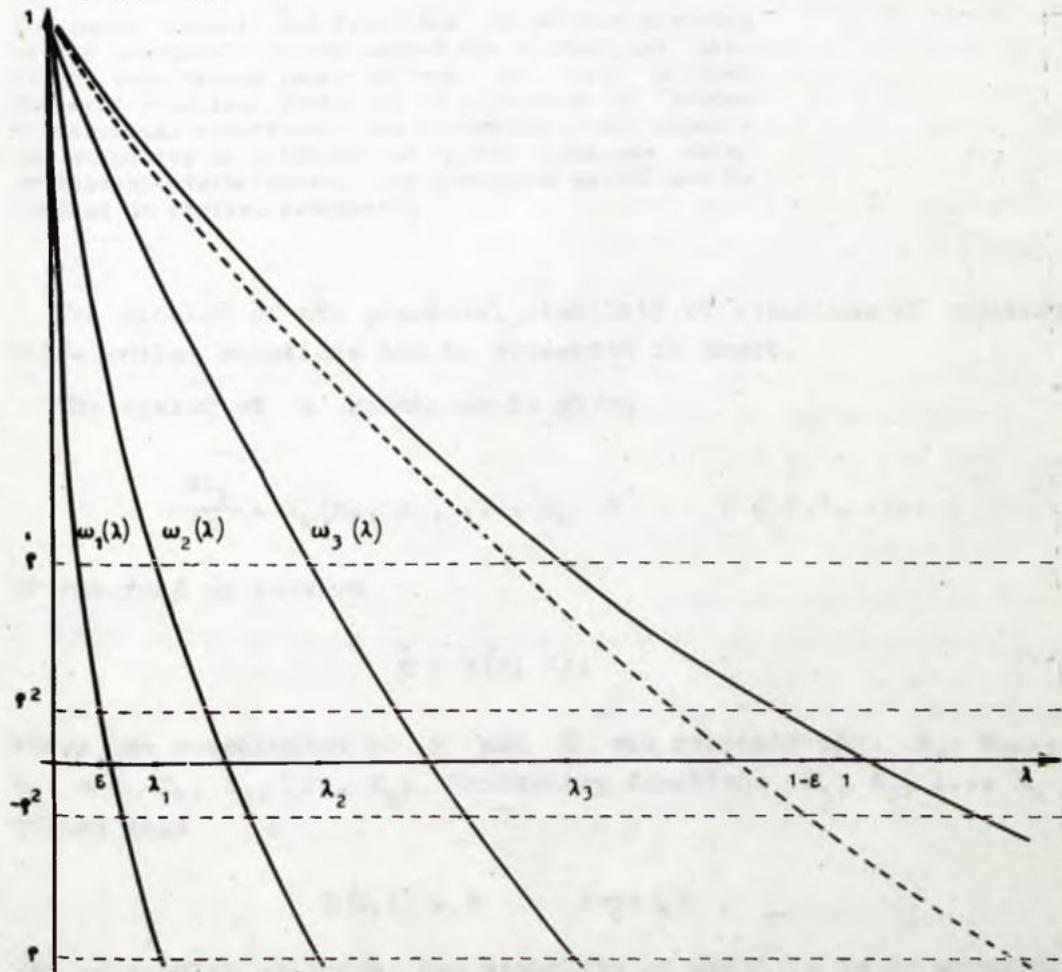


Fig. 1.

References

- [1] PEACEMAN D.W., RACHFORD H.H.: On numerical solution of parabolic and elliptic differential equations. J. SIAM vol. 3 /1955/.
- [2] FORSYTHE G.E., WASOW W.R.: Finite - difference methods for partial differential equations. NEW YORK - LONDON /1960/.
- [3] GRAM Ch. ... ed al.: Selected numerical methods. COPENHAGEN, REGNECENTRALLEN /1962/.
- [4] WACHSPRESS E.L.: Optimum alternating - direction - implicit iteration parameters for a model problem. J. SIAM vol. 10. 2/1962/.
- [5] WACHSPRESS E.L.: Extended application of alternating direction implicit iteration model problem theory. J. SIAM vol. 11. 4 /1963/.

AN APPLICATION OF NUMERICAL METHODS TO  
EXAMINE THE PRACTICAL STABILITY OF SOLUTIONS  
OF ORDINARY DIFFERENTIAL EQUATIONS

by Jacek OLSZEWSKI

Received March 14th, 1966

The paper presents modifications of certain theorems of the Liapunov's direct method and a practical stability test method based on them. The first theorem concerns practical stability of solutions of linear differential equations, the second one - on practical stability of solutions of linear equations under persistent disturbances. The presented method can be applied in digital computers.

The problem of the practical stability of solutions of ordinary differential equations can be presented in short.

The system of  $n$  equations is given

$$\frac{dx_k}{dt} = x_k(x_1, x_2, \dots, x_n, t) \quad k = 1, 2, \dots, n$$

or recorded as vectors

$$\dot{x} = X(x, t), \quad /1/$$

where the coordinates of  $x$  and  $X$  are respectively:  $x_1, x_2, \dots, x_n$  and  $X_1, X_2, \dots, X_n$ . Concerning functions  $X_1, X_2, \dots, X_n$  assume that

$$X(0, t) = 0 \quad 0 \leq t \leq T .$$

The particular solution, the stability of which is to be ascertained is:

$$x(t) = 0 \quad 0 \leq t \leq T .$$

/2/

We denote by  $H(a)$  the spherical surface with the radius  $a$  and its centre in point  $x = 0$ , thus the points for which  $\|x\| = a$  belong to  $H(a)$ , and by  $S(a)$  - the space bounded by the surface  $H(a)$   $/\|x\| < a/$ . The solution /2/ will be viewed as practically stable, if for three constants  $\epsilon$ ,  $\gamma$  and  $T$   $/\epsilon > \gamma > 0/$ , given in advance, the following statement holds: it results from  $x(0) \in S(\gamma)$  that  $x(t) \in S(\epsilon)$  for  $0 \leq t \leq T$ .

If the general solutions of equations /1/ are known then the stability of the solution /2/ can be deduced directly from it. In many cases it is very difficult or even impossible to obtain the general solution of system /1/. The method of examining the stability, known as the second Liapunov method, does not require the knowledge of the general solution of system /1/. Let us present the modification of one from among the theorems of this method on practical stability. In further considerations the main role will be played by a certain class of scalar functions  $V(x, t)$  assuming that:

1. they are continuous and have continuous partial derivatives of the first order in the region  $S(\epsilon)$  for  $0 \leq t \leq T$
2. there exists a certain constant  $m > 0$  such that

$$\begin{aligned} V(x, t) &\leq m \quad \text{for } x \in H(\gamma) \quad \text{and} \quad 0 \leq t \leq T \quad \text{and} \\ V(x, t) &\geq m \quad \text{for } x \in H(\epsilon) \quad \text{and} \quad 0 \leq t \leq T . \end{aligned}$$

It follows from assumption 1 that there exists grad  $V(x, t)$  and the derivative of the function  $V(x, t)$  can be computed along every trajectory described by the system of equations /1/:

$$\begin{aligned} \dot{V}(x, t) &= \dot{x} \cdot \text{grad } V(x, t) + \frac{\partial V(x, t)}{\partial t} \\ &= X \cdot \text{grad } V(x, t) + \frac{\partial V(x, t)}{\partial t} \end{aligned}$$

### Theorem 1

If a function  $V(x, t)$  exists such that its derivative computed along the trajectory of the system /1/, satisfies the condition

$$\dot{V}(x, t) < 0 \quad \text{for} \quad x \in K(\varepsilon, \gamma) \quad \text{and} \quad 0 \leq t \leq T. \quad /3/$$

where  $K(\varepsilon, \gamma)$  denotes the region in which occurs  $\gamma \leq \|x\| \leq \varepsilon$ , then the solution /2/ is practically stable.

### Proof

Consider the trajectory of the system /1/ described by point  $x(t)$ , for which  $x(0) \in S(\gamma)$  and suppose that it passes through the spherical surface  $H(\gamma)$ , this signifies that for a certain moment of time  $\tau /0 \leq \tau \leq T/ x(\tau) \in K(\varepsilon, \gamma)$ . The value of the function  $V(x, t)$  decreases along each trajectory in the region  $K(\varepsilon, \gamma)$ . Therefore none of the trajectories can reach the surface  $H(\varepsilon)$ .

As is seen, the most difficult problem is to find the function  $V(x, t)$  satisfying all above given conditions.

For differential equation systems:

$$\frac{dx_i}{dt} = \sum_{j=1}^n p_{ij}(t)x_j \quad i = 1, 2, \dots, n$$

it is convenient to accept that  $V(x, t)$  is a quadratic form:

$$V(x, t) = \sum_{i=1}^n \sum_{j=1}^n a_{ij}(t)x_i x_j.$$

Then, the derivative of the function  $V(x, t)$  is also a quadratic form

$$\dot{V}(x, t) = \sum_{i=1}^n \sum_{j=1}^n b_{ij}(t)x_i x_j,$$

where  $b_{ij}(t)$  are expressed by  $p_{ij}(t)$  and  $a_{ij}(t)$  by:

$$b_{11}(t) = \frac{da_{11}(t)}{dt} + 2 \sum_{j=1}^n p_{j1}(t) a_{j1}(t) \quad i = 1, 2, \dots, n,$$

$$b_{1j}(t) = \frac{da_{1j}(t)}{dt} + \sum_{k=1}^n [p_{ki}(t)a_{kj}(t) + p_{kj}(t)a_{ki}(t)]$$

$$i, j = 1, 2, \dots, n, \quad i \neq j.$$

Assuming, according to Dubošin [1], that the coefficients  $b_{ij}(t)$  are functions given in advance, the functions  $a_{ij}(t)$  are described by the following system of linear differential equations:

$$\frac{da_{11}}{dt} = b_{11}(t) - 2 \sum_{j=1}^n p_{j1}(t) a_{ji} \quad i = 1, 2, \dots, n,$$

/4/

$$\frac{da_{1j}}{dt} = b_{1j}(t) - \sum_{k=1}^n [p_{ki}(t)a_{kj} + p_{kj}(t)a_{ki}]$$

$$i, j = 1, 2, \dots, n, \quad i \neq j.$$

To ascertain the stability of solution /2/ it now suffices, according to the above given theorem, to accept such  $b_{ij}(t)$  which satisfies the condition /3/, and find even one particular solution of the system of equations /4/ satisfying the assumption 2 of the function  $V(x, t)$ . Assumption 1 is satisfied as  $V(x, t)$  is accepted as a quadratic form.

The initial conditions, otherwise the values  $a_{ij}^{(0)}$ , being given for the function  $a_{ij}(t)$ , the solution will be obtained by means of a numerical integrating of equations /4/. After each integrating step, we shall check whether the conditions

$$V(x, \tau) \leq m \quad \text{for } x \in H(\eta) \quad \text{and}$$

$$V(x, \tau) \geq m \quad \text{for } x \in H(\varepsilon)$$

are satisfied. If so then for all moments of the time  $\tau /0 \leq \tau \leq T$ , the solution /2/ may be treated as practically stable. From the geometrical viewpoint the above conditions mean that all points of the surface described by the equation

$$V(x, \tau) = m \quad /5/$$

belong to the region  $k(\varepsilon, \eta)$ , i.e.:

$$\max(D(x_m)) \leq \varepsilon \quad \text{and} \\ \min(D(x_m)) \geq \eta,$$

where  $D(x_m)$  is the distance between the beginning of the system of coordinates and point  $x_m$  being on the surface described by the equation /5/.

As  $V(x, \tau)$  is a quadratic form with coefficients  $a_{ij}(\tau)$ ,  $\max(D(x_m))$  and  $\min(D(x_m))$  will be expressed by means of the greatest and the smallest eigenvalue of the matrix  $\{a_{ij}(\tau)\}$ .

It is easy to see that

$$\max(D^2(x_m)) = \frac{m}{\lambda_{\min}(\tau)} \quad \text{and}$$

$$\min(D^2(x_m)) = \frac{m}{\lambda_{\max}(\tau)},$$

where  $\lambda_i(\tau)$  are eigenvalues of the matrix  $\{a_{ij}(\tau)\}$ .

It is therefore enough to check whether

$$\lambda_{\min}(\tau) \geq \frac{m}{\varepsilon^2} \quad \text{and}$$

$$\lambda_{\max}(\tau) \leq \frac{m}{\eta^2}.$$

The computation of all values  $\lambda_i(\tau)$  is not especially difficult, as the matrix of coefficients of quadratic form is symmetri-

cal. It, however, requires a relatively big amount of counting. In this case methods can be used that permit to estimate the upper and lower bound for  $\lambda_i(\tau)$  by means of a few simple operations. The simplest of them is the method of eigenvalue localization, known as Geršgorin circles [2]. It is based on the theorem that points representing eigenvalues of the matrix  $\{a_{ij}(\tau)\}$  on a complex plane, belonging to the region which is the sum of  $n$  circles:

$$\left| a_{ii}(\tau) - s \right| \leq \sum_{\substack{k=1 \\ k \neq i}}^n |a_{ik}(\tau)| \quad i = 1, 2, \dots, n.$$

Eigenvalues of the symmetrical matrix are real, thus, we can write:

$$a_{ii}(\tau) - \sum_{\substack{k=1 \\ k \neq i}}^n |a_{ik}(\tau)| \leq s \leq a_{ii}(\tau) + \sum_{\substack{k=1 \\ k \neq i}}^n |a_{ik}(\tau)|$$

$$i = 1, 2, \dots, n.$$

We then have for  $\lambda_i(\tau)$  the upper bound  $L(\tau)$  and the lower one  $l(\tau)$ :

$$L(\tau) = \max_i (a_{ii}(\tau) + \sum_{\substack{k=1 \\ k \neq i}}^n |a_{ik}(\tau)|)$$

$$l(\tau) = \min_i (a_{ii}(\tau) - \sum_{\substack{k=1 \\ k \neq i}}^n |a_{ik}(\tau)|).$$

Somewhat better estimation of values  $L(\tau)$  and  $l(\tau)$  gives the method of Cassini ovals [2]. The proper theorem is the following: points representing eigenvalues of the matrix  $\{a_{ij}(\tau)\}$ , belong to the region which is the sum of  $\frac{n(n-1)}{2}$  ovals:

$$\left| s - a_{ii}(\tau) \right| \left| s - a_{jj}(\tau) \right| \leq \sum_{\substack{k=1 \\ k \neq i}}^n \left| a_{ik}(\tau) \right| \sum_{\substack{k=1 \\ k \neq j}}^n a_{jk}(\tau)$$

$$i, j = 1, 2, \dots, n, \quad i \neq j$$

Proceeding similarly as previously, we obtain

$$- \sum_{\substack{k=1 \\ k \neq i}}^n \left| a_{ik}(\tau) \right| \sum_{\substack{k=1 \\ k \neq j}}^n \left| a_{jk}(\tau) \right| \leq (s - a_{ii}(\tau)) (s - a_{jj}(\tau)) \leq$$

$$\leq \sum_{\substack{k=1 \\ k \neq i}}^n \left| a_{ik}(\tau) \right| \sum_{\substack{k=1 \\ k \neq j}}^n \left| a_{jk}(\tau) \right| \quad i, j = 1, 2, \dots, n, \quad i \neq j$$

The values  $L(\tau)$  and  $l(\tau)$  will be determined by the solutions  $s'_{ij}(\tau)$  and  $s''_{ij}(\tau)$  of  $\frac{n(n-1)}{2}$  quadratic equations:

$$s^2 - (a_{ii}(\tau) + a_{jj}(\tau))s + a_{ii}(\tau)a_{jj}(\tau) - \sum_{\substack{k=1 \\ k \neq i}}^n \left| a_{ik}(\tau) \right| \sum_{\substack{k=1 \\ k \neq i \\ k \neq j}}^n \left| a_{jk}(\tau) \right| =$$

$$i, j = 1, 2, \dots, n, \quad i \neq j$$

Assuming  $s'_{ij}(\tau) > s''_{ij}(\tau)$  we obtain:

$$L(\tau) = \max_{ij} (s'_{ij}(\tau)) \quad \text{and}$$

$$l(\tau) = \min_{ij} (s'_{ij}(\tau))$$

It follows from the above that the checking of the assumption 2 of the function  $v(x, t)$  consists in numerical integrating of equations /4/, and that for every moment of time  $\tau = 0, h, 2h, 3h, \dots, T$ , where  $h$  is an integration step, the validity of the below given conditions is being checked as follows:

$$L(\tau) \leq \frac{m}{\eta^2} \quad \text{and}$$

$$l(\tau) \geq \frac{m}{\epsilon^2},$$

where  $L(\tau)$  and  $l(\tau)$  are determined by means of one of the two above presented methods.

The problem of ascertaining the practical stability of the solution /2/ is more complicated, as the equation system /1/ is the one of nonlinear differential equations. In many such cases it may be helpful to modify a little the theorem on stability under persistent perturbances proved in its initial form by Malkin [3].

We shall consider the system of differential equations

$$\dot{x} = X(x, t) + R(x, t), \quad /6/$$

where the functions  $R(x, t)$  are perturbances assumed to be small. We shall write this assumption as follows:

$$\|R(x, t)\| < \eta, \quad \text{for } x \in s(\epsilon) \quad \text{and} \quad 0 \leq t \leq T, \quad \eta > 0.$$

The particular solution, the stability of which is to be stated is also:

$$x(t) = 0. \quad /7/$$

Now, we shall treat the equation system /1/ as the one of simplified equations, perturbances not being taken into account.

### Theorem 2

If a function  $V(x, t)$  exists such that its derivative, computed along the trajectory of the unperturbed system /1/, satisfies the condition:

$$\dot{v}(x, t) < -\mu \quad \text{for} \quad x \in K(\epsilon, \eta) \quad \text{and} \quad 0 \leq t \leq T, \quad /8/$$

where  $\mu$  is a positive constant depending on  $\epsilon$  and  $\eta$ , and also if the conditions

$$\left| \frac{\partial v(x, t)}{\partial x_i} \right| \leq M \quad i = 1, 2, \dots, n \quad \text{dla} \quad 0 \leq t \leq T, \\ x \in K(\epsilon, \eta)$$

$$\text{and} \quad \eta_1 < \frac{M}{n\mu}$$

will be satisfied, then the solution /7/ of the equation system /6/ is practically stable.

### Proof

Let us denote the derivative of the function  $v(x, t)$ , computed along the trajectory of the system /6/, by  $\dot{v}_*(x, t)$ :

$$\begin{aligned} \dot{v}_*(x, t) &= (x(t) + R(x, t)) \cdot \text{grad } v(x, t) + \frac{\partial v(x, t)}{\partial t} \\ &= \dot{v}(x, t) + R(x, t) \cdot \text{grad } v(x, t). \end{aligned}$$

Similarly as we did, let us consider the trajectory of the system /6/, described by point  $x(t)$ , for which  $x(0) \in S(\eta)$  and let us suppose that it passes through the spheric surface  $S(\eta)$  i.e. such  $\tau$  ( $0 \leq \tau \leq T$ ) exists, for which we have  $x(\tau) \in K(\epsilon, \eta)$ . For points belonging to the region  $K(\epsilon, \eta)$  we have

$$\begin{aligned} \dot{v}_*(x, \tau) &\leq -\mu + \text{grad } v(x, \tau) \cdot R(x, \tau) \leq -\mu + \sum_{i=1}^n \left| \frac{\partial v(x, \tau)}{\partial x_i} \right| R_i(x, \tau) \leq \\ &\leq -\mu + n M \|R(x, \tau)\| \leq -\mu + n M \eta_1 < 0, \end{aligned}$$

which means that along every trajectory of the system /6/ in the region  $K(\epsilon, \eta)$ , the value of the function  $v(x, t)$  decreases.

Therefore, none of these trajectories will reach the surface  $H(\varepsilon)$ .

Notice, that this theorem permits us to examine the stability of solution /7/ even when the forms of function  $R(x, t)$  are unknown, for example, they may be some external, accidental influences, noises and so on, in physical and technical processes.

Let us also notice that in the case of a nonlinear differential equation system:

$$\dot{x} = X(x, t)$$

the above theorem allows us to solve the problem of ascertaining the practical stability of solution:

$$x(t) = 0$$

by a method similar to the above described. For this purpose we rewrite our equation system as follows:

$$\dot{x} = X_1(x, t) + \zeta_w(x, t), \quad /9/$$

where  $X_1(x, t)$  denotes the linear functions of variables  $x_1, x_2, \dots, x_n$ , and  $\zeta_w(x, t)$  do not comprise terms of a lower order than 2 because of  $x_1, x_2, \dots, x_n$ .

Further we shall treat  $\zeta_w(x, t)$  as perturbations disturbing the system:

$$\dot{x} = X_1(x, t) \quad . \quad /10/$$

In order to apply theorem 2 to this case such a value  $\mu$  must be accepted for which occurs the following

$$\mu > n \max (\|\zeta_w(x, t)\|) \quad \text{for } x \in k(\varepsilon, \eta) \text{ and } 0 \leq t \leq T.$$

Now, similarly as for linear differential equations, we may assume that the function  $V(x, t)$  is a quadratic form

$$V(x, t) = \sum_{i=1}^n \sum_{j=1}^n a_{ij}(t) x_i x_j ,$$

and therefore, this function derivative, computed along the trajectory of the system /10/, will also be a quadratic form

$$\dot{v}(x, t) = \sum_{i=1}^n \sum_{j=1}^n b_{ij}(t) x_i x_j$$

The relations between  $a_{ij}(t)$  and  $b_{ij}(t)$  are given in the form of equations /4/. We further proceed as in the method which makes use of theorem 1, and we assume such functions  $b_{ij}(t)$  which satisfy condition /8/. Besides examining whether the assumption 2 of the function  $v(x, t)$  is satisfied, one has to check whether for all moments of the time  $\tau = 0, h, 2h, 3h, \dots, T$  the following occurs:

$$\left| \frac{\partial v(x, \tau)}{\partial x_i} \right| \leq M \quad \text{for} \quad x \in K(\epsilon, \eta), \quad i = 1, 2, \dots, n.$$

It is easy to prove that in case  $v(x, \tau)$  is a quadratic form with coefficients  $a_{ij}(\tau)$ , the above condition can be written as follows:

$$\sum_{j=1}^n a_{ij}^2(\tau) \leq \frac{M^2}{4\epsilon^2}, \quad i = 1, 2, \dots, n.$$

Both presented theorems are particular cases of theorems contained in position [3], where a more general idea of the practical stability is considered.

Now, in the Computing Centre of the Polish Academy of Sciences work is being carried out in which the described method is applied.

References

- [1] DUBOŠIN G.J.: Osnovy teorii ustojčivosti dvizhenija. 1952, Izdatelstvo Moskovskogo Universiteta.
- [2] PARODI M.: La localisation des valeurs caractéristiques des matrices. 1959, Gauthier - Villars.
- [3] LA SALLE J., LEFSCHETZ S.: Stability by Liapunov's Direct Method. 1961, Academic Press.

APPROXIMATION OF THE SOLUTION OF A UNIFORM  
SHELL DEFORMATION CAUSED BY THE DETONATION  
OF THE EXPLOSIVE CHARGE OF FINITE THICKNESS

by Franciszek LABISCH

Received May 4th, 1966

The deformation of a thin shell with a constant surface density caused by a detonation front is considered. An exact solution of the obtained mathematical problem is given for the case of a charge of infinite thickness. The properties of this solution are described. Two theorems on the approximation of a solution for the case of a charge with finite thickness are proved.

Paper [3] presents a solution of the problem of a deformation of an initially plane shell with a constant or linearly variable surface density produced by the pressure of detonation products of an infinite explosive charge. Three cases are considered. In each of them the flow of burnt products is found behind the detonation wave, in the class of simple waves, i.e. the vector of the flow velocity

$$\vec{v} = (u, v, w) \quad \text{is} \quad \vec{v} = \vec{f}(a), \quad \text{where } a \text{ is the sound speed [1]}$$

By means of a certain theorem from paper [2] the solution of this problem was reduced to the solution of a nonlinear system of ordinary differential equations, the initial conditions being given. In this way the proof of the existence of an exact solution of this problem was obtained. Moreover, for the first case the solution is in a finite form. For the two remaining ones the solution is obtained by means of integration systems of ordinary differential equations. It was carried out on the ZAM-2 computer, using a standard subroutine.

The applied method permits to reduce multidimensional quasilinear systems of partial differential equations of a hyperbolic type of certain Cauchy's problems to the systems of ordinary differential equations. This proves the existence of the solution. The method also deserves attention because of a high degree of accuracy of the obtained numerical results, an easy way of programming, and a short time of computing.

The assumption of an infinite thickness of the detonation charge is essential. In practice, there appear only charges of a finite thickness.

The present paper gives the solution of the problem of the deformation of the shell with a constant surface density in the case of a detonation wave propagating in a charge of infinite thickness.

The properties of this solution are described, as well as the difficulties appearing while considering a charge of finite thickness. Two theorems were proved on the approximation of the problem solution in the case of a finite explosion charge by means of solving a case of infinite thickness.

## 1. THE PROBLEM DESCRIPTION

Given a thin plane shell of a constant surface density, contacting a uniform detonation charge of a constant thickness  $d$ . A plane detonation wave, normal to the nondeformed part of the shell, propagates in the detonation charge with a constant velocity  $U$ .

Assume that Chapman - Jouguet conditions on the detonation wave are satisfied and that the flow of burnt products behind the detonation front can be described by means of gas dynamic equations. Moreover, it is accepted that the shell deformation occurs in a way which does not permit a direct interaction of separate elements. Assuming that the origin of the frame of reference is on the straight line of the shell intersection with the detonation front, and considering only a plane case because of the constant surface density, one obtains the following mathematical problem.

Find the flow, described by the equations of gas dynamics

$$a_t + u \cdot a_x + v \cdot a_y + \frac{K-1}{2} a (u_x + v_y) = 0$$

$$u_t + u \cdot u_x + v \cdot u_y + \frac{2}{K-1} a \cdot a_x = 0$$

$$v_t + u \cdot v_x + v \cdot v_y + \frac{2}{K-1} a \cdot a_y = 0$$

limited by a free line, depending on the solution and starting in point  $x = 0$ ,  $y = d$ , on which  $a = 0$ , and such that for  $x = 0$ ,  $0 \leq y \leq d$  the Chapman - Jouguet condition is satisfied

$$u(0, y, t) = a_*, \quad v(0, y, t) = 0 \quad /1.2./$$

and which on the line

$$x = \xi(\alpha; t); \quad y = \eta(\alpha; t) \quad /1.3./$$

satisfies the flow condition:

$$-\eta_\alpha \cdot \xi_t + \xi_\alpha \cdot \eta_t = -\eta_\alpha \cdot u + \xi_\alpha \cdot v. \quad /1.4./$$

The curve /1.3./ depends on the solution of the set /1.1./ and can be determined by the equations

$$\begin{aligned} m_0 \cdot \xi_{tt}(\alpha; t) &= \eta_\alpha \cdot p_* \left( \frac{a(\alpha; t)}{a_*} \right)^{\frac{2K}{K-1}} \\ m_0 \cdot \eta_{tt}(\alpha; t) &= -\xi_\alpha \cdot p_* \cdot \left( \frac{a(\alpha; t)}{a_*} \right)^{\frac{2K}{K-1}} \end{aligned} \quad /1.5./$$

and by the initial conditions

$$\xi(\alpha; -\frac{\alpha}{U}) = 0, \eta(\alpha; -\frac{\alpha}{U}) = 0,$$

$$\xi_t(\alpha; -\frac{\alpha}{U}) = u, \eta_t(\alpha; -\frac{\alpha}{U}) = 0 . \quad /1.6./$$

Here,  $a$  is the sound speed,  $a_*$  - the critical speed,  $p_*$  - the critical pressure<sup>1/</sup>,  $u, v$  - components of the velocity vector of the flow of the burnt products,  $K$  - the adiabatic exponent, the parameter  $\alpha$  is a Lagrange's variable. This variable is chosen so that for  $\alpha \leq 0$  is  $\xi(\alpha; 0) = \alpha$ ,  $\eta(\alpha; 0) = 0$  and in point  $x=0$ ,  $y=0$  the condition  $\alpha + U \cdot t = 0$  is satisfied. Here  $t$  is the time,  $U$  - the known constant speed of the detonation wave propagation.

## 2. FORMAL SOLUTION OF THE PROBLEM

The vector  $\vec{r}(\alpha; t) = [\xi(\alpha; t); \eta(\alpha; t)]$  describing the shape of the deformed shell is sought for in the form

$$\vec{r}(\alpha; t) = \vec{r}(\omega); \omega \stackrel{df}{=} \alpha + U \cdot t \quad \text{see [3]} -$$

the equations /1.5./ take the form

$$m_0 \cdot U^2 \cdot \xi_{\omega\omega}(\omega) = p_* \cdot \eta_{\omega\omega}(\omega) \cdot \left( \frac{a(\omega)}{a_*} \right)^{\frac{2K}{K-1}}$$

$$m_0 \cdot U^2 \cdot \eta_{\omega\omega}(\omega) = -p_* \xi_{\omega\omega}(\omega) \cdot \left( \frac{a(\omega)}{a_*} \right)^{\frac{2K}{K-1}} \quad /2.1./$$

the conditions /1.6./ are

$$\xi(0) = 0, \eta(0) = 0;$$

$$\xi_\omega(0) = 1, \eta_\omega(0) = 0 \quad /2.2./$$

<sup>1/</sup>Critical speed and critical pressure are the local values of the speed of sound and of the pressure in the points where the speed of flow equals the speed of sound.

and the flow condition /1.4./ is

$$-\eta_\omega \cdot u + \xi_\omega \cdot v = 0 . \quad /2.3./$$

One has to find the flow that satisfies the condition  $u = a_*$ ,  $v = 0$ , on the interval OC /fig. 1 a/, flowing around the curve OA, and being limited from above by a free line CB, on which  $a = 0$ . The curve OA depends on the flow, and is to be determined by equations /2.1./ and conditions /2.2./.

In the hodograph plane  $\frac{u}{a_*}, \frac{v}{a_*}$  we obtain the following image of the flow /fig. 1 b/.

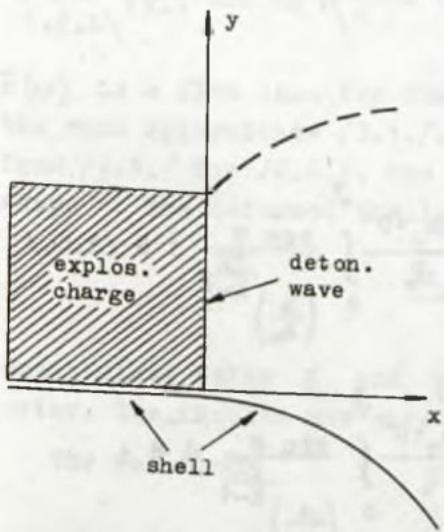


Fig. 1a.

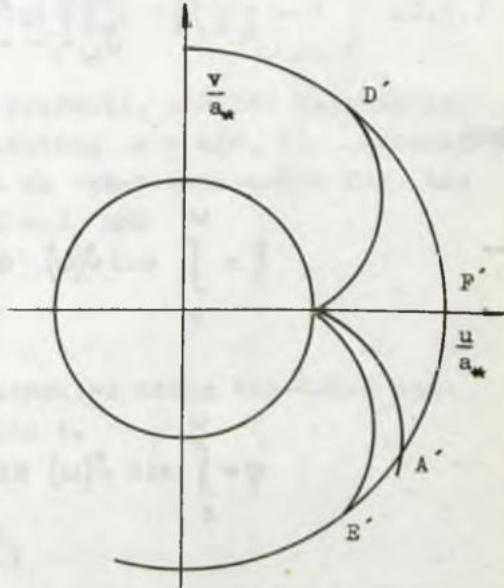


Fig. 1b.

Assume the existence of a stationary solution of the considered problem. It is justified by a constant surface density and by the accepted conditions on the detonation front.

The curve OA corresponds to the curve O'A', the segment OC is mapped into point O', the entire epicycloide O'D' corresponds to point O. The image of the flow of burnt products in a hodograph plane is therefore limited by the curve O'A', the epicycloide O'E' and the arc of the maximal speed D'F'A'. It follows from the theorem proved in [3] that

$$\xi_\omega = \cos \vartheta; \quad \eta_\omega = \sin \vartheta, \quad /2.4./$$

where  $\vartheta$  - is the angle between the axis  $x$  and the tangent to the curve OA.

Substituting /2.4./ for /2.1./ one obtains

$$\vartheta_\omega = - \frac{p_* \left( \frac{a}{a_*} \right)^{\frac{2K}{K-1}}}{m_0 \cdot U^2} \quad /2.5./$$

and

$$\xi = \int_0^\omega \cos \vartheta(\omega) d\omega = - \frac{m_0 \cdot U^2}{p_*} \int_0^\vartheta \frac{\cos \theta}{\left( \frac{a}{a_*} \right)^{\frac{2K}{K-1}}} d\theta \quad /2.6./$$

$$\eta = \int_0^\omega \sin \vartheta(\omega) d\omega = - \frac{m_0 \cdot U^2}{p_*} \int_0^\vartheta \frac{\sin \theta}{\left( \frac{a}{a_*} \right)^{\frac{2K}{K-1}}} d\theta,$$

where  $a = a(\theta, K)$  is the value of the sound speed, depending on the angle  $\theta$ , and on the adiabatic exponent on the deformed part of the shell, therefore, on the curve OA, or on its image O'A'. In order to find the shape of the deformed part of the shell, it is sufficient to find the image of the curve in the hodograph plane, thus the function  $a = a(\theta, K)$

3. THE EXACT SOLUTION OF THE PROBLEM ON THE DEFORMATION OF A SHELL WITH A CONSTANT SURFACE DENSITY IN THE CASE OF INFINITE THICKNESS OF THE DETONATION CHARGE

If the detonation charge thickness is infinite, the free line where  $a = 0$  is not to be taken into account. In this case, on the basis of the theorem proved in [2], the deformed shell is admissible as the initial manifold for simple wave flows. Therefore, there exists a simple wave flow satisfying the system /1.1./, the Chapman-Jouguet condition /1.2./ and flowing around the curve  $\bar{r} = \bar{r}(\omega)$ . It is known that the image of this flow is the entire epicycloid  $O'E'$ . There occurs

$$\theta = -\sqrt{\frac{K+1}{K-1}} \operatorname{arc} \operatorname{tg} \sqrt{\left(\frac{a_*}{a}\right)^2 - 1} + \operatorname{arc} \operatorname{tg} \left( \sqrt{\frac{K+1}{K-1}} \cdot \sqrt{\left(\frac{a_*}{a}\right)^2 - 1} \right) \quad /3.1./$$

$\bar{r}(\omega)$  is a flow line for the burnt products, and its mapping is the same epicycloide /3.1./. Substituting  $a = a(0, K)$ , determined from /3.1./ for /2.6./, one obtains an exact expression for the shape of the deformed shell. For  $K = 3$  and

$$\frac{m_0 \cdot U^2}{p_*} = 1$$

appropriate value  $\xi$  and  $\eta$  are computed using the ZAM-2 computer. The results are given in Table 1.

The relation

$$\frac{p}{p_*} = \left(\frac{a}{a_*}\right)^{\frac{2K}{K-1}}$$

being taken into account, the solution obtained may be presented as follows

$$\xi = -\frac{m_0 U^2}{p_*} \cdot \int_0^\theta \frac{\cos \theta}{\frac{p}{p_*}} d\theta; \quad /3.2./$$

$$\eta = - \frac{m_0 \cdot U^2}{p_*} \cdot \int_0^y \frac{\sin \theta}{\frac{p}{p_*}} d\theta ,$$

/3.2./

where

$$\theta = - \sqrt{\frac{K+1}{K-1}} \operatorname{arc tg} \sqrt{\left(\frac{p}{p_*}\right)^{\frac{1-K}{K}}} - 1 + \operatorname{arc tg} \left( \sqrt{\frac{K+1}{K-1}} \cdot \sqrt{\left(\frac{p}{p_*}\right)^{\frac{1-K}{K}}} - 1 \right) .$$

It is evident from /3.2./ that any curve obtained in this way is similar to the curve

$$\xi = - \int_0^y \frac{\cos \theta}{\frac{p}{p_*}} d\theta$$

/3.4./

$$\eta = - \int_0^y \frac{\sin \theta}{\frac{p}{p_*}} d\theta ,$$

where  $\theta$  is given by /3.3./ and

$$\frac{m_0 \cdot U^2}{p_*}$$

is the similarity coefficient.

The flow of burnt products is mapped on the entire epicycloide. It, therefore, results that the maximal angle of the shell deviation is given by

$$\vartheta_m = \left( 1 - \sqrt{\frac{K+1}{K-1}} \right) \cdot \frac{\pi}{2}$$

For  $K = 1,4$   $\vartheta_m = - 130,4^\circ$ , and for  $K = 3$ ,  $\vartheta_m = - 37,3^\circ$ .

From the formula for the radius of curvature  $R$  of a plane curve in the natural form

$$R = \frac{1}{\xi^2 \omega \omega + \eta^2 \omega \omega}$$

one immediately obtains that the value of the radius of curvature for the curve /3.4./ is given by  $\frac{p}{p_*}$ .

This fact can be used when solving the given problem by a graphical method.

For  $\xi = 0$ , therefore for  $\omega = 0; \eta = 0, p = p_*$  and

$$\xi_\omega = 1, \eta_\omega = 0, \xi_{\omega\omega} = 0, \eta_{\omega\omega} = \theta_\omega.$$

With the formulae

$$\frac{d\eta}{d\xi} = \frac{\eta_\omega}{\xi_\omega}$$

$$\frac{d^2\eta}{d\xi^2} = \frac{\xi_\omega \cdot \eta_{\omega\omega} - \eta_\omega \cdot \xi_{\omega\omega}}{\xi_\omega^3}$$

one obtains

$$\left. \frac{d\eta}{d\xi} \right|_{\omega=0} = 0$$

$$\left. \frac{d^2\eta}{d\xi^2} \right|_{\omega=0} = \frac{-p_*}{m_0 \cdot U^2}$$

and the development in the neighbourhood of  $\xi = 0$

$$\eta = -\frac{p_*}{2 m_0 \cdot U^2} \cdot \xi^2 + O(\xi^3).$$

It may be shown that

$$\lim_{a \rightarrow 0} \int_{a_*}^a \frac{\operatorname{tg} \vartheta - \cos \theta - \sin \theta}{\left(\frac{a}{a_*}\right)^{\frac{2K}{K-1}}} \cdot \theta_a da,$$

where  $\theta_a$ , determined from /3.1./, does not exist, and then it results that the curve determining the shape of the deformed shell has no asymptote.

#### 4. THE APPROXIMATION OF THE SOLUTION OF THE PROBLEM OF A UNIFORM SHELL DEFORMATION IN THE CASE OF A CHARGE OF FINITE THICKNESS

The speed of the flow of burnt products on the detonation front is equal to the local speed of sound. Here, the system /1.1./ is of a parabolic type. The two characteristics, existing in a stationary case for a hyperbolic region, coincide in the parabolic region. Both these characteristics are normal to the flow line, and they coincide with the detonation front. The region of influence of the singular point in which the detonation wave reaches the bounds of the detonation charge, comprises the point in which the detonation front reaches the shell. A simple wave flow /4/, does not exist in the case of a finite thickness of the detonation charge.

It, however, appears that in a certain neighbourhood of the detonation front, the results obtained in the case of an explosive charge of infinite thickness, are an arbitrary good approximation in the case of a charge of finite thickness.

For increasing  $d$  the line  $O'A'$  in the hodograph plane approaches more and more the epicycloide  $O'E'$ , given by /3.1./, at the deformed shell - the curve /3.2./. The following two theorems can be proved.

Theorem 1

For every  $\epsilon > 0$  and for every interval  $I = [\tilde{\theta}, 0]$ ;  $\theta_m < \tilde{\theta} < 0$  there exists  $d = d(\tilde{\theta}, \epsilon)$  such that  $|\xi - \xi_d| \leq \epsilon$  and  $|\eta - \eta_d| \leq \epsilon$  for  $\theta \in I$ .

Theorem 2

For every  $\epsilon > 0$  and for every value  $d > 0$  there exists an interval  $I = [\tilde{\theta}, 0]$  such that  $|\xi - \xi_d| \leq \epsilon$  and  $|\eta - \eta_d| \leq \epsilon$  for  $\theta \in I$

$\vec{r} = [\xi, \eta]$  - is the solution for  $d = \infty$

$\vec{r}_d = [\xi_d, \eta_d]$  - is the solution for a finite value  $d$ .

Let the curve OA present the shape of the shell deformed by a charge with  $d = \infty$ , and the curve OB - the shape of the shell deformed by a charge with  $d < \infty$

Let the arc of the epicycloide O'A' present the hodograph image of the curve OA, in the hodograph plane, and the curve O'B' - the image of the curve OB. The value  $p_1 < p(\tilde{\theta})$  for  $p = p(\tilde{\theta})$  determined from /3.3./ is being chosen so as to obtain

$$m_0 \cdot U^2 \cdot \sin \tilde{\theta} \left( \frac{1}{p(\tilde{\theta})} - \frac{1}{p_1} \right) = \epsilon . \quad /4.1./$$

Thus

$$p_1 = \frac{m_0 U^2 \cdot \sin \tilde{\theta} \cdot p(\tilde{\theta})}{\epsilon \cdot p(\tilde{\theta}) + m_0 U^2 \sin \tilde{\theta}} . \quad /4.2./$$

It is known that

$$\xi = - m_0 U^2 \int_0^{\tilde{\theta}} \frac{\cos \theta}{p(\theta)} d\theta$$

$$\xi_d = - m_0 U^2 \int_0^{\beta} \frac{\cos \theta}{p_d(\theta)} d\theta,$$

where  $p_d(\theta)$  is given by /3.3/.

$p_d(\theta)$  is the pressure as the function of the angle  $\theta$  along the curve OB, or O'B'.

The maximal difference among the solutions evidently appears for the value  $\theta$  on the left end of the interval I, therefore, for  $\theta = \tilde{\theta}$ , it means in points A and B.

Points A and B correspond to the value  $\tilde{\theta}$ , and  $p_d(\tilde{\theta}) = p_1$  in the point B or B'

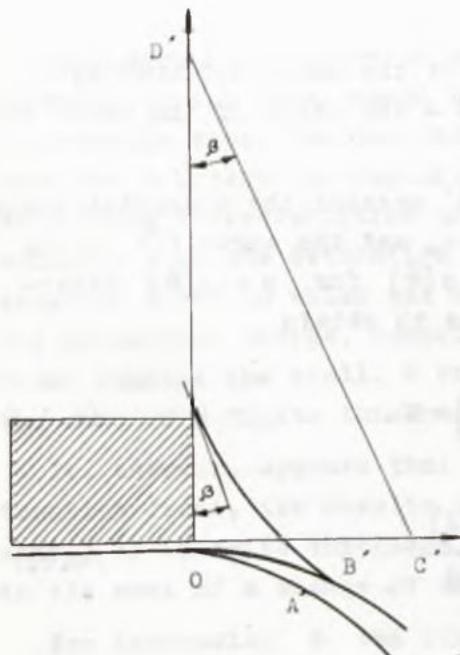


Fig. 2a

$$OC = \frac{\sin \tilde{\theta}}{p_1}$$

$$OD = d$$

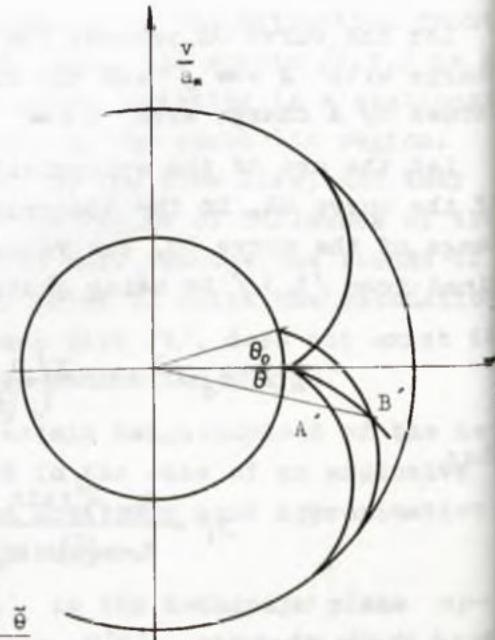


Fig. 2b

Thus

$$\left| \xi - \xi_d \right| \leq m_0 U^2 \left| \int_0^{\tilde{\theta}} \left( \frac{\cos \theta}{p(\theta)} - \frac{\cos \theta}{p_d(\theta)} \right) d\theta \right| < m_0 U^2 \left| \int_0^{\tilde{\theta}} \cos \theta d\theta \right| \cdot \left| \frac{1}{p_1} - \frac{1}{p(\tilde{\theta})} \right|$$

$$= m_0 U^2 \sin \tilde{\theta} \left( \frac{1}{p} - \frac{1}{p_1} \right) = \varepsilon$$

and

$$\left| \eta - \eta_d \right| \leq m_0 U^2 \left| \int_0^{\tilde{\theta}} \left( \frac{\sin \theta}{p(\theta)} - \frac{\sin \theta}{p_d(\theta)} \right) d\theta \right| \leq m_0 U^2 (1 - \cos \tilde{\theta}) \left( \frac{1}{p} - \frac{1}{p_1} \right) \leq$$

$$\leq m_0 U^2 \sin \tilde{\theta} \left( \frac{1}{p} - \frac{1}{p_1} \right) = \varepsilon .$$

Such a charge thickness  $d$  should be found, for which the above mentioned approximations occur. For this purpose let us consider the characteristic starting in a physical plane, at point B and passing through the singular point  $x = 0, y = d$ . The image of the characteristic in the hodograph plane is well known. In the physical plane the above mentioned characteristic intersects the axis  $y$  in the singular point, making the angle  $\beta = \frac{\theta_0}{2} + \frac{\pi}{2} - \alpha \left( \frac{\theta_0}{2} \right)$  where  $\theta_0 = \theta(p) - \theta(p_1)$ ;  $\theta(p_1)$  is the value of  $\theta$ , computed according to /3.3./ for  $p = p_1$ , and  $\alpha = \alpha \left( \frac{\theta_0}{2} \right)$  is to be computed from the relation

$$\theta - \theta_0 = \left[ \frac{\pi}{2} - \alpha - \sqrt{\frac{K+1}{K-1}} \operatorname{arc} \operatorname{tg} \left( \sqrt{\frac{K-1}{K+1}} \operatorname{ctg} \alpha \right) \right] \quad /4.3./$$

for  $\theta = \frac{\theta_0}{2} + \alpha$  is the Mach angle determined by the relation

$$\sin^2 \alpha = \frac{a^2}{u^2 + v^2}$$

For a certain region  $\theta = \text{arc} \operatorname{tg} \frac{v}{u}$  the considered characteristic forms with the axis  $x$  an angle which is no smaller than  $\beta$  /fig. 2a/. The exact shape of the characteristic in the physical plane is unknown. It is known that

$$\frac{\cos \tilde{\theta}}{p(\tilde{\theta})} > \xi(\tilde{\theta}).$$

When increasing the thickness of the charge so as to obtain

$$\frac{\sin \tilde{\theta}}{p_1 \cdot d} = \operatorname{tg} \beta,$$

one gets the thickness  $d$ , for which the theorem 1 holds.

The proof of theorem 2 is based on the same construction, but realized in a different order. One looks for such values  $p$  and  $p_1$  as to obtain

$$m_0 U^2 \sin \theta \frac{1}{p} - \frac{1}{p_1} = \varepsilon$$

and

$$\frac{\sin \theta}{p_1 \operatorname{tg} \left( \frac{\theta_0}{2} + \frac{\pi}{2} - \alpha \left( \frac{\theta_0}{2} \right) \right)} = d.$$

$\theta$  - is a function of  $p$ , determined by /3.3./  
 $\theta_0 = \theta(p) - \theta(p_1)$  and the value  $\alpha \left( \frac{\theta_0}{2} \right)$

should be determined from /4.3./.

The interval  $I = [\tilde{\theta}, 0]$  where

$\tilde{\theta}$  is the value  $\theta$  which according to /3.3./ corresponds to the value  $p$  found from the above mentioned equations.

The same relation among the values  $\tilde{\theta}, \varepsilon, d$  is obtained by both the above theorems.

Numerical results obtained by means of the ZAM-2 computer are given in table 2.

The values of the functions:

$$\xi = - \int_0^{\vartheta} \frac{\cos \theta}{p(\theta)} d\theta, \quad \eta = - \int_0^{\vartheta} \frac{\sin \theta}{p(\theta)} d\theta$$

and appropriate values  $\theta$  and  $\frac{p}{p_*}$ .

Table 1.

$\frac{p}{p_*}$	$\theta$	$\xi$	$\eta$
1.00	0.000	0.000	0.000
0.95	-0.165	0.003	0.000
0.90	-0.468	0.009	0.000
0.85	-0.865	0.017	0.000
0.80	-1.340	0.027	0.000
0.75	-1.887	0.039	-0.001
0.70	-2.503	0.054	-0.001
0.65	-3.187	0.072	-0.002
0.60	-3.942	0.093	-0.003
0.55	-4.772	0.118	-0.005
0.50	-5.681	0.148	-0.008
0.45	-6.678	0.184	-0.012
0.40	-7.777	0.229	-0.018
0.35	-8.993	0.285	-0.026
0.30	-10.352	0.358	-0.038
0.25	-11.893	0.454	-0.057
0.20	-13.674	0.590	-0.088
0.15	-15.802	0.797	-0.143
0.10	-18.494	1.164	-0.257
0.05	-22.352	2.060	-0.594
0.01	-28.521	6.511	-2.790
0.009	-28.820	6.990	-3.052
0.008	-29.141	7.563	-3.369
0.007	-29.490	8.270	-3.766
0.006	-29.874	9.150	-4.268
0.005	-30.301	10.319	-4.946
0.004	-30.789	11.931	-5.897
0.003	-31.365	14.309	-7.331
0.002	-32.080	18.366	-9.843
0.001	-33.079	21.234	-11.683
0.000	-37.30	$\infty$	$-\infty$

The thickness of the detonation charge depending on the deflection angle and the admissible approximation error.

Table 2.

	$\frac{p}{p}$	$\theta$	d
$\epsilon = 0.1$	0.9	- 0.46	0.047
	0.8	- 1.34	0.086
	0.7	- 2.50	0.146
	0.6	- 3.94	0.246
	0.5	- 5.68	0.426
	0.4	- 7.77	0.779
	0.3	- 10.35	1.571
	0.2	- 13.67	3.825
	0.1	- 18.50	14.824
$\epsilon = 0.01$	0.9	- 0.46	0.020
	0.8	- 1.34	0.065
	0.7	- 2.50	0.159
	0.6	- 3.94	0.338
	0.5	- 5.68	0.687
	0.4	- 7.77	1.404
	0.3	- 10.35	3.059
	0.2	- 13.67	7.846
	0.1	- 18.50	31.428
$\epsilon = 0.001$	0.9	- 0.46	0.027
	0.8	- 1.34	0.120
	0.7	- 2.50	0.316
	0.6	- 3.94	0.701
	0.5	- 5.68	1.451
	0.4	- 7.77	2.998
	0.3	- 10.35	6.569
	0.2	- 13.67	16.886
	0.1	- 18.50	67.628

#### References

- [1] J. BONDER: Ondes simples dans les écoulements compressibles plans en régime non stationnaire. Arch. Mech. Stos. T. VIII 1956, zeszyt 4.
- [2] M. BURNAT: Zadaca Cauchy dla szymajemych tečenij tipa prostych voln.
- [3] F. LABISCH: The detonation of a thin shell by a detonation wave. Referat wygłoszony na "VII Symposium on advanced problems and methods in fluid dynamics". Jurata 1965, Fluid Dynamics T. III, 1966.
- [4] R. COURANT, K.O. FRIEDRICHHS: Supersonic flow and shock waves. Interscience New York 1948.

Optimization: An introduction  
to linear quadratic programming  
by Richard W. Cottle  
and G. B. Dantzig  
Stanford University Press  
Cambridge December 1960. £1.00

This book presents a comprehensive treatment of the theory and practice of linear quadratic programming. It is intended for students and practitioners in mathematics, economics, engineering, and management science. The book is divided into two parts. The first part, which contains the basic theory, is intended for students and practitioners in mathematics, economics, engineering, and management science. The second part, which contains applications, is intended for students and practitioners in management science.

## CONTENTS

Part I. Optimization: An introduction to linear quadratic programming  
1. Introduction  
2. Linear programming  
3. Quadratic programming  
4. Nonlinear programming  
5. Applications  
6. References  
7. Bibliography  
8. Index

Part II. Optimization: An introduction to linear quadratic programming  
1. Introduction  
2. Linear programming  
3. Quadratic programming  
4. Nonlinear programming  
5. Applications  
6. References  
7. Bibliography  
8. Index

## METHODS OF OPTIMALITY

This book presents a comprehensive treatment of the theory and practice of linear quadratic programming. It is intended for students and practitioners in mathematics, economics, engineering, and management science.



DESCRIPTION OF THE PROGRAM OF  
CRITICAL PATH'S DETERMINATION

by Janusz JANOWSKI  
Józef WINKOWSKI

Received December 21th, 1965

The paper describes a computer program destined to determine critical paths. The given solution may be used particularly in PERT programs. It presents advantage, as far as computation time is concerned, due to the specific information layout, and the elimination a superfluous reviewing of the storage.

The mathematical method of the solving of the problem has been taken from the literature. But the idea of the program itself is the authors' and Andrzej Salwicki's original idea.

1. THE PROBLEM

Let  $(X, \Gamma)$  be a graph with a finite not empty set  $X$  of vertexes and the mapping  $\Gamma$ , which assigns the set  $\Gamma x \subset X$  to every  $x \in X$ . Such ordered pairs  $(x, y)$  of vertexes that  $y \in \Gamma x$  are the arcs in  $(X, \Gamma)$ . The sequence  $(x_0, \dots, x_n)$  determines the path through vertexes  $x_0, \dots, x_n$ . It is composed of arcs  $(x_i, x_{i+1})$  ( $i = 0, \dots, n-1$ ) if  $x_{i+1} \in \Gamma x_i$  for  $i = 0, \dots, n-1$ . This path begins at  $x_0$  and reaches  $x_n$ . If  $x_0 = x_n$  then the path is a cycle.

Further, we shall deal with special graphs. Let  $\Gamma^{-1}$  be a mapping which assigns to every  $x \in X$  the set  $\Gamma^{-1} x \subset X$  of the  $y$  for which  $x \in \Gamma y$ . The graph  $(X, \Gamma)$  is the network of activities if:

/1/ there exists exactly one vertex  $x \in X$ , called the starting such that  $\Gamma^{-1} x = \emptyset$  /i.e. the set  $\Gamma^{-1} x$  is empty/ ,

- /2/ there exists exactly one vertex  $\bar{x} \in X$  called final such that  $\Gamma\bar{x} = \emptyset$ ,
- /3/ there are no cycles in  $(X, \Gamma)$ ,
- /4/ the number  $l(x, y) \geq 0$  corresponds to every arc  $(x, y)$  and it is called its length.

The vertices of the network of activities will be called events, the arcs - activities, and the lengths of arcs - activities times.

The path  $(x_0, \dots, x_n)$  length is the number

$$l(x_0, \dots, x_n) = \sum_{i=0}^{n-1} l(x_i, x_{i+1})$$

and will be called the path time.

The problem we shall deal with consists in determination the longest from among paths  $\xi$  connecting the starting and the final graph events. The  $\xi$  is the longest of such paths if

$$l(\xi) = \max_{\xi} l(\xi).$$

It is named critical path.

It is not difficult to notice that the problem is correctly formulated. Indeed, because of /1/ - /3/ there exist paths from  $x$  to  $\bar{x}$  in a finite number. Evidently, there can be many critical paths. All of them determine so-called critical subgraph. Those events, which critical paths are going through, are the subgraph vertexes, and its arc are the activities belonging to the critical paths. Every path of the critical graph with the starting  $x$  and the final  $\bar{x}$  is critical in the original network of activities.

## 2. THE METHOD OF SOLUTION

As  $(X, \Gamma)$ , being the network of activities, has no cycles, a partial order can be introduced into the set of events assuming  $x \preceq y$  /i.e.  $x$  is earlier than  $y$ / if and only if there exists

a path from  $x$  to  $y$ . If the path is the arc  $(x, y)$  then  $x$  is called predecessor of  $y$  and  $y$  the successor of  $x$ .

A complete order may be introduced into the set of events, being in accordance with the above described partial order. In other words it has the property that  $x \prec y$  implicates  $x \preceq y$  /i.e.  $x$  is earlier than  $y$  in the sense of a complete order/. It is sufficient to arrange the events into a sequence according to the following rule.  $\underline{x}$  is the first element. The successive element is the one of the events which does not yet appear in the sequence, if all earlier ones in the sense of partial order appear in this sequence. The succession of elements determines the required complete order. Evidently,  $\underline{x}$  - is the first of them, and  $\bar{x}$  - the last one.

Let  $d(x)$  be for  $x \in X$ ,  $x \neq \underline{x}$  the time /i.e. the length/ of the longest path from  $\underline{x}$  to  $x$ , and let  $d(\underline{x}) = 0$ . This value will be called the earliest possible time of the event  $x$ . Reviewing the events according to the sequence determined by the complete order,  $d(x)$  can be determined for all  $x \in X$ . Namely, when  $d(x)$  has been computed for  $x \prec y$  one puts

$$d(y) = \max_{x \in \Gamma^{-1} y} (d(x) + l(x, y))$$

In such a way  $\bar{x}$  is reached and  $d(\bar{x})$  determined.  $d(\bar{x})$  is the time of the longest path from  $\underline{x}$  to  $\bar{x}$ .

Let  $e(x)$  be for  $x \in X$ ,  $x \neq \bar{x}$  the difference of  $d(\bar{x})$  and the time of the longest path from  $x$  to  $\bar{x}$ , and let  $e(\bar{x}) = d(\bar{x})$ . The value  $e(x)$  will be called the latest admissible time of the event  $x$ . It is the latest of the times, after which the event cannot occur, if the final event is to occur at its earliest possible time. Beginning from the event  $\bar{x}$ , where  $e(\bar{x}) = d(\bar{x})$ , and putting

$$e(y) = \min_{x \in \Gamma y} (e(x) - l(y, x))$$

in turn when  $e(x)$  is already known for  $y \prec x$ ,  $e(x)$  can be determined for all  $x \in X$ .

It is easy to notice that  $e(\underline{x}) = 0$  and  $d(x) \leq e(x)$ .

The equality occurs if and only if the event  $x$  is on one of the longest paths from  $\underline{x}$  to  $\bar{x}$ , i.e. - if it is the vertex of the critical subgraph. All arcs of the critical paths are the arcs of this subgraph and there are no others in it.

Thus, the critical subgraph has the set of events for which the earliest possible time is equal with the latest admissible time.

### 3. THE DESCRIPTION OF THE PROGRAM

Proceeding to the description of the program for the computer, let us assume that all events of the network of activities are denoted by natural numbers not exceeding  $\alpha$ , so that different numbers correspond to different events. Thus the events can be identified with their numbers. Every activity is determined by two events - the beginning and the end, and has a given time. To describe the network it is sufficient to give information on the activities, as follows:

List of activities

beginning	}	information on activity
end		
time		
beginning	}	information on activity
:		

These are the program input data.

The list of activities is transformed after its input, it is registered and stored. Simultaneously, two new lists are being formed of events and connections.

The list of events is of the form:

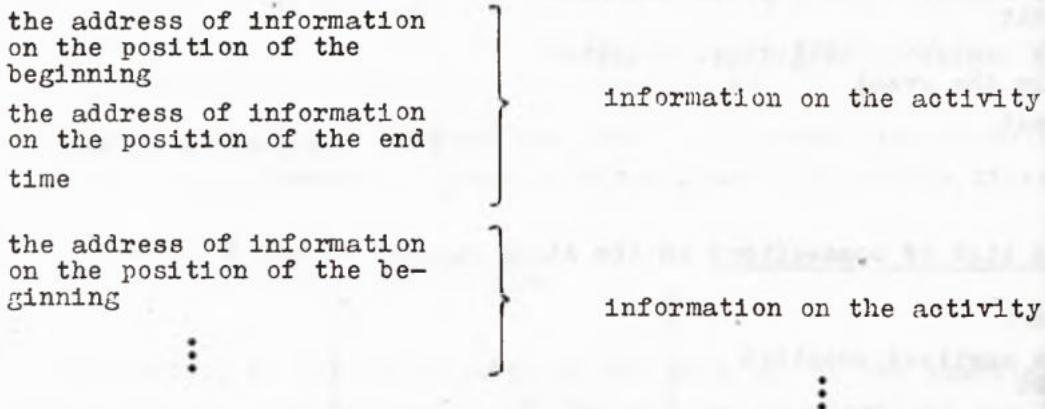
event		
the address of the information on the position of the event	}	information on the event
the number of activities reaching the event		
the number of activities starting from the event		
event	}	information on the event
:		

The list of connections in its final version is the following:

event	}	head	information on the position of the event		
the earliest possible time					
the latest admissible time					
the address of the be- ginning of the list of starting activities	}	list of reach- ing activities	information on the position of the event		
$1 +$ the address of the end of the list of starting activities					
the address of infor- mation on the reach- ing activity					
:	}	the list of starting ac- tivities	information on the position of the event		
the address of informa- tion on the starting activity					
:					
event	}	head	information on the position of the event		
:					

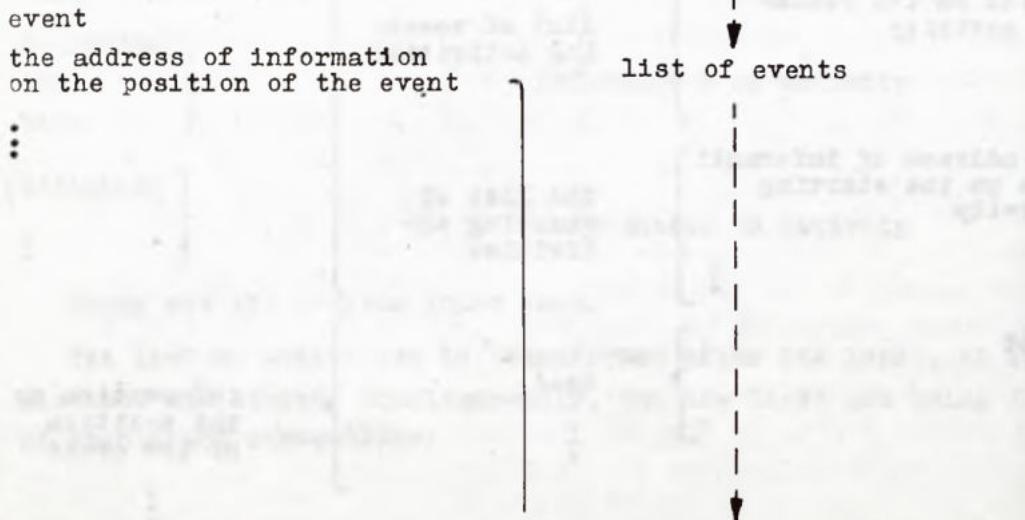
The transformation of the list of activities consists in replacing the numbers of the events by address giving information on the positions of these events in the list of connections.

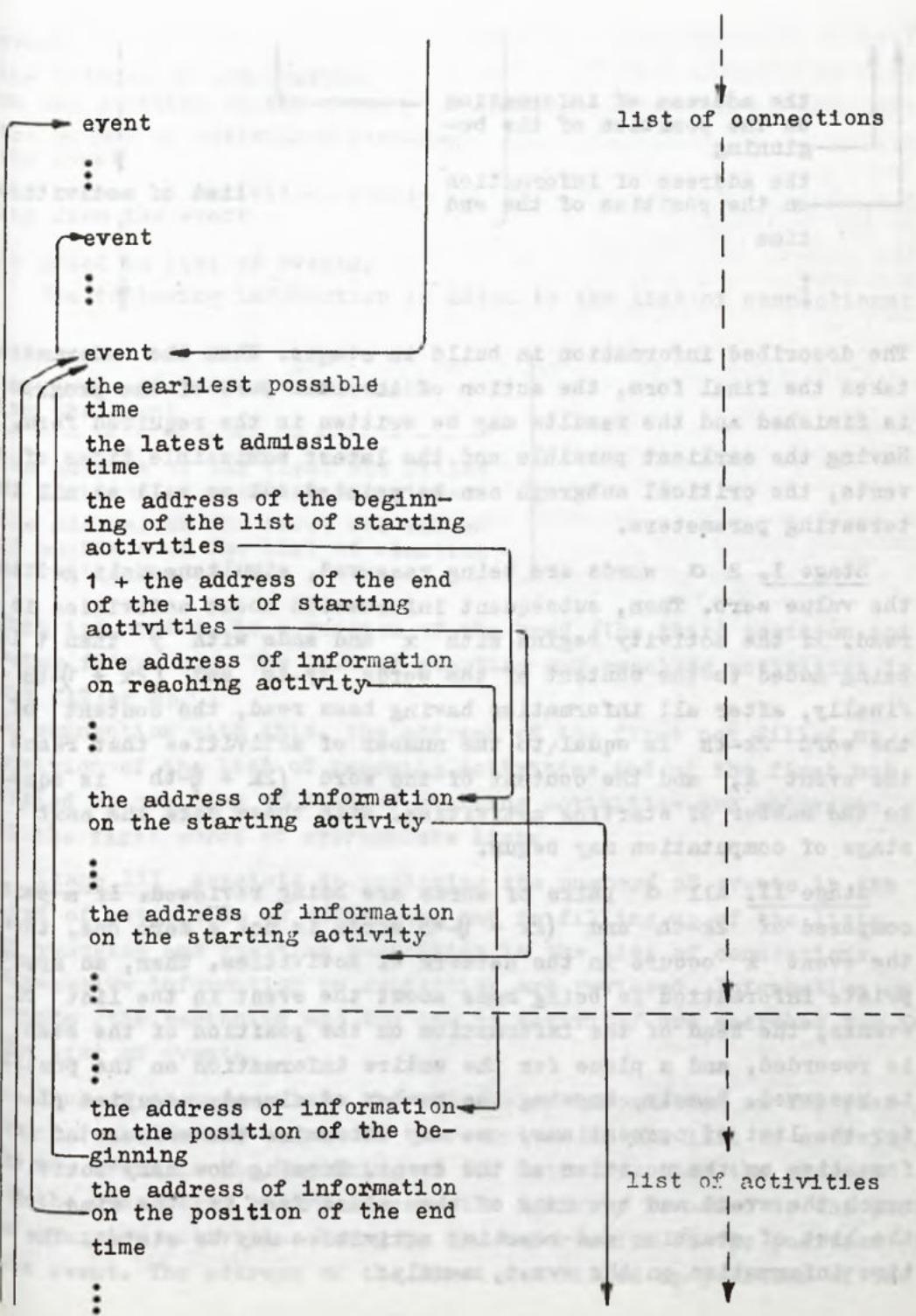
Thus it becomes the following:

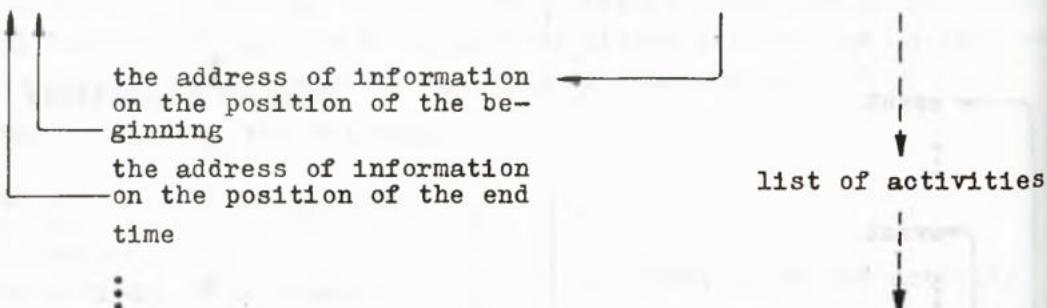


Let us assume that the smallest components of separate information are words. The address of information should be understood as the address of the information initial word.

The above given information structure can be characterized in a scheme as given below. Directed lines lead from the addresses to words having these addresses.







The described information is build in stages. When the information takes the final form, the action of the main part of the program is finished and the results may be written in the required form. Having the earliest possible and the latest admissible times of events, the critical subgraph can be pointed out as well as all interesting parameters.

Stage I.  $2 \alpha$  words are being reserved, simultaneously getting the value zero. Then, subsequent information about activities is read. If the activity begins with  $x$  and ends with  $y$  then 1 is being added to the content of the words  $2y$ -th and  $(2x + 1)$ -th. Finally, after all information having been read, the content of the word  $2k$ -th is equal to the number of activities that reach the event  $k$ , and the content of the word  $(2k + 1)$ -th is equal to the number of starting activities. With these data the next stage of computation may begun.

Stage II. All  $\alpha$  pairs of words are being reviewed. If a pair composed of  $2k$ -th and  $(2k + 1)$ -th words is not a zero one, then the event  $k$  occurs in the network of activities. Then, an appropriate information is being made about the event in the list of events, the head of the information on the position of the event is recorded, and a place for the entire information on the position is reserved. Namely, knowing the number of already occupied places for the list of connections, one may determine the address of information on the position of the event. Knowing how many activities reach the event and how many of them start from it, the size of the list of starting and reaching activities may be stated. The entire information on the event, namely:

event  
the address of information  
on the position of the event  
the number of activities reaching  
the event  
the number of activities start-  
ing from the event

is added to list of events.

The following information is added to the list of connections:

event  
the number of activities reach-  
ing the event  
-----  
the address of the first not filled  
up position of reaching activities  
the address of the first not filled  
up position of the list of starting  
activities

This information is a version of the head /the third position not being filled up/. The list of starting and reaching activities is not filled up.

In connection with this, the address of the first not filled up position of the list of reaching activities and of the first not filled up position of list of starting activities are addresses of the first words of appropriate lists.

Stage III consists in replacing the numbers of events in the list of activities by addresses, and in filling up of the lists of reaching and starting activities in the list of connections. Successive information on activities are reviewed. Information on events /the beginning and the end of activity/ are searched for in the list of events.

Each of them comprises the address of information on the position of the event, which can be rewritten in the list of activity. If the event is the beginning of the activity, then the address of the considered information on the activity is added to the list of starting activities being in the information on the position of the event. The address of the first not filled up position of the

list is simultaneously increased. The end is handled likewise, the list of reaching activities being taken into account. After the entire list of activities has been reviewed, information on the activity has the following form:

the address of information on the position of the beginning	}
the address of information on the position of the end	
time	

Information on the positions of events has the following form:

event	}
the number of activities reaching the event	
-----	
the address of the beginning of the list of starting activities	
'1 + the address of the end of the list of starting activities	}
the address of information on the reaching activity	
:	
the address of information on the starting activity	}
:	

head

list of reach-  
ing activities

list of start-  
ing activities.

The third position of the head remains empty. The searching for information on the event in the list of events is facilitated due to its lexicographical order /if such one is arranged at stage II/. Then, the information which corresponds to the given event may be found by dividing the list of about a half of it, and identifying the half, containing the information. This half is divided once more and so on, to the identification of information on the event. In order to ascertain whether the information is in the given part of the list, it is sufficient to compare the identifying number of the event with identifying numbers of the extreme events in the

above mentioned part. If this number is between them, then the information on the given event is in the segment of the list.

Stage IV consists in the previously described proper ordering of events. This is done by ordering the information in the list of events so, as to place earlier events before the later ones. For this purpose information is sought on the event having no reaching activities /initial/. This information is recorded first on the list of events. From this moment it is treated as the first actually considered one. In the information on the positions of successors the content of the second position of the head /the number of activities reaching the event/ is decreased by one. For this purpose the entire list of activities starting from the event is being reviewed. Each position of the list is the information address on the appropriate activity that contains the address on the position of the successor. This is the address of the position of the end. When the second head position of the information on the position of the successor takes the value zero, the information on the successor is being made and added to the list of events, directly after the last information recorded there during the realization of this stage. All components of the information formed may be reproduced, having the address of information on the successor position, and the information itself. The number of the reaching activities is the difference between the address of the beginning of the starting list of activities, and the address of information on the position of the event increased by 5. The number of the starting activities is the difference between the address of the end of the list of starting activities increased by 1, and the address of the beginning of the list of starting activities.

After having reviewed the entire list of starting activities, the subsequent information is being taken into account out of those which are already recorded in the list of events during the realization of the stage, but are not yet considered. Further, one proceeds similarly as for the information on the initial event. Such information always exists if the entire list of events is not filled up. It is easy to see, that in an opposite case there would be a cycle in the graph.

Such a proceeding results in a situation wherein the final event of the network is considered. Then, the entire list of events is already filled up and this is the end of the stage. From the way of proceeding it follows that the information on an event cannot be recorded in the list if the information on earlier /in the sense of a partial order/ events are not recorded. Thus, the successiveness of the appearance of information in the list determines the required complete order in the set of events. The content of all the information remains unchanged, but the heads of information on the position of events, the contents of the second position /the amount of activities reaching the event/ becomes zero.

Stage V. All information needed for writing the results is being determined. Namely, the earliest possible and the latest admissible times of events. It is assumed that the earliest possible time of the initial event is zero. Following to the end of the list, the subsequent information on events is being considered. For every event  $y$ , the earliest possible time  $d(y)$  is

$$d(y) = \max_{x \in \Gamma^{-1} y} (d(x) + l(x, y))$$

Therefore, it is sufficient to review the list of reaching activities, and adding their times to the earliest possible times of the beginnings, choose the greatest of the obtained numbers. This is the searched  $d(y)$ . The times of the reaching activities are in the information on activities, the addresses of which are in the list. The earliest possible times are in the heads of information on the position of the activity beginnings, and the addresses of the latest are in the information on the activities. These times are determined in advance, as the proceeding goes on in accordance with the complete order in the set of events, thanks to a proper arrangement of the list of events /compare stage IV/.

All events being considered, one gets the final  $\bar{x}$  and determines its earliest possible time  $d(\bar{x})$ . The latest admissible time of this event, i.e.  $e(\bar{x})$  is assumed to be  $d(\bar{x})$ . Going in the direction of the beginning of the list, subsequent information on

events are being considered. For every event  $y$  the latest admissible time  $e(y)$  is

$$e(y) = \min_{x \in \Gamma_y} (e(x) - l(y, x))$$

Thus, the list of starting activities should be reviewed, their times subtracted from the latest admissible times of the ends, and the smallest of the obtained numbers - chosen. This is  $e(y)$ . The times of starting activities are in the content of information on activities, the addresses of which are in the list.

The latest admissible times are in the heads of information on the position of the activity ends, and their addresses are in the information on activities. The above cited times are determined in advance, as the proceeding follows in a successiveness inverted as compared with a complete order in a set of events. Finally, one obtains the latest admissible time  $e(x)$  of the initial event  $x$ . Then, the latest admissible times are already determined for all events. Therefore the stage and the main part of the program are finished.

#### 4. THE PROPERTIES OF THE PROGRAM

Let  $q$  denote the number of events, and  $p$  - the number of activities in a network. Separate lists have the following numbers of positions

list of activities	$3p$
list of events	$4q$
list of connections	$5q + 2p$
total	$9q + 5p$

Indeed, every activity occupies two positions in the list of connections. Its information address is in the list of starting activities /for the beginning of the activity, and in the list of reaching activities /for the end/. The remaining estimations are beyond any doubt.

Neglecting the working positions it could be accepted that the program information requires

$$\max (2\alpha, 9q + 5p)$$

words, as in stage I  $2\alpha$  words are needed. This quantity is sufficient, because information on activities may be optionally read twice. During the first reading the information is not being registered into the storage, this is done during the second reading as then, the list of events is already made /i.e. between stages I and II/.

The stages III, IV and V of the program are most laborious and they should be first taken into consideration when estimating the time of counting. A certain information about the time gives the estimation of the number of calls to information. Calling to several positions of information is treated as one call to the information, the operations connected with registering being neglected.

In stage III there are calls to all the information in the list of activity, amounting to  $p$ . Addresses are established /for the beginning and the end/ for every activity. This requires about  $2 \log_2 q$  calls to the list of events. It gives the total of  $2p \log_2 q$  calls to this list.

In stage IV there are  $q$  calls to the list of events,  $p+q$  calls to the list of connections /each event and its successors/ and  $p$  calls to the list of activities. In stage V are twice more calls, because similar operations are twice realized.

Thus, one obtains the following table of the number of calls

	stage III	stage IV	stage V	total
list of activities	$p$	$p$	$2p$	$4p$
list of events	$2p \log_2 q$	$q$	$2q$	$3q + 2p \log_2 q$
list of connections		$p + q$	$2(p+q)$	$3(p+q)$

The described program can be easily extended. For instance, in order to obtain the PERT program, it is sufficient to supplement the information on activity with an additional position-time dispersion. The dispersion of times is determined by means of postulating the independence of activity times as random variables, and taking into account only the activities which decide about the times. These dispersions, can be registered in the fourth and fifth positions of the head of information on the position of the event. Then, the addresses, which are in them, are to be reproduced according to the information on the events.

The chosen latest admissible times can be given in advance by means of an appropriate extension of the program data.

Then, the computed times must be replaced by appropriate times, given in advance, if they are later. Critical graph should be understood as the one having the events, in which the latest admissible times of events do not exceed the earliest possible.

Many initial events may be permitted /without reaching activities/ and final /without starting activities/ in the activity network. In this case the principles of giving times change somewhat.

The network of activities can be reduced by means of pointing out chosen events and neglecting all computations corresponding to the earlier events than the ones pointed out /in the sense of a partial order/.

#### References

- [1] BERGE, C.: Theorie des Graphes et ses Applications, Paris 1958.
- [2] POSPIELOV G.S., TEJMAN A.I.: Avtomatizacija processov upravlenija razrabotkami bolsich sistem ili sloznykh kompleksov, Izvestija A.N. SSSR, Tekhnicheskaja Kibernetika No 4, 1963.



\*)  
OPTYMALIZACJA CIĘCIA TAŚMYKrzysztof MOSZYŃSKI  
Włodzimierz OSTALSKI

Pracę złożono 2.4.1966 r.

Praca podaje opis algorytmu znajdowania optymalnego rozcięcia taśmy, które w rozpatrywanym przypadku jest równoważne rozwiązaniu w liczbach całkowitych zadania programowania nielinowego.

## 1. SFORMUŁOWANIE FIZYCZNE PROBLEMU

Dany jest rulon taśmy o szerokości  $\Delta$  i długości /po rozwinięciu/  $L$ . Należy ten rulon pociąć na arkusze o zadanych szerokościach:

$$\Delta_1, \Delta_2, \dots, \Delta_n \quad /1/$$

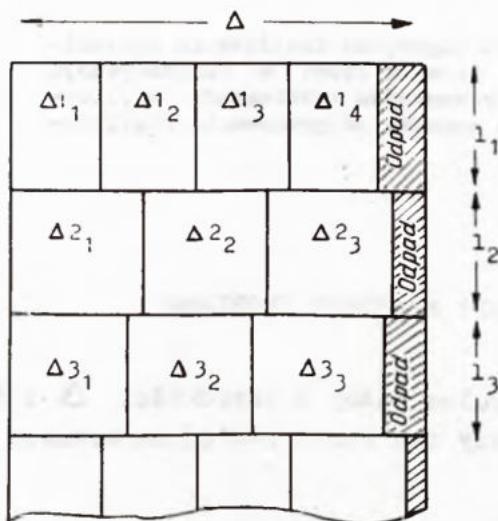
w ten sposób, żeby łączna długość arkuszy o szerokości  $\Delta_i$ , była równa  $a_i$   $i = 1, 2, \dots, n$ . Cięcie powinno odbywać się w ten sposób, żeby zużyć jak najmniejszą ilość materiału.

Będziemy tu rozpatrywać następujący sposób cięcia rulonu /por. rysunek/:

\* Niniejsza praca została podjęta z inicjatywy ministra dr M. Lesza i stanowi opracowanie artykułu [3], uzupełnione dodatkowymi uwagami, nie zawiera jednak żadnych nowych wyników.

Autorzy pragną złożyć gorące podziękowania mgr T. Mańkowi za jego istotny wkład w przygotowanie tej publikacji.

- Odcinamy arkusze o długościach  $l_1, l_2, \dots, l_N$  jednostek i o wspólnej szerokości  $\Delta$ . O liczbach  $l_i$  chwilowo nic nie zakładamy.
- Każdy z powstałych w ten sposób arkuszy o długości  $l_i$  i szerokości  $\Delta$  dzielimy wzdłuż na mniejsze arkusze, o długości  $l_{ij}$  i szerokościach  $\Delta_{ij}, \Delta_{i2}, \dots, \Delta_{ik}$ , gdzie liczby  $\Delta_{ij}$  dla  $j = 1, 2, \dots, k_i$  należą do zbioru /1/.



Problem polega na optymalnym doborze długości  $l_i$  dla  $i = 1, 2, \dots, N$  i odpowiadających im szerokości  $\Delta_{ij}, \dots, \Delta_{ik}$  ze zbioru /1/ tak, aby zużycie materiału sprowadzić do minimum, przy zachowaniu żądanego warunków cięcia.

## 2. SFORMUŁOWANIE MATEMATYCZNE PROBLEMU

Dane są liczby:

$$\Delta_1, \Delta_2, \dots, \Delta_n$$

$$a_1, a_2, \dots, a_n,$$

takie, że:

$$0 \leq \Delta_i \leq \Delta \quad i = 1, 2, \dots, n.$$

$$0 \leq a_i.$$

Mależy znaleźć:

- układ liczb  $l_1, l_2, \dots, l_N, \quad l_j \geq 0 \quad i = 1, 2, \dots, N$
- liczby całkowite nieujemne  $P_{ij} \quad i = 1, 2, \dots, n;$   
 $j = 1, 2, \dots, N,$

w ten sposób, żeby były spełnione warunki

$$\sum_{j=1}^N P_{ij} l_j = a_i \quad i = 1, 2, \dots, n \quad /2/$$

$$\sum_{i=1}^n P_{ij} \Delta_i \leq \Delta \quad j = 1, 2, \dots, N \quad /3/$$

oraz żeby wyrażenie

$$L = \sum_{j=1}^N l_j \quad /4/$$

osiągało minimum.

### 3. METODA ROZWIĄZANIA ZADANIA

Przypuśćmy, że w jakiś sposób znaleźliśmy układ liczb całkowitych nieujemnych  $P_{ij} \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, N$ , taki, że spełniony jest warunek /3/ dla  $N = n$  oraz, że macierz

$$P_0 = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix}$$

daje się odwrócić.

Zagadnienie /2/, /3/, /4/ przy tak przyjętych wartościach  $p_{ij}$ , jeśli w ogóle posiada rozwiązanie, to posiada je tylko jedno. Jest nim układ liczb  $l_i \geq 0$  otrzymanych z rozwiązania układu równań /2/.

Załóżmy dodatkowo, że rozwiązanie w tym przypadku istnieje\*. Będziemy starali się teraz znaleźć rozwiązanie lepsze od otrzymanego powyżej.

Można tego dokonać w sposób następujący:

Macierz  $P_0$  rozszerzymy przez dołączenie do niej pewnej ilości dodatkowych kolumn, powiększając jednocześnie odpowiednio ilość niezerowych współczynników /jedynek/ formy liniowej /4/. Współrzędne dodatkowych kolumn muszą być całkowite, nie ujemne, oraz muszą spełniać warunek /3/.

Rozszerzone w ten sposób zadanie jest typowym zagadnieniem programowania liniowego. Można je rozwiązywać znaną metodą simplex [1]. Nie jest obojętne jednak, jakie kolumny będziemy dołączać. Nie wystarczy bowiem spełnienie tylko warunku /3/; będziemy starali się dołączać tylko takie kolumny, które dają szansę polepszenia dotyczącego czasu otrzymanego wyniku.

Przejedźmy teraz do dokładnego opisu naszkicowanego wyżej postępowania. Będziemy na każdym kroku dołączać po jednej kolumnie, zaczynając od macierzy  $P_0$ , znajdując na każdym kroku rozwiązanie optymalne metodą simplex.

Przypuśćmy, że po pewnej ilości kroków otrzymaliśmy zagadnienie o macierzy

\*)

Przy doborze wspomnianej macierzy dobrze odwołać się do praktyki. Znane są bowiem zwykle pewne sposoby cięcia dopuszczalne, lecz nie optymalne.

$P = \begin{bmatrix} P_{ij} \end{bmatrix}$  o n wierszach i m kolumnach;  $m \geq n$ .

Tablica dla tego zagadnienia będzie postaci:

$$\left[ \begin{array}{cccc|c} P_{11} & P_{12} & \cdots & \cdots & P_{1m} & a_1 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ P_{n1} & P_{n2} & \cdots & \cdots & P_{nm} & a_n \\ \hline 1 & 1 & \cdots & \cdots & 1 & x \end{array} \right], \quad /5/$$

gdzie ostatni wiersz zawiera współczynniki minimalizowanej formy liniowej /4/.

Przypuśćmy, że rozwiązaniu optymalnemu zagadnienia /5/ odpowiada baza B.

$$B = \left[ \bar{P}_{11}, \bar{P}_{12}, \dots, \bar{P}_{1n} \right],$$

gdzie  $\bar{P}_v = \begin{bmatrix} P_{1v} \\ \vdots \\ P_{nv} \end{bmatrix}$  - kolumna nr. v macierzy P.

Utwórzmy teraz wektor

$$\bar{u}^T = \gamma^T \cdot B^{-1}, \quad /6/$$

gdzie  $\gamma^T = [1, 1 \dots, 1]$  - wektor o n współrzędnych, wszystkich równych 1.

Jak wiadomo z teorii algorytmu simplex, kryterium na to, aby rozwiązanie odpowiadające bazie B było optymalne, jest spełnienie warunków:

$$1 - \bar{u}^T \bar{P}_j \geq 0 \quad \text{dla } j = 1, 2, \dots, m. \quad /7/$$

W przypadku, gdy rozwiązanie otrzymane dla bazy B nie jest optymalne /tzn. gdy warunek /7/ nie jest spełniony dla wszystkich  $j = 1, 2, \dots, m/$ , należy znaleźć nową bazę usuwając z bazy B jedną kolumnę i wstawiając na jej miejsce inną.

Zgodnie z algorytmem simplex, jako kolumnę wprowadzaną do nowej bazy wybieramy tę, dla której wyrażenie

$$1 - \bar{u}^T \bar{P}_j$$

ma wartość najmniejszą.

Przy dołączaniu do macierzy  $P$  w /5/ nowej kolumny  $P$  będziemy starali się więc ją tak dobrać, aby wyrażenie

$$M = \bar{u}^T \bar{P}^*$$

przyjmowało wartość maksymalną.

Ścisłej, będziemy poszukiwali takich liczb całkowitych nieujemnych:

$$P_1^*, P_2^*, \dots, P_n^*$$

że

$$\sum_{i=1}^n P_i^* \Delta_i \leq \Delta$$

/8/

oraz, że wyrażenie:

$$M = \bar{u}^T \bar{P}^*; \quad \bar{P}^* = \begin{bmatrix} P_1^* \\ \vdots \\ P_n^* \end{bmatrix}$$

/9/

przyjmuje maximum.

Mogą mieć miejsce następujące przypadki:

$$/i/ \quad 1 - M_{\max} < 0$$

$$/ii/ \quad 1 - M_{\max} \geq 0.$$

Ad /i/ W tym przypadku dołączenie do macierzy  $P$  kolumny  $\bar{P}^*$  odręcznej powiadającej  $M_{\max}$  daje szansę polepszenia wyniku przy rozwiązywaniu zagadnienia programowania liniowego o rozszerzonej tablicy.

$P_{11}$	$P_{12}$	$\dots$	$P_{1m}$	$P_1^*$	$a_1$
$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$
$P_{n1}$	$P_{n2}$	$\dots$	$P_{nm}$	$P_n^*$	$a_n$
1	1		1	1	X

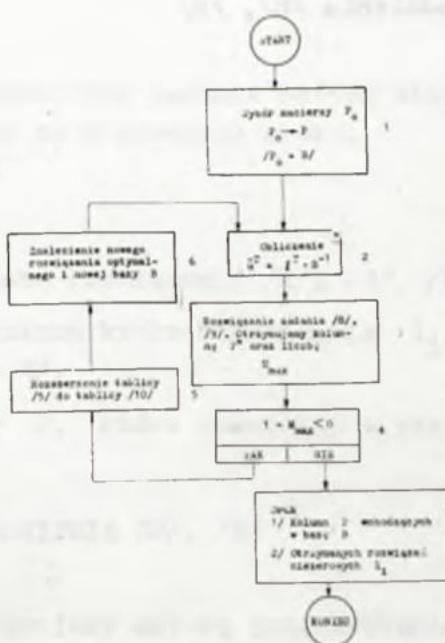
/10/

ad /ii/ W tym przypadku rozwiążaniem optymalnym rozszerzonego zagadnienia /10/, jest otrzymane uprzednio rozwiązanie zagadnienia /5/, gdyż dla wszystkich kolumn  $P$  w tablicy /10/ spełnione są warunki /7/.

Oznacza to, że znalezione rozwiązanie jest rozwiązaniem zagadnienia /2/, /3/, /4/, gdyż forma /4/ osiąga wartość minimalną.

#### 4. ALGORYTM OBLCZENIOWY

Opisany powyżej algorytm najlepiej przedstawić w postaci sieci działań:



### Opis sieci działań

#### Skrzynka 1

Tworzymy macierz kwadratową  $P_0$  zgodnie z warunkami sformułowanymi w punkcie 3. Oczywiście w tym przypadku  $P_0 = B$  jest bazą odpowiadającą rozwiążaniu optymalnemu.

Przyjmujemy  $P = P_0$ ,  $B = P_0$ .

#### Skrzynka 2

Obliczamy wektor /6/

$$u^T = \gamma^T B^{-1}$$

gdzie  $B$  jest bazą odpowiadającą aktualnie znalezionemu rozwiązaniu optymalnemu.

#### Skrzynka 3

Rozwiążujemy zagadnienie /8/, /9/

$$\Delta_1 \cdot P_1^* + \dots + \Delta_n \cdot P_n^* \leq \Delta$$

$$M = u_1 \cdot P_1^* + \dots + u_n \cdot P_n^* = \max$$

$$0 \leq P_i^* - \text{całkowite} \quad i = 1, 2, \dots, n$$

$$\text{dla } u^T = [u_1, u_2, \dots, u_n].$$

Szczegółły dotyczące rozwiązywania tego zadania zawarte są w punkcie 5.

#### Skrzynka 4

Badamy wartość  $1 - M_{\max}$ .

#### Skrzynka 5

Przechodzimy od tablicy /5/ do tablicy /10/ przez dołączenie nowej kolumny  $P^*$  oraz jedynki w ostatnim wierszu.

#### Skrzynka 6

Rozwiążujemy rozszerzone zadanie metodą simplex startując z rozwiązania otrzymanego na poprzednim kroku.

#### Skrzynka 7

Drukujemy otrzymane rozwiązanie /2/, /3/, /4/. Jest nim:

1. otrzymane na ostatnim kroku rozwiązanie  $l_{11}, l_{12}, \dots, l_{in}$   
/pozostałe  $l_j = 0/$ ,

2. kolumny macierzy  $P$ , które wchodziły w ostatnio znalezioną bazę.

#### ROZWIĄZANIE ZAGADNIENIA /8/, /9/

Zadanie to rozwiążujemy metodą programowania dynamicznego.  
Forma maksymalizowana ma postać:

$$\sum_{i=1}^n p_i u_i \quad u_i - ustalone.$$

Warunek:  $\sum_{i=1}^n p_i \Delta_i \leq \Delta \quad 0 \leq p_i - całkowite.$

Wprowadzimy ciąg funkcji  $f_1(y)$ :

$$f_1(y) = \max_{p_i} (p_i u_i)$$

$$\text{dla } p_i \Delta_i \leq y.$$

Stąd widać, że dla nieujemnych  $u_i$  ma miejsce równość:

$$f_1(y) = u_i \left[ \frac{y}{\Delta_i} \right] \quad u_i \geq 0$$

$[x]$  - część całkowita liczby  $x$ .

Natomiast dla  $u_i < 0$  -  $f_1(y) = 0$ , a więc i  $p_i = 0$ .

Zadanie nasze przyjmuje teraz postać:

$$\text{znaleźć } \max_{\{y_i\}} \sum_{i=1}^n f_1(y_i)$$

$$\text{przy warunkach: } \sum_{i=1}^n y_i \leq 0 \quad y_i \geq 0.$$

Określamy następnie rekurencyjnie ciąg funkcji  $F_k(z)$  w sposób następujący:

$$F_1(z) = f_1(z)$$

$$F_{k+1}(z) = \max_{0 \leq y \leq z} [f_{k+1}(z) + F_k(z - y)] \quad k = 1, 2, \dots, n-1.$$

Wszystkie funkcje  $F_k(z)$  określamy w całym przedziale  $0 \leq z \leq \Delta$ , tworząc tablicę:

$z$	$F_1$	$y_1(z)$	$F_2$	$y_2(z)$			$F_n$ ( $z$ )	$y_n(z)$
...	...	...	...	...		...	...	...
...	...	...	...	...		...	...	...
$\Delta$								

$y_{k+1}(z)$  - wartość  $y$  dla której ma miejsce równość /11/

Po otrzymaniu tablicy wszystkich funkcji  $F_k(z)$  / $k=1, 2, \dots, n$ / wyznaczamy przez przeszukiwanie ciąg argumentów tworzący wg terminologii R. Bellmana - strategię optymalną [2], [4].

Strategię optymalną wyznaczamy wg następujących wzorów:

$$z_n - \text{takie, że } F_n(z) = \max F_n(z)$$

$$y_n = y_n(z_n).$$

Wartości te znajdujemy z tablicy.

Następnie:

$$z_{n-1}: F_{n-1} z_{n-1} = \max_{0 \leq z \leq z_n - b_i} F_{n-1}(z)$$

$$b_i = \sum_{k=1}^{n-1} y_{n-k}$$

$$\bar{y}_{n-1} = y_{n-1}(z_{n-1}).$$

Ciąg  $\bar{y}_1, \bar{y}_2, \dots, \bar{y}_n$  jest poszukiwaną strategią.

Z własności funkcji  $F_k(z)$  wynika, że:

$$z_n = \Delta, \quad \bar{y}_n = y_n(\Delta), \quad z_{n-1} = \Delta - \bar{y}_n \quad /12/$$

$$z_{n-1} = \Delta - \sum_{k=0}^{i-1} \bar{y}_{n-k}.$$

Oczywiście w praktyce należy korzystać z wzorów /12/.

Mając wyznaczoną strategię optymalną - znajdujemy wartości  $P_1^*$ , które należy dołączyć do macierzy /5/:

$$\overline{P_1^*} = \left[ \begin{array}{c} \bar{y}_1 \\ 1 \end{array} \right].$$

#### Literatura

- [1] DANTZIG G.B.: Linear Programming and Extensions, Princeton University Press 1963.
- [2] BELLMAN R.: Dynamic Programming, Princeton 1957.
- [3] MEDNICKIJ B.G.: Optimalnyj raskroj polosy, Ekonomika i Matematicheskie Metody. T. I wyp. 2, 1965.
- [4] BELLMAN R.E., DREYFUS S.E.: Applied Dynamic Programming, Princeton University Press, 1962.

#### OPTIMIZATION OF STRIPE CUTTING

#### Summary

The paper refers to a method of solution of certain problems of stripe cutting, described by B.G. Mednickij in [3], and contains no new results. We give this algorithm in a more convenient form for users of computers. Mednickij's algorithm gives the solution in finite number of steps, starting with any feasible solution. Each step consists of a solution of two linear programs. The first is solved by simplex method, the second by dynamic programming method.

ROZWIĄZANIE ZADANIA TRANSPORTOWEGO  
NA MASZYNE MATEMATYCZNEJ

Jan SZMELTER  
Bogdan GADEK

Pracę złożono 15.3.1966 r.

W artykule przedstawiono metodę rozwiązywania typowego zadania transportowego, szczególnie przydatną do wykorzystania przy użyciu elektronicznej maszyny cyfrowej, gdzie duża macierz kosztów musi być przechowywana w pamięci pomocniczej, zaś rozwiązanie bazowe w pamięci operacyjnej. Pokazana metoda wymaga minimalnej ilości miejsc roboczych oraz daje krótki program dzięki specjalnej numeracji zmiennych /patrz § 2.2/. Program odznacza się dużą szybkością działania, co udało się uzyskać przez takie porządkowanie zmiennych w każdej iteracji aby uniknąć czasochłonnego szukania "ścieżki zastępczej". Dodatkową korzyścią jest uniknięcie oddzielnego rozpatrywania rozwiązań zdegenerowanych.

1. WSTĘP

Zadanie transportowe można sformułować w następujący sposób:

w ( $m$ ) magazynach o numerach  $0, 1, 2, \dots, i, \dots, m-1$  znajduje się jednorodny produkt w ilościach odpowiednio  $X_0, X_1, X_2, \dots, X_i, \dots, X_{m-1}$  jednostek; jest ( $n$ ) odbiorów o numerach  $m, m+1, m+2, \dots, k, \dots, m+n-1$  o zapotrzebowaniach  $X_m, X_{m+1}, X_{m+2}, \dots, X_k, \dots, X_{m+n-1}$  jednostek; zakładamy, że suma zapasów produktu w magazynach jest równa sumie zapotrzebowan; znamy ceny jednostkowe transportu z dowolnego magazynu ( $i$ ) do dowolnego odbiorcy ( $k$ ) -  $c_{ik}$  ( $i=0, 1, 2, \dots, m-1; k=m, m+1, m+2, \dots, m+n-1$ ). Znaleźć taki rozkład przewozów -  $X_{ik}$  ( $i=0, 1, 2, \dots, m-1; k=m, m+1, m+2, \dots, m+n-1$ ), aby zapotrzebowania wszystkich odbiorów zostały pokryte i aby łączny koszt transportu był możliwie najmniejszy;

lub matematycznie:

zminimalizować funkcję celu

$$F = \sum_{i=0}^{m-1} \sum_{k=m}^{m+n-1} x_{ik} c_{ik}, \quad /1/$$

zachowując warunki

$$\sum_{i=0}^{m-1} x_{ik} = x_k \quad \text{dla } k=m, m+1, \dots, m+n-1 \quad /2/$$

$$\sum_{k=m}^{m+n-1} x_{ik} = x_i \quad \text{dla } i=0, 1, \dots, m-1 \quad /2/$$

$$x_{ik} \geq 0 \quad \text{dla wszystkich } i \text{ oraz } k, \quad /3/$$

gdzie  $C$  jest daną macierzą, a  $X$  danym wektorem, przy czym zachodzi:

$$\sum_{i=0}^{m-1} x_i = \sum_{k=m}^{m+n-1} x_k. \quad /4/$$

Z teorii programowania liniowego [1] wiadomo, że:

1. zadanie transportowe posiada rozwiążanie dopuszczalne, t.zn. takie, dla którego spełnione są warunki /2/ i /3/;
2. wobec zależności /4/ mamy tylko  $(m+n-1)$  niezależnych warunków /2/, zatem bazowe rozwiązanie dopuszczalne zawiera co najwyżej  $(m+n-1)$  niezerowych wartości  $x_{ik}$ ;
3. rozwiązanie, dla którego funkcja celu osiąga minimum/rozwiążanie optymalne/ znajduje się w zbiorze rozwiązań bazowych.

Zadanie transportowe rozwiązuje się na ogólną metodą iteracyjną. Polega ona na znalezieniu dowolnego rozwiązania bazowego, a następnie poprawieniu go w ten sposób, aby po pewnej ilości krokówtrzymać rozwiązanie optymalne.

## 2. OPIS PROGRAMU ROZWIĄZUJĄCEGO ZADANIE

### 2.1. Ogólny schemat programu

W Ośrodku Obliczeniowym przy Katedrze Mechaniki Technicznej Politechniki Łódzkiej prowadzono badania nad ułożeniem takiego programu, który by pozwolił zrealizować znane rozwiązanie Dansiga w sposób szczególnie przydatny do wykonywania obliczeń przy użyciu maszyny liozącej. Ułożony według niego program dla maszyny ZAM-2 Beta daje możliwość rozwiązywania zadań o rozmiarach  $(m+n) \leq 200$ . Program ten zajmuje ok. 120 miejsca pamięci i odznacza się dużą szybkością działania.

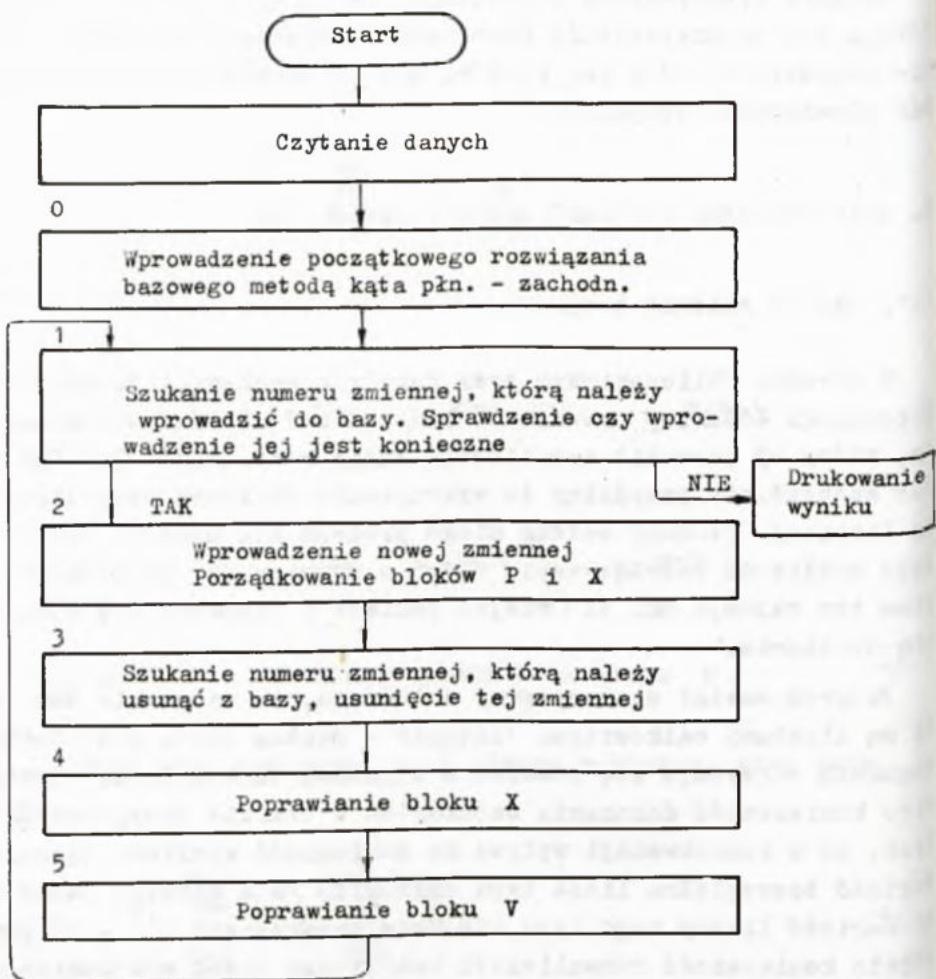
Program został ułożony przy założeniu, że wszystkie dane wielkości są liczbami całkowitymi /integer - według opisu ALGOLU-60/, rozwiązań otrzymuje się również w liczbach całkowitych. Powstaje więc konieczność dokonania zaokrągleń w trakcie przygotowania danych, co w konsekwencji wpływa na dokładność wyników. Ograniczona wartość bezwzględna liczb typu całkowite /dla maszyny ZAM-2 bezwzględna wartość liczby tego typu nie może przekraczać  $2^{17} - 1$ / powoduje często konieczność normalizacji tabeli cen przed wykonaniem obliczeń. Normalizację można przeprowadzić dodając do wszystkich elementów kolumny lub wiersza pewną liczbę całkowitą. Na rys. 1 pokazano ogólny schemat omawianego programu.

### 2.2. Rezerwacja miejsc w pamięci

Program rezerwuje miejsca w następujących blokach:

1. Blok  $X = X_0, X_1, X_2, \dots, X_{m-1}, X_m, X_{m+1}, \dots, X_{m+n-1}$ .

Na poczatk u obliczeń blok  $X$  zawiera dane wartości zapasów w magazynach  $0, 1, 2, \dots, m-1$  oraz zapotrzebowania odbiorów  $m, m+1, \dots$



Rys. 1. Ogólny schemat programu transportowego. Numery odpowiadają numerom na schematach szczegółowych.

$m+2, \dots, m+n-1$ . W trakcie liczenia tworzone są w nim kolejne rozwiązania bazowe, a na końcu rozwiązanie optymalne. Ponieważ rozwiązanie bazowe posiada tylko  $m+n-1$  niezerowych wartości  $X_{ik}$ , jedno miejsce bloku  $X$  jest niewykorzystane. Indeks tego miejsca jest pamiętany jako wartość zmiennej  $t$ .

2. Blok indeksów  $P = P_0, P_1, \dots, P_{m-1}, P_m, P_{m+1}, \dots, P_{m+n-1}$

Blok  $P$  zawiera indeksy wskazujące aktualne rozwiązanie bazowe według zasad:

$$x_{ik} = X_{1P_1} = X_1 \quad \text{dla } i=0,1, \dots, m-1 \quad \text{oraz } i \neq t$$

$$x_{ik} = X_{P_k k} = X_k \quad \text{dla } k=m, m+1, \dots, m+n-1 \quad \text{oraz } k \neq t .$$

/5/

3. Blok pomocniczy  $V = V_0, V_1, \dots, V_{m-1}, V_m, V_{m+1}, \dots, V_{m+n-1}$

W bloku  $V$  pamiętane są liczby służące do sprawdzania, czy dane rozwiązanie jest optymalne /potenowały/.

4. Blok jednostkowych kosztów transportu  $C$  -

$$c_{0,m} \quad c_{0,m+1} \quad \dots \quad c_{0,m+n-1}$$

$$c_{1,m} \quad c_{1,m+1} \quad \dots \quad c_{1,m+n-1}$$

\*\*\*      \*\*\*      \*\*\*      \*\*\*

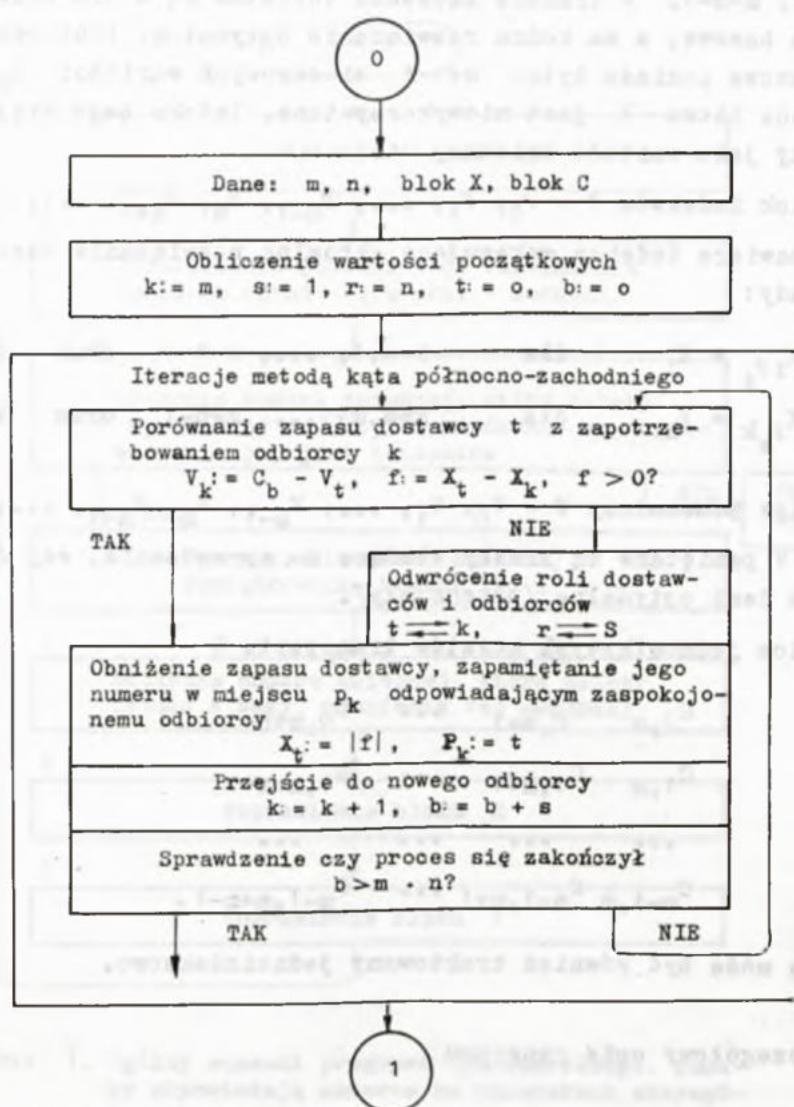
$$c_{m-1,m} \quad c_{m-1,m+1} \quad \dots \quad c_{m-1,m+n-1} .$$

Blok ten może być również traktowany jednoindeksowo.

## 2.3. Szczegółowy opis programu

### 2.3.1. Wprowadzenie rozwiązania poczatkowego metodą "kąta północno-zachodniego".

Wstępne rozwiązanie bazowe zdecydowano wprowadzić metodą "kąta północno-zachodniego" przystosowaną do przyjętego sposobu zapisu informacji w pamięci maszyny. Próby przeprowadzone w naszym ośrodku wykazały, że stosowanie innych znanych z literatury [1] metod wpro-



Rys. 2. Wprowadzenie początkowego rozwiązania bazowego metodą kąta północno-zachodniego. Sieć działań.

wadzania rozwiązania początkowego nie daje spodziewanych korzyści, a powoduje konieczność pewnej rozbudowy programu.

Rys. 2 pokazuje sieć działań omawianej części programu. Na rysunku tym, podobnie jak na wszystkich następnych przedstawiających schematy działań, podano przy każdej czynności szczegółowe komentarze słowne omawiające jej sens oraz wzory, według których wykonywane są działania w maszynie. Symbol := oznacza operację nadania wartości i sens jego jest identyczny jak w ALGOLU 60. Symbol  $\leftarrow$  oznacza zamianę wartości zmiennych występujących po jego obu stronach. W wyniku naznaczonych działań otrzymujemy w blokach X i P oraz w komórce t wyczerpujące informacje dotyczące początkowego rozwiązania bazowego. W bloku V zaś, układ liczb posiadających następującą własność:

jeśli utworzymy nową tabelę cen  $C'$  w ten sposób, że

$$C'_{ik} = C_{ik} - v_1 - v_k. \quad /6/$$

dla  $i=0, 1, 2, \dots, m-1; k=m, m+1, m+2, \dots, m+n-1$ , to funkcja celu dla początkowego rozwiązania i dla tak przekształconej macierzy cen będzie równa zeru:

$$F' = \sum_{i=0}^{m-1} \sum_{k=m}^{m+n-1} C'_{ik} x_{ik} = 0. \quad /7/$$

Latwo dowieść, że optymalne rozwiązanie odpowiadające macierzy  $C'$  jest identyczne z optymalnym rozwiązaniem dla macierzy  $C$  [2].

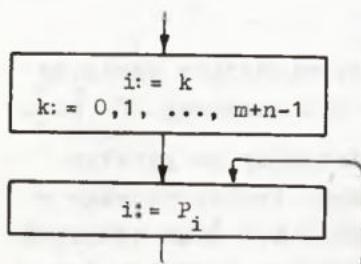
Powstawanie rozwiązania początkowego przedstawimy na prostym przykładzie. Na rys. 3 pokazano dane dla zadania transportowego w przypadku istnienia dwóch dostawców o numerach 0 i 1 oraz czterech odbiorców o numerach 2, 3, 4 i 5, na rys. 4 kolejne iteracje tworzenia rozwiązania startowego dla tych danych. Wszystkie działania zostały wykonane według schematu z rys. 2, co Czytelnik może łatwo prześledzić. W poszczególnych kolumnach tabeli umieszczono wartości bloków P, X i V dla kolejnych iteracji, ostatnie kolumny odpowiadają rozwiązaniu startowemu. W trakcie obliczeń tworzą się licz-

Zapotrzebowania				
Nr	Ceny			
0	5	7	6	1
1	14	16	10	8
	2	3	4	5

i	Wartości początkowe	Iteracja				Wartości końcowe
		1	2	3	4	
Blok P						
0	-	-	-	4	4	4
1	-	-	-	-	-	-
2	-	0	0	0	0	0
3	-	-	0	0	0	0
4	-	-	-	-	1	1
5	-	-	-	-	-	1
Blok X						
0	60	40	10	<b>10</b>	<b>10</b>	<b>10</b>
1	40	40	40	40	10	0
2	20	<b>20</b>	<b>20</b>	<b>20</b>	<b>20</b>	<b>20</b>
3	30	30	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>
4	40	40	40	<b>30</b>	<b>30</b>	<b>30</b>
5	10	10	10	10	10	<b>10</b>
Blok V						
0	0	0	0	0	0	0
1	-	-	-	-	4	4
2	-	5	5	5	5	5
3	-	-	7	7	7	7
4	-	-	-	6	6	6
5	-	-	-	-	-	4
k t						
k		<b>2</b>	3	4	1	5
t		0	0	0	4	1

Rys. 3. Przykład zadania transportowego.



Rys. 5. Schemat przeszukiwania rozwiązania na podstawie wartości bloku p.

Rys. 4. Powstawanie rozwiązania początkowego. W kolumnach bloku X podkreślono liczby już w danej iteracji należące do rozwiązania.

by k i t zapisane u dołu tabeli, oznaczające numery dostawców i odbiorców kojarzonych w danej iteracji. Po skończeniu iteracji otrzymuje się wartości końcowe /zapisane na rys. 4 w ostatniej kolumnie/. Wartości te stanowią początkowe rozwiązanie bazowe, od którego zaczynamy następny etap obliczeń.

Powstałe rozwiązanie startowe ma następujące własności:

1. Nie ma w nim zamkniętych pętli, oznacza to, że zmieniając wartości zmiennej i według schematu rekurencyjnego z rys. 5 zawsze natrafimy na wartość  $i=t$ . Dowiedzono, że jeśli dla danego zadania istnieje rozwiązanie zawierające zamkniętą pętlę, można dla niego utworzyć rozwiązanie nie zawierające pętli, dla którego funkcja celu ma tę samą lub mniejszą wartość; wynika stąd, że rozwiązania optymalnego możemy szukać wśród rozwiązań dopuszczalnych nie zawierających pętli [2].

2. Rozwiązanie to można przedstawić graficznie /dla przykładu rozwiązania z rys. 6 odpowiedni wykres pokazano na rys. 7/, w tym celu należy wypisać wszystkie numery dostawców i odbiorców, a następnie połączyć je strzałkami tak, aby każda strzałka zaczynała się od numeru  $i /i=0, 1, \dots, m+n-1/$ , a kończyła na odpowiadającym mu numerze  $v_i$ . Istnieje wzajemna odpowiedniość pomiędzy takim wykresem, a danym rozwiązaniem dopuszczalnym. Dlatego przekształcenia dokonywane na rozwiązaniu można demonstrować wykresem.

2.3.2. Szukanie zmiennej, którą należy wprowadzić do bazy ( $x_{rs}$ ); sprawdzenie czy wprowadzenie jej jest jeszcze konieczne.

Dla rozwiązania bazowego należy wprowadzić zmienną o indeksach tych samych, które posiada najmniejszy element macierzy  $C'$  (6). Gwarantuje to monotoniczne zbliżanie się do rozwiązania optymalnego. W przypadku zaś, gdy zachodzi zależność:

$$f = C'_{rs} = \min_{i,k} (C'_{ik} - v_i - v_k) \geq 0 \quad /8/$$

dla  $i=0, 1, 2, \dots, m-1; \quad k=m, m+1, m+2, \dots, m+n-1$

dane rozwiązanie jest optymalne [2].

Zapotrzebowania

20	30	40	10
----	----	----	----

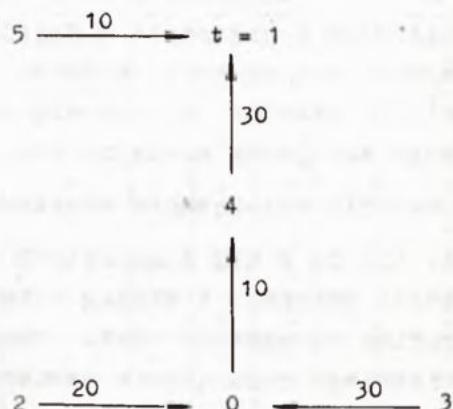
Nr Przesyłki

0	20	30	10	
1			30	10
	2	3	4	6

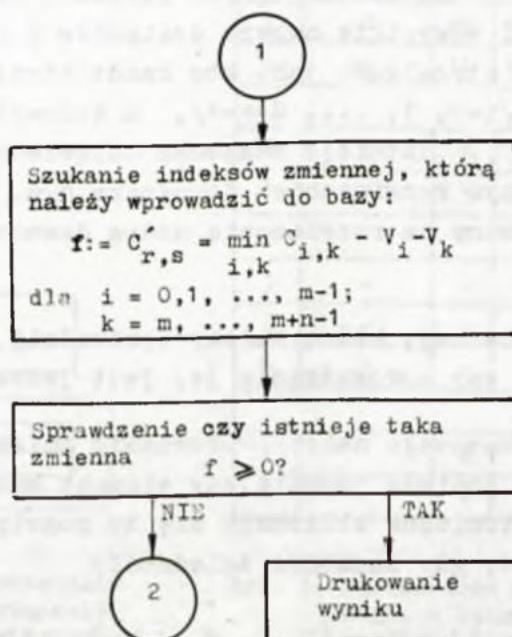
Zapa-sy

60
40

Rys. 6. Sprawdzenie poprawności rozwiązań otrzymanego programem z rys. 2.  
Sumy wartości przesyłek w wierszach są równe zapasom, w kolumnach zapotrzebowaniom.



Rys. 7. Graficzne przedstawienie rozwiązania dopuszczalnego zadania transportowego. Liczby przy strzałkach oznaczają wartości przesyłek.



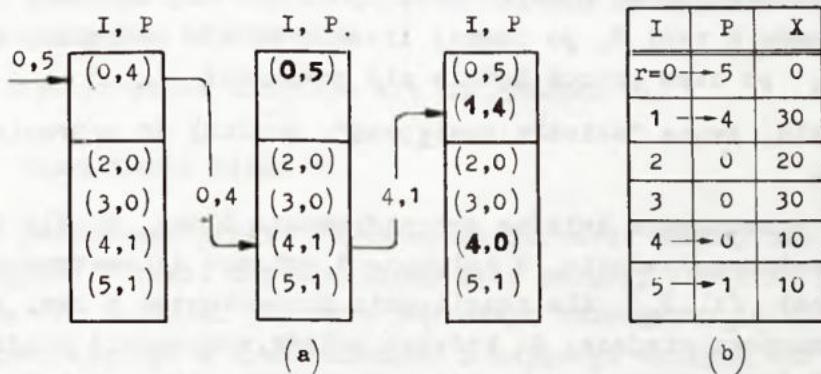
Rys. 8. Szukanie indeksów zmiennej, którą należy wprowadzić do bazy.

$v_k$				
Nr	$c_{i,k}$			
	5	7	6	4
0	0	0	0	-3
1	+5	+5	0	0
	2	3	4	5

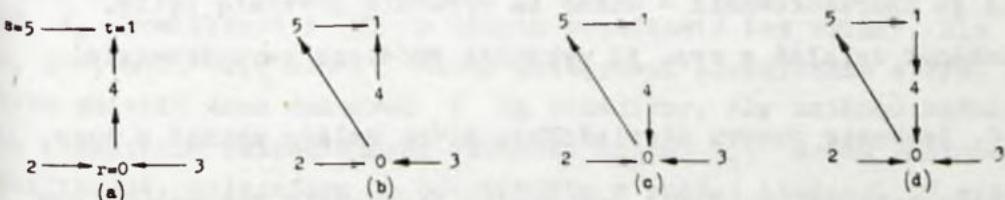
  

$v_i$	
0	
4	

Rys. 9. Do bazy należy wprowadzić zmienną o numerach  
 $r = 0$ ,  $s = 5$ ;  
 $f = \min(c_{i,k}) = -3$ .



Rys. 10. Schemat porządkowania bloku P(a); ostateczny wynik porządkowania (b).



Rys. 11. Graficzne przedstawienie tworzenia pętli podczas porządkowania bloku P. a - rozwiązanie początkowe. Porównaj z rys. 10.

Rys. 8 przedstawia sieć działań części programu, wykonującej omówione sprawdzenie, rys. 9 - wynik obliczeń dla zamieszczonego przykładu, wykonanych według sieci działań z rys. 8.

### 2.3.3. Wprowadzenie nowej zmiennej; porządkowanie bloków X i P.

Zmienną  $X_{rs}$  musimy wprowadzić do bloku  $X$  w miejscu o numerze  $r$ . W tym miejscu może jednak być inna zmienna  $X_{rP_r}$ , musimy ją zatem przenieść w miejsce o numerze  $P_r$  itd. aż do miejsca o numerze  $t$ , które jest wolne, o czym wspomniano wyżej. Wartość wprowadzanej zmiennej wynosi na tym etapie zero. W ten sposób otrzymamy nowe uporządkowanie bloków  $X$  i  $P$ , odznaczające się własnością:

jeżeli, poczynając od numeru  $k=r$ , poruszać się będziemy według schematu z rys. 5, po pewnej liczbie kroków natrafimy na numer  $i=r$ , po czym proces będzie się powtarzał.

Powstała pętla, zwana "ścieżką zastępczą", posłuży do poprawiania rozwiązania.

Rys. 10a przedstawia kolejne uporządkowania bloku  $P$  dla poprzednio omawianego zadania. W kolumnie 1 wpisano schematycznie skojarzenia liczb  $(i, P_i)$  dla rozwiązania początkowego z rys. 4, strzałką oznaczono miejsce, do którego należy wprowadzić skojarzenie  $(0,5)$ . Kolejne kolumny pokazują poszczególne etapy działania; ostatnia kolumna pokazuje ostateczne uporządkowanie. Jednocześnie z porządkowaniem bloku  $P$ , dokonywane są odpowiednie przestawienia w bloku  $X$ . Rys. 11 pokazuje graficznie przebieg porządkowania. Na rys. 11a widać rozwiązywanie początkowe, rys. 11d przedstawia rozwiązanie po uporządkowaniu - widać tu wyraźnie powstałą pętlę.

Schemat działań z rys. 12 wykonuje omówione porządkowanie.

### 2.3.4. Szukanie numeru niewiadomej, którą należy usunąć z bazy.

Łatwo wykazać, że jeśli poruszając się według schematu z rys. 5, poczynając od numeru  $k=r$  będziemy na przemian zwiększać i zmniejszać wartości  $X_1$  o stałą liczbę, aż do osiągnięcia ponownie wartości  $i=r$ , otrzymamy nowe rozwiązanie, dla którego będą spełnione

ne warunki /2/. Można zatem dobrać taką liczbę  $g$ , aby przynajmniej jedna ze zmiennych, należących poprzednio do rozwiązania  $X_{1P_i}$  została wyzerowana i aby spełnione były warunki /4/, a następnie usunąć tę zmienną, przyjmując  $t=1$ . Dla rozwiązania z rysunku 10b i rys. 11d  $g=10\%$ .

Prosty program przedstawiony na rys. 13 realizuje powyższe zadanie.

### 2.3.5. Poprawianie bloku X

Mając wartość  $g$ , przebiegamy jeszcze raz pętlę zmieniając odpowiednio wartość  $X_1$ .

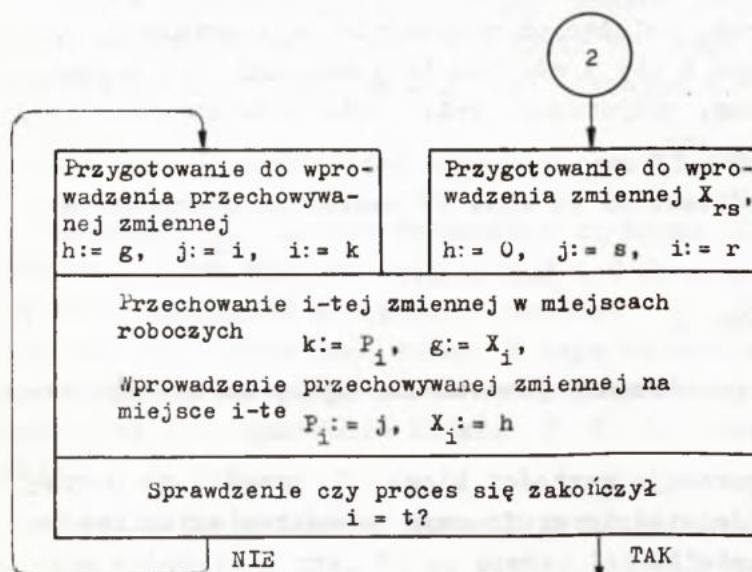
Tabela na rys. 14 pokazuje wartości bloku  $X$  przed i po poprawieniu; rys. 15 jest odpowiednim graficznym przedstawieniem operacji wykonanych w tej tabeli.

Schemat programu znajduje się na rysunku 16.

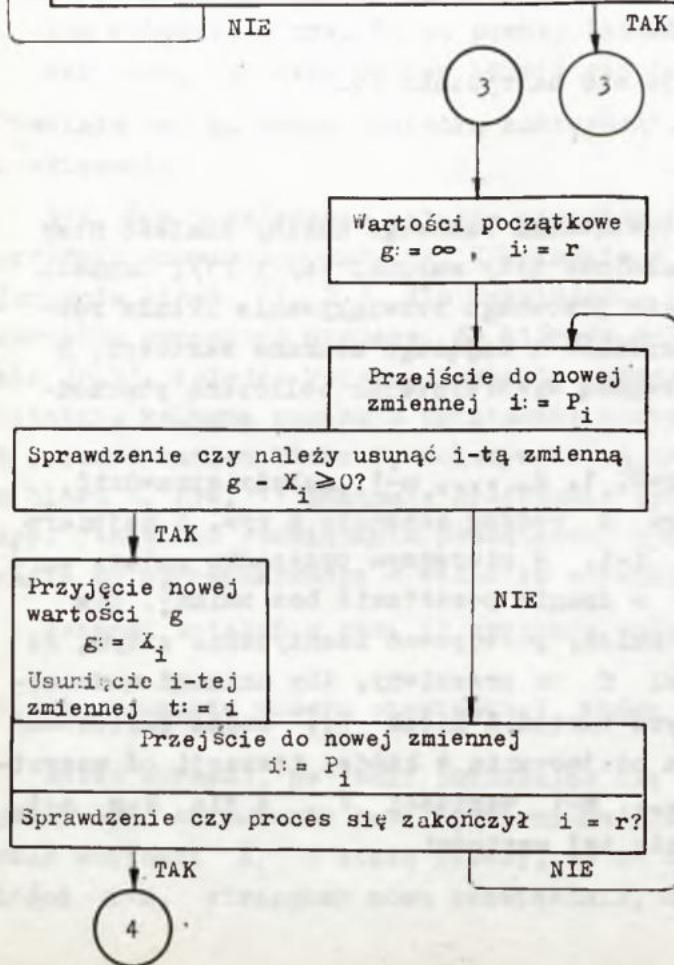
### 2.3.5. Poprawienie bloku V

Dla powstałego nowego rozwiązania bazowego musimy znaleźć nowy układ liczb  $V$  taki, aby spełnione były warunki /6/ i /7/. Okazało się, że nie zachodzi potrzeba ponownego rozwiązywania układu równań, wynikającego z tych warunków i dającego szukane wartości, a można posłużyć się prostą regułą wykorzystując obliczoną poprzednio wartość  $f = C'_{rs}$ .

Aby poprawić  $V_k$  dla  $k=0, 1, 2, \dots, m-1$  należy sprawdzić, czy poruszając się od numeru  $k$  według schematu z rys. 5 najpierw natrafimy na  $i=r$ , czy na  $i=t$ . W pierwszym przypadku należy wartość  $V_k$  powiększyć o  $f$ , w drugim pozostawić bez zmiany. Dla  $k=m, m+1, m+2, \dots, m+n-1$  należy postępować identycznie z tym, że trzeba zmienić znak zmiennej  $f$  na przeciwny. Aby uniknąć nadmiernego wzrastania bezwzględnych wartości liczb  $V_k$ , można zastosować normalizację, polegającą na odejmowaniu w każdej iteracji od wszystkich  $V_k$  dla  $k=0, 1, 2, \dots, m-1$  wartości  $V_0$ , a dla  $k=m, m+1, m+2, \dots, m+n-1$  – dodawaniu tej wartości.

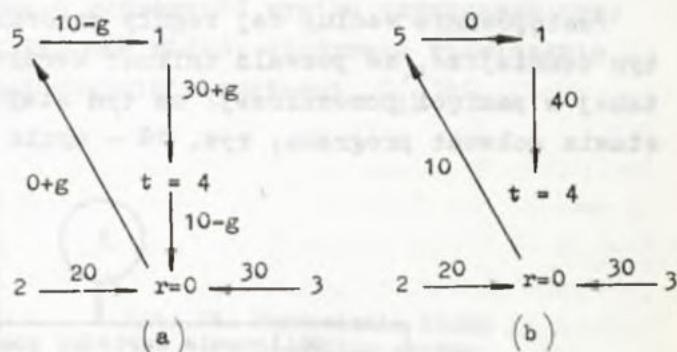


Rys. 12. Wprowadzenie nowej zmiennej, porządkowanie bloków  $X$  i  $P$

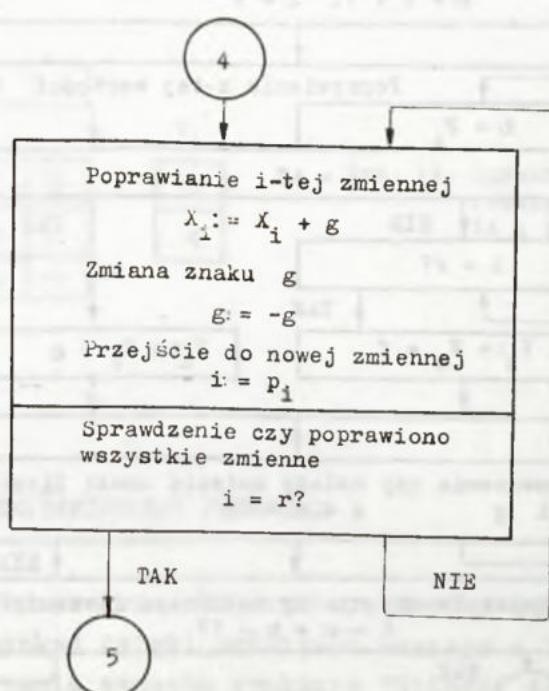


Rys. 13. Szukanie numeru zmiennej, która należy usunąć z bazy; usunięcie tej zmiennej.

P	Blok X		
r=0	5	0+g	0+10
1	—→ 4	30+g	30+10
2	0	20	20
3	0	30	30
t=4	—→ 0	10-g	10-10
5	—→ 1	10-g	10-10

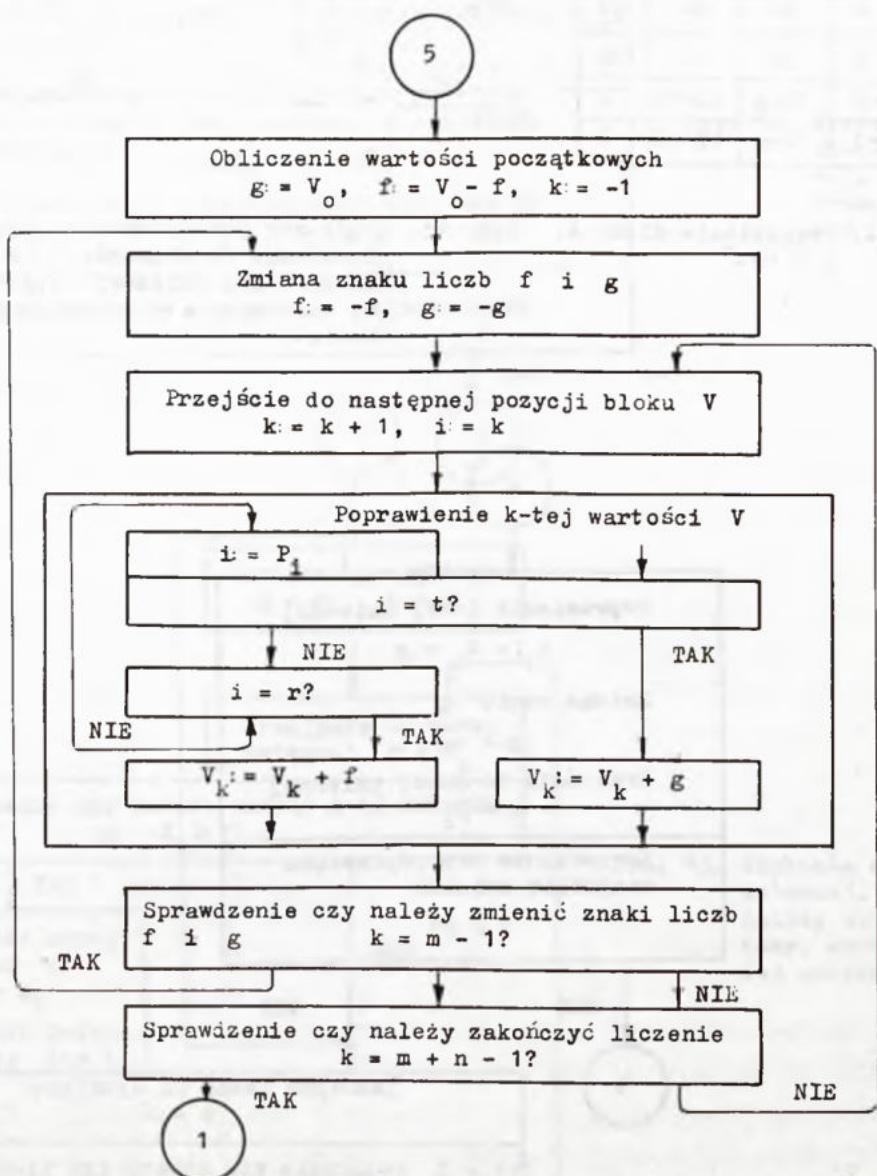
Rys. 14. Poprawianie Bloku X; Rys. 15. Graficzne przedstawienie powstania nowego rozwiązania - (a) szukanie wartości zmiennej  $g$  ( $g = 10$ ).

Rys. 15. Graficzne przedstawienie powstania nowego rozwiązania - (a) szukanie wartości zmiennej  $g$  ( $g = 10$ ),  
(b) rozwiązanie po dokonaniu poprawienia.



Rys. 16. Schemat programu służącego do poprawiania bloku X.

Postępowanie według tej reguły znacznie upraszcza program i jest tym cenniejsze, że pozwala uniknąć korzystania z tabeli cen /pamiętanej w pamięci pomooniczej/ na tym etapie obliczeń. Rys. 17 przedstawia schemat programu, rys. 18 – wynik obliczeń dla rozpatrywane-



Rys. 17. Schemat programu tworzącego nowy układ liczb  $V_i$ .

go przykładu. Rys. 19 pokazuje poprawność wyniku uzyskanego przy zastosowaniu omawianej reguły. Jak widać, otrzymane rozwiązanie jest już optymalne, ponieważ wszystkie wartości  $c'_{ik} \geq 0$ .

$$g = 0 \quad f = -3$$

	P	Blok V		
		a	b	c
r=0	5	0	0+f	-3
1	4	4	4+g	4
2	0	5	5-f	8
3	0	7	7-f	10
t=4	-	6	6-g	6
5	1	4	4-8	4

Rys. 18. Poprawianie bloku V; według schematu z rys. 17.

- (a) wartości początkowe,
- (c) wartości poprawione.

$V_k$				
$c'_{i,k}$				
8	10	6	4	
0	0	0	3	0
1	2	2	0	0
	2	3	4	5

$V_i$	
-3	
	4

Rys. 19. Sprawdzenie poprawności rozwiązania z rys.rys. 14 i 18.

### 3. ZALETY PRZEDSTAWIONEGO PROGRAMU

Przedstawiony program odznacza się bardzo szybkim działaniem. Daje się to uzyskać dzięki ominięciu znanego z literatury trudnego do zautomatyzowania sposobu szukania "ścieżki zastępozej" oraz dzięki uniknięciu rozwiązywania układu równań przy obliczaniu nowych wartości  $V_k$ ; w ciągu jednej iteracji tabela cen jest przeszukiwana tylko jeden raz.

Drugą zaletą omówionego programu jest uniknięcie komplikacji związanych z koniecznością oddzielnego rozpatrywania tzw. rozwiązań zdegenerowanych, czyli takich, dla których liczba niezerowych wartości  $x_{ik}$  jest mniejsza niż  $(m+n-1)$ . Wynika to z odmiennego traktowania zmiennych, które wchodzą do rozwiązania i mają wartości zerowe oraz takich, które nie wchodzą do rozwiązania /zawsze mających wartości zerowe/. Rozwiązania zdegenerowane program traktuje tak samo jak niezdegenerowane, co również zostało zilustrowane na omawianym przykładzie. W programie dostawców i odbiorców traktuje się w jednakowy sposób. Dzięki temu liczba rozkazów została zmniejszona dwukrotnie.

#### Literatura

- [1] GASS S.: Programowanie liniowe. Warszawa 1963, tłumaczenie z angielskiego
- [2] ZUHOVICKIJ S.I., ADVIEJEVA L.I.: Linijnoje i vypukloje programmirovaniye. Moskwa 1964.

#### A SOLUTION OF THE TRANSPORT PROBLEM USING A DIGITAL COMPUTER

#### Summary

The paper deals with the classical transport problem. The described method of solution is well adjusted to the computers for which the large matrix of costs must be recorded in an auxiliary memory, and the basic solution can be held in the high speed memory. The program is short on account of the assumed way of numerating the indices which permits to use only 4 arrays of variables mentioned in 2.2. The simplicity of the program is reached by a special reordering of the variables after each iteration. Therefore the computing process goes very quickly without any waste of time for seeking the "auxiliary path". The usual complications with the degenerated solutions don't exist in the described method.

## **STATISTICAL METHODS**



Instytut Maszyn Matematycznych  
Algorytmy № 6  
© 1966.07

519.27:681.14-523.8

GENERATOR LICZB LOSOWYCH O ROZKŁADZIE  
RÓWNOMIERNYM DLA MASZYNY ZAM-2

Ryszard ZIELIŃSKI

Pracę złożono 8.X.1965 r.

Generator liczb losowych o rozkładzie równomiernym jest podstawowy dla uzyskiwania liczb losowych o dowolnych rozkładach, dla realizacji procesów losowych oraz dla rozwiązywania wszelkiego rodzaju zadań z zakresu modelowania w warunkach przypadkowości. W niniejszym opracowaniu podano wyniki studiów nad niektórymi generatorami liczb losowych o rozkładzie równomiernym, pełne wyniki testowania generatora RNB który wybrano jako najlepszy generator dla maszyny ZAM-2/ oraz tablice tego generatora umożliwiające stosowanie go w przypadkach bardziej złożonych sposobów losowania, na przykład losowania warstwowego.

Najczęściej używanymi na maszynie ZAM-2 generatorami są: generator RNA [6] realizowany za pomocą sekwencji rozkazów w języku SAS:

UM.X

MN.C

PM.X

ze stałą początkową X 234.642.457.001 i stałą C 115.354.631.461 oraz generator kwadratowy KGA 2 realizowany za pomocą sekwencji rozkazów:

UM.X

MN.X

PW17

PM.X

ze stałą początkową  $X \equiv 234.642.457.001$ . Generatorem KGA nie będziemy się dalej zajmowali ze względu na to, że czas generowania jednej liczby jest około 30% dłuższy od czasu generowanie jednej liczby w generatorze RNA oraz ze względu na to, że ilość różnych liczb generatora KGA jest stosunkowo niewielka i przy podanej wyżej stałej wynosi ok. 100 000. Zalety tego generatora /brak określowości na poszczególnych pozycjach generowanych liczb/ czynią go szczególnie przydatnym tylko w specjalnych zastosowaniach.

Generator RNA należy do klasy generatorów mnożnikowych postaci:

$$x_n = (x_{n-1} \cdot c) \bmod M$$

lub ogólniej – do klasy generatorów mieszanych:

$$r_n = (\lambda \cdot r_{n-1} + A) \bmod M.$$

Wszędzie dalej będziemy zakładali, że  $M = 2^{35}$ .

Teoria tych generatorów jest obszernie opracowana, natomiast otwartą kwestią pozostaje wybór stałych  $C, \lambda, A$  dla generatora realizowanego na określonej maszynie cyfrowej z punktu widzenia czasu generowania, ilości potrzebnych miejsc pamięci oraz długości słowa. Stałe te powinny być przy tym wybrane tak, aby określony zestaw testów nie prowadził do odrzucenia hipotezy o tym, że ciąg liczb z danego generatora tworzy próbę prostą z populacji o założonym rozkładzie.

Generatory mieszane są generatorami okresowymi. Długość okresu zależy od stałych  $C$  i  $\lambda$ , przy czym okresowość poszczególnych cyfr binarnych w liczbach otrzymywanych z tego generatora jest różna. Zagadnienie okresowości generatorów mieszanych i mnożnikowych było rozpatrywane w pracy [8], na podstawie której opracowano tabelkę 1:

Tabela 1.

Okresowość generatora mnożnikowego i mieszanego na maszynie ZAM-2

Stała $\beta > 2$ , K nieparzyste	Okres gener.	Okres k-tego bitu				
		$34 \geq k \geq \beta + 1$	$k = \beta$	$\beta - 1 \geq k \geq 2$	$k = 1$	$k = 0$
$C = K \cdot 2^\beta + 1$	$2^{35-\beta}$	$2^{k+1-\beta}$		(*)		1
$C = K \cdot 2^\beta - 1$	$2^{35-\beta}$	$2^{k+1-\beta}$	(*)	(**)	2	1
$\lambda = K \cdot 2^\beta + 1$	$2^{35}$			$2^{k+1}$		
$\lambda = K \cdot 2^\beta - 1$	$2^{36-\beta}$	$2^{k+2-\beta}$		(**)		2

\*) bity mają taką samą wartość jak odpowiednie bity wartości początkowej generatora.

\*\*) bity mają stałą wartość lub okres równy 2, co zależy od ostatnich bitów wartości  $X_0$  lub wartości  $R_0$  i A.

Ponieważ stała C w generatorze RNA jest równa:

$$C = 2^4 \cdot 3^3 \cdot 37 \cdot 107 \cdot 6079 + 1$$

okres tego generatora wynosi  $2^{31}$ .

Najdłuższy możliwy okres generatora mnożnikowego wynosi  $2^{33}$  /przy  $\beta = 2$ . Nazwiemy generatorem RNB generator mnożnikowy ze stałą:

$$C = 2^2 \cdot 23^7 + 1.$$

Stała ta w zapisie oktalnym równa jest:

$$C \equiv 145.361.406.635 .$$

Najdłuższy możliwy okres generatora mieszanego otrzymujemy przy stałej:

$$\lambda = K \cdot 2^\beta + 1,$$

Gdzie K jest dowolną liczbą nieparzystą oraz  $\beta \geq 2$ . Własności statystyczne tego generatora zależą od  $\lambda$  i od stałej addytywnej A. W pracy [4] cytowany jest wynik Greenbergera z 1962 roku,

że współczynnik autokorelacji w generatorze mieszanym znajduje się w granicach:

$$\frac{1}{\lambda} - \frac{6A}{\lambda M} \left(1 - \frac{A}{M}\right) \pm \frac{\lambda}{M}$$

natomiast w pracy [5] w dyskusji tego wzoru dla  $K=1$  podaje się, że współczynnik autokorelacji jest równy zeru, gdy

$$A \cdot 2^{-35} = \frac{1}{2} \pm \frac{1}{6} \sqrt{3}$$

Taki generator, w którym:

$$\beta = 2$$

$$K = 1$$

$$A \cdot 2^{-35} = \frac{1}{2} - \frac{1}{6} \sqrt{3} \equiv 066.062.613.515$$

nazwiemy generatorem RNC, natomiast ten sam generator ze stałą

$$A \cdot 2^{-35} = \frac{1}{2} + \frac{1}{6} \sqrt{3} \equiv 226.546.647.430$$

generatorem RNC1. Generator ten realizuje się na maszynie ZAM-2 za pomocą następującej sekwencji rozkazów w języku SAS:

UA1009

UA.R

LW2

DO.R

DO.A

PA.R

SN+1

W dalszym ciągu pracy omówione zostaną wyniki testowania generatorów RNB, RNC; RNC1. Ponieważ liczby losowe z generatora RNB wypełniają przedział  $/0,1/$  czterokrotnie gęściej niż liczby z generatora RNA – ten ostatni generator nie był badany. Z innych rozważanych generatorów wymienimy generator Fibonacciego. Ze względu na wyniki zawarte w pracy [9] należałoby ograniczyć się do ciągu liczb pseudolosowych składającego się tylko z parzystych /lub tylko nie-

parzystych/ wyrazów ciągu Fibonacciego, ale - jak wykazały doświadczenia\*) - wtedy czas generowania jednej liczby jest dłuższy od czasu generowania jednej liczby w generatorze RNB. Z tego powodu generatora Fibonacciego bardziej szczegółowo nie badano.

#### Generator RNB

Generator RNB sprawdzono dwunastoma testami. Dla każdego tysiąca kolejnych liczb z tego generatora obliczano wartości następujących statystyk:

- statystyka A /por. statystyka M w [1]/: obliczano w każdym tysiącu wartość  $X_A$  średniej. Za wartość statystyki A przyjmowano wartość dystrybuanty rozkładu zmiennej losowej  $X_A$ . Według rozwinięcia Edgewortha otrzymujemy:

$$P\{X_A < x\} = \Phi[109.54452(x-0.5)] - 0.00005 \psi^{(3)}[109.54452(x-0.5)],$$

gdzie  $\Phi(t)$  oznacza wartość dystrybuanty, a  $\psi^{(3)}(t)$  wartość k-tej pochodnej dystrybuanty w rozkładzie normalnym  $N(0.1)$  w punkcie t. Ponieważ ostatni wyraz nie przekracza bezwzględnie wartości  $3 \cdot 10^{-5}$  przyjmowano, że rozkład  $X_A$  jest normalny ze średnią 0.5 i odchyleniem średnim  $1/109.54452$ ;

- statystyka B /por. statystyka V w [1]/: obliczano w każdym tysiącu wartość  $X_B$  średniego kwadratu. Za wartość statystyki B przyjmowano wartość dystrybuanty rozkładu zmiennej losowej  $X_B$  według rozwinięcia Edgewortha:

$$\begin{aligned} P\{X_B < x\} = & \Phi[106.066(x-1/3)] - 0.003367 \psi^{(2)}[106.066(x-1/3)] + \\ & - 0.0000357 \psi^{(3)}[106.066(x-1/3)] + \\ & + 0.000006 \psi^{(5)}[106.066(x-1/3)]. \end{aligned}$$

Ponieważ każdy z dwóch ostatnich wyrazów nie przekracza bezwzględnie wartości  $3 \cdot 10^{-5}$ , w obliczeniach korzystano tylko z dwóch pierwszych wyrazów tego rozwinięcia;

\*) Por. Dodatek: Porównanie czasu pracy generatorów.

- statystyka C /por. statystyka F w [1]/: przedział  $(0,1)$  podzielono na 64 równe przedziały i obliczano ilość  $n_1$  liczb w każdym z tych przedziałów. Następnie obliczano wartość statystyki  $\chi^2$  /obliczoną wartość statystyki  $\chi^2$  oznaczamy wszędzie przez  $\chi^2_e$ :

$$\chi^2_e = \sum_{i=1}^{64} \frac{\left(n_i - \frac{1000}{64}\right)^2}{\left(\frac{1000}{64}\right)}$$

Ponieważ ilość stopni swobody jest duża / $f=63$ / przyjęto, że zmieniona losowa U:

$$U = \sqrt{\frac{9f}{2}} \left[ \left( \frac{\chi^2_e}{f} \right)^{\frac{1}{2}} - 1 + \frac{2}{9f} \right] = 16.837458 \left( \sqrt{\frac{\chi^2_e}{63}} - 0.99647266 \right)$$

ma rozkład  $N(0,1)$  i za wartość statystyki C przyjmowano wartość dystrybuanty tego rozkładu w punkcie równym obliczonej wartości statystyki U;

- statystyka D /por. statystyka C w [1]/: przedział  $(0,1)$  podzielono na 8 równych przedziałów i obliczano liczbę  $n_{ij}$  ( $i=0, \dots, 7; j=0, \dots, 7$ ) takich par  $(x_k, x_{k-1})$  ( $k=0,1,\dots,999$ ), że  $x_k$  jest wartością z i-tego, natomiast  $x_{k-1}$  wartością z j-tego przedziału. Następnie obliczana wartość statystyki  $\chi^2$ :

$$\chi^2_e = \sum_{i=0}^7 \sum_{j=0}^7 \frac{\left(n_{ij} - \frac{1000}{64}\right)^2}{\left(\frac{1000}{64}\right)}$$

Podobnie jak przy obliczaniu statystyki C wprowadza się zmienną U o rozkładzie  $N(0,1)$  i wartość dystrybuanty tego rozkładu w punkcie równym zaobserwowanej wartości statystyki U przyjmuje się za wartość statystyki D;

- statystyka E /por. statystyka R w [1]/: bada się serie wartości leżących powyżej i poniżej wartości  $1/2$ . Przyjmuje się, że rozkład liczby  $X_E$  wszystkich serii jest normalny z parametrami:

$$E(X_E) = 500; \quad D^2(X_E) = 250$$

Za wartość statystyki E przyjmuje się wartość dystrybuanty tego rozkładu w punkcie równym obliczonej wartości statystyki  $X_E$ ;

- statystyka F: bada się serie wartości leżących powyżej i poniżej wartości  $1/2$ . Rejestruje się ilości serii o liczbie wyrazów  $1, 2, 3, 4, 5, 6$  oraz o liczbie wyrazów równej lub większej  $7$ . Zaobserwowany rozkład serii porównuje się testem  $\chi^2$  z rozkładem teoretycznym (za teoretyczną ilość serii o długości  $k$  przyjmuje się  $X_E/2^k$ ). Oblicza się wartość  $X_F$  dystrybuanty rozkładu  $\chi^2$  o 6 stopniach swobody w punkcie równym zaobserwowanej wartości statystyki  $\chi^2$ . Wartość  $1-X_F$  przyjmuje się za wartość statystyki F;

- statystyka G: bada się serie monotoniczne /serie wyrazów malejących i serie wyrazów rosnących/. Rejestruje się wartość statystyki  $X_G$  - łącznej ilości serii monotonicznych. Przyjmuje się, zgodnie z [3], że rozkład statystyki  $X_G$  jest normalny z parametrami:

$$E(X_G) = \frac{1}{3}(2n - 1) = 1999/3$$

$$D^2(X_G) = \frac{1}{90}(16n - 29) = 39.96373 .$$

Za wartość statystyki G przyjmuje się wartość dystrybuanty tego rozkładu w punkcie równym obliczonej wartości statystyki  $X_G$ ;

- statystyka H: bada się serie monotoniczne. Testem  $\chi^2$  porównuje się rozkład zaobserwowany i rozkład teoretyczny liczby serii o długości  $1, 2, 3, 4, 5, 6$  i  $7$  /rozkład teoretyczny podany jest w [3]/. Wartość dystrybuanty rozkładu  $\chi^2$  o 6 stopniach swobody w punkcie równym zaobserwowanej wartości  $\chi^2$  przyjmuje się za wartość statystyki H;

- statystyka I: generuje się liczby losowe o rozkładzie normalnym jako standaryzowane sumy dwunastu kolejnych liczb generatora RNB. Testem  $\chi^2$  porównuje się zgodność uzyskanego rozkładu z roz-

kładem normalnym. Wartość dystrybuanty rozkładu  $\chi^2$  w punkcie równym zaobserwowanej wartości statystyki  $\chi^2$  przyjmuje się za wartość statystyki I;

- statystyka K: generuje się liczby losowe o rozkładzie wykładniczym /algorytm z pracy [7]/ i testem  $\chi^2$  porównuje zgodność rozkładu zaobserwowanego z rozkładem teoretycznym. Za wartość statystyki K przyjmuje się wartość dystrybuanty rozkładu  $\chi^2$  w punkcie równym zaobserwowanej wartości statystyki  $\chi^2$ ;

- statystyka L: generuje się liczby losowe o rozkładzie trójkątnym jako sumy dwóch kolejnych liczb z generatora RNB i testem  $\chi^2$  porównuje się zgodność rozkładu zaobserwowanego z rozkładem teoretycznym. Za wartość statystyki L przyjmuje się wartość dystrybuanty rozkładu  $\chi^2$  w punkcie równym zaobserwowanej wartości statystyki  $\chi^2$ ;

- statystyka M: bada się rozkład mniejszej z dwóch liczb z generatora RNB. Testem  $\chi^2$  porównuje się zgodność zaobserwowanych realizacji z rozkładem teoretycznym. Za wartość statystyki M przyjmuje się wartość dystrybuanty rozkładu  $\chi^2$  w punkcie równym zaobserwowanej wartości statystyki  $\chi^2$ .

Wartości wymienionych statystyk /pomnożone przez 1000/ wraz z wartościami początkowymi w kolejnych tysiącach liczb z generatora RNB podano w tab. 2. Obliczenia wykonano dla pierwszych 200,000 liczb tego generatora.

Jeżeli ciąg liczb z generatora RNB stanowi próbę prostą z populacją o rozkładzie równomiernym w przedziale  $(0,1)$ , to każda z dwoistych statystyk powinna mieć rozkład równomierny w przedziale  $(0,1)$ .

Empiryczne rozkłady tych statystyk podano w tab. 3.

Sprawdzono testem  $\chi^2$  zgodność każdego z empirycznych rozkładów z tab. 3 z rozkładem równomiernym. Wartości  $P(\chi^2 < \chi_e^2)$ , gdzie  $\chi_e^2$  jest zaobserwowaną wartością statystyki  $\chi^2$ , zestawiono w tabeli 4. Dla porównania w tej samej tabelce podano wartości  $P(\chi^2 < \chi_e^2)$  obliczone dla pierwszych stu wartości statystyk A, B, ..., M.

Tablica 2

L.p.	Wartość początkowa	S t a t y s t y k a									
		A	B	C	D	E	F	G	H	I	K
1	234.642.457.001	115	104	592	651	285	440	809	476	773	427
2	357.421.133.041	007	004	520	984	776	876	742	827	933	133
3	032.127.211.101	162	201	078	117	624	868	401	567	809	581
4	343.374.771.141	557	540	183	278	795	637	483	869	714	747
5	064.216.353.201	987	992	605	465	306	782	110	315	622	139
6	207.433.637.241	800	847	738	525	829	480	880	052	956	308
7	142.671.125.301	942	926	869	392	500	278	977	263	502	390
8	326.376.515.341	732	732	952	130	795	043	895	192	394	179
9	005.610.307.401	137	140	900	897	624	958	344	594	251	255
10	267.166.403.441	347	310	960	975	306	443	908	211	499	137
11	354.755.101.501	255	192	342	173	021	318	024	036	342	616
12	204.624.301.541	841	907	579	312	205	646	177	462	316	287
13	340.440.303.601	476	507	076	148	885	163	788	437	891	773
14	226.251.207.641	973	979	325	219	776	195	692	915	440	147
15	050.573.115.701	428	299	396	042	376	445	401	382	594	434
16	307.106.125.441	908	911	795	984	979	321	962	174	237	431
17	025.706.340.001	804	729	609	930	999	117	865	052	695	768
18	376.474.054.041	428	386	320	005	141	939	344	357	256	345
19	210.353.172.101	816	853	122	295	475	926	637	157	272	782
20	270.234.012.141	985	991	609	941	885	094	766	269	068	890
21	171.032.434.201	656	671	320	387	757	048	344	760	692	802
22	213.666.760.241	645	535	565	622	329	721	665	545	719	879
23	200.305.305.301	775	611	768	925	671	422	073	155	743	307
24	265.035.736.341	297	331	052	465	013	022	065	336	692	846
25	137.234.570.401	254	437	940	867	950	389	829	108	437	596

Tablice generatora RNB

L.p.	Wartość pozaź-	S t a t y s t y k a										M
		A	B	C	D	E	F	G	H	I	K	
26	233.641.724.441	157	113	626	992	873	040	919	120	223	332	903
27	206.621.462.501	191	190	600	383	376	811	141	710	053	813	505
28	134.123.722.541	042	049	176	969	016	069	919	452	158	290	472
29	237.724.764.601	712	717	774	520	918	894	766	072	475	576	197
30	116.204.730.641	811	831	148	450	860	430	619	322	719	012	809
31	023.727.676.707	321	267	238	419	141	273	692	265	695	576	998
32	216.305.746.741	170	081	880	530	624	211	291	791	163	585	736
33	224.113.221.001	008	007	720	947	897	497	125	191	671	036	183
34	204.747.775.041	750	580	909	878	329	640	579	434	895	875	347
35	242.040.153.101	732	743	783	900	352	147	919	132	044	256	883
36	156.374.033.141	187	190	919	378	206	393	742	047	569	490	501
37	027.207.515.201	184	175	579	038	500	992	460	110	464	230	693
38	343.523.101.241	044	019	626	383	329	049	004	020	155	873	133
39	255.362.467.301	770	782	219	187	550	790	999	016	870	361	166
40	141.776.157.341	042	111	511	070	500	759	266	192	210	546	023
41	206.422.051.401	557	385	972	964	600	433	766	257	092	170	523
42	320.116.245.441	833	844	617	078	776	242	755	877	322	390	619
43	264.327.043.501	873	870	525	905	776	906	241	133	618	823	570
44	015.324.343.541	110	211	066	605	897	363	637	027	440	037	335
45	301.162.445.601	617	743	818	356	624	341	692	513	986	588	528
46	162.141.451.641	104	114	716	360	873	447	718	886	755	531	810
47	067.125.457.701	057	084	912	682	073	552	197	668	425	300	566
48	133.605.567.741	243	315	921	014	044	030	520	241	905	487	571
49	050.461.102.001	873	864	258	238	264	666	766	843	368	325	858
50	263.424.716.041	637	658	731	456	715	945	084	049	334	666	110

ciąg dalszy Tablicy 2

ciąg dalszy Tablicy 2

Tablice generatora RNB

I.p.	Wartość początkowa	A	B	C	D	E	F	G	H	I	K	L	M
51	267.766.134.101	207	146	424	516	736	843	766	758	668	841	692	186
52	167.035.054.141	079	090	461	139	671	892	895	595	021	866	258	409
53	237.725.576.201	778	818	290	774	550	502	006	012	651	037	015	451
54	077.760.222.241	504	438	204	000	844	798	742	672	922	594	103	441
55	273.100.650.301	115	220	928	998	525	820	460	674	239	844	412	101
56	314.037.400.341	751	712	419	130	103	835	718	099	246	635	649	493
57	214.350.332.401	100	190	378	406	648	940	460	983	225	787	785	421
58	205.173.566.441	575	719	933	884	285	170	490	827	136	072	673	230
59	306.675.424.501	643	721	520	270	285	443	766	652	174	101	818	589
60	011.425.764.541	008	004	262	930	757	777	940	388	121	143	289	263
61	325.361.126.601	480	622	895	647	352	131	219	661	199	128	379	070
62	203.077.172.641	506	588	145	136	694	730	579	634	707	952	531	275
63	143.364.240.701	950	932	849	374	352	512	291	308	242	676	069	363
64	040.210.410.741	874	920	900	387	450	191	766	914	715	618	175	349
65	344.167.763.001	094	060	136	068	694	139	809	374	047	737	301	794
66	273.302.637.041	489	376	238	286	624	584	344	698	812	133	672	357
67	033.155.115.101	416	357	638	845	264	165	766	022	507	335	947	820
68	102.777.075.141	933	886	193	433	908	523	372	982	922	909	122	137
69	244.004.657.201	125	267	798	663	897	482	718	907	179	445	383	086
70	121.116.343.241	435	446	076	338	500	465	242	500	327	911	571	399
71	172.260.031.301	165	256	392	994	648	777	372	146	739	235	526	452
72	346.440.621.341	464	480	761	798	600	606	908	681	047	759	121	141
73	202.037.613.401	421	489	148	690	927	703	608	815	801	182	668	857
74	354.052.107.441	830	895	574	438	171	411	637	015	666	128	473	736
75	017.105.005.501	588	654	758	374	795	746	197	851	013	476	041	441

Tablice generatora RNB

ciąg dalszy Tablicy 2

L.p.	Wartość początkowa	S t a t y s t y k a												M
		A	B	C	D	E	F	G	H	I	K	L		
76	301.430.405.541	783	709	197	839	306	317	829	540	378	509	818	709	709
77	155.520.607.601	404	451	842	447	962	054	742	775	425	840	587	255	255
78	302.035.713.641	635	689	266	731	757	086	940	281	491	889	656	708	708
79	151.664.021.701	553	512	502	667	475	188	829	037	906	618	430	314	314
80	314.713.231.741	221	171	842	516	425	718	848	847	266	326	445	114	114
81	330.037.644.001	473	541	036	104	285	820	742	781	748	418	906	985	985
82	315.361.560.041	501	661	039	406	624	542	637	486	740	213	922	037	037
83	034.605.076.101	391	476	052	865	329	908	029	051	251	139	802	400	400
84	103.242.116.141	180	217	139	847	141	886	097	526	057	748	537	400	400
85	264.424.740.201	875	910	659	596	873	703	550	963	491	238	253	895	895
86	312.055.464.241	369	434	902	671	736	228	880	685	205	822	809	559	559
87	274.100.212.301	164	107	824	266	329	218	344	391	187	990	651	423	423
88	241.445.042.341	311	188	801	690	575	861	141	148	374	284	470	406	406
89	170.270.074.401	336	417	016	005	525	855	607	840	179	879	758	617	617
90	065.531.430.441	415	317	795	834	671	147	742	361	502	395	366	261	261
91	336.155.366.501	359	247	013	320	935	507	930	479	391	370	049	857	857
92	246.334.026.541	893	935	254	628	425	229	020	059	239	179	764	499	499
93	032.621.270.601	980	965	989	977	205	883	242	524	499	222	226	320	320
94	117.775.434.641	661	667	789	023	829	870	372	333	170	908	739	485	485
95	633.224.602.701	205	333	378	151	450	909	177	437	268	651	337	744	744
96	122.517.052.741	758	494	738	998	450	587	430	863	671	080	841	335	335
97	025.050.525.001	031	064	716	360	989	496	829	504	322	009	631	127	127
98	032.641.501.041	262	090	813	470	918	211	908	268	197	468	533	046	046
99	015.476.057.101	809	818	694	030	992	035	158	331	026	829	839	293	293
100	351.006.137.141	756	817	204	675	500	320	637	003	358	132	347	612	612

ciąg dalszy Tablicy 2

Tablica generatora RNB

L.p.	Wartość początkowa	S t a t y s t y k a											
		A	B	C	D	E	F	G	H	I	K	L	M
101	142.406.021.201	756	817	374	125	329	931	766	650	499	427	437	492
102	131.715.605.241	020	024	167	117	844	527	948	387	336	952	226	917
103	121.361.373.301	877	888	974	605	224	198	097	364	617	805	147	805
104	356.211.263.341	509	493	387	056	736	899	766	269	985	403	977	722
105	200.261.355.401	848	866	365	793	715	310	981	068	084	668	902	642
106	223.011.751.441	806	861	442	795	956	466	788	887	457	677	829	993
107	205.066.747.501	898	917	045	160	908	427	141	363	266	493	639	958
108	051.140.447.541	249	285	638	438	033	088	430	774	525	314	474	617
109	155.662.751.601	645	533	452	859	999	015	999	001	988	630	144	347
110	015.736.155.641	383	472	167	282	795	435	460	464	941	519	152	704
111	110.626.363.701	544	625	160	626	776	472	020	082	126	025	420	124
112	242.423.673.741	806	816	815	406	575	210	829	637	833	303	574	650
113	254.222.406.001	160	196	634	197	425	508	692	936	384	858	698	360
114	324.322.422.041	522	640	090	066	525	150	579	546	506	783	942	587
115	076.630.040.101	992	993	899	977	003	043	040	188	012	430	309	137
116	245.053.160.141	369	476	088	342	400	333	291	924	199	140	847	906
117	076.730.101.201	886	910	867	543	285	397	242	518	237	867	245	822
118	262.156.726.241	531	580	502	587	736	759	788	224	454	896	760	628
119	013.303.554.301	856	839	130	042	873	350	608	849	740	085	388	975
120	275.456.504.341	449	330	651	525	908	770	829	629	668	686	413	376
121	253.013.636.401	420	586	170	465	057	173	416	994	854	062	153	209
122	065.073.272.441	914	891	930	552	550	999	885	340	809	463	715	695
123	124.641.330.501	624	595	142	465	647	889	291	801	858	291	890	315
124	272.646.070.541	407	442	190	351	897	597	372	421	433	908	281	588
125	367.665.432.601	053	055	122	303	205	393	742	513	446	145	426	891

Tablice generatora RNB

L.p.	Wartość początkowa	S t a t y s t y k a											
		A	B	C	D	E	F	G	H	I	K	L	M
126	254.677.676.641	958	970	082	675	001	127	110	225	076	253	284	123
127	303.271.144.701	745	673	574	401	329	603	141	104	643	670	851	098
128	255.431.514.741	698	672	136	219	829	823	809	807	739	793	874	187
129	256.535.267.001	240	227	834	921	352	829	520	876	358	469	752	559
130	053.204.343.041	508	331	461	969	776	730	579	134	643	666	616	081
131	001.223.021.101	052	067	498	548	967	227	158	332	078	394	311	251
132	350.421.201.141	800	848	046	351	376	958	608	997	114	188	857	861
133	332.613.163.201	968	955	208	583	329	110	637	321	961	303	459	920
134	204.021.047.241	565	396	642	045	475	782	197	895	468	076	900	668
135	272.666.735.301	958	972	033	470	715	913	242	003	546	368	631	874
136	000.424.725.341	513	488	042	475	859	641	718	788	384	810	941	927
137	011.307.117.401	828	808	861	824	329	830	242	193	047	435	543	527
138	114.755.613.441	621	629	651	698	425	567	344	256	530	279	805	323
139	236.254.711.501	975	965	369	989	243	922	766	486	411	599	131	275
140	314.254.511.541	827	825	034	329	500	314	627	454	646	255	135	301
141	111.631.113.601	121	173	897	716	908	665	890	412	873	586	274	549
142	335.642.417.641	316	427	475	758	264	155	141	315	068	190	364	732
143	133.774.725.701	401	439	117	570	648	983	550	361	055	027	539	020
144	144.540.335.741	011	052	925	865	425	212	637	987	719	237	209	209
145	055.211.150.001	708	716	993	937	757	713	692	114	843	662	341	444
146	120.267.264.041	964	957	651	258	205	516	637	323	453	237	113	344
147	246.057.002.101	678	736	801	338	082	187	266	711	433	758	281	269
148	044.270.222.141	700	497	651	960	400	078	242	067	774	986	466	369
149	307.037.244.201	302	138	992	188	928	372	097	691	300	373	562	562
150	200.264.170.141	908	912	045	570	550	812	579	177	071	400	071	400

## Tablice generatora RNB

## ciąg dalszy Tablicy 2

L.p.	Wartość poczat- kowa	S t a t y s t y k a									
		A	B	C	D	E	F	G	H	I	K
151	060.713.116.301	655	770	675	493	715	198	919	492	479	546
152	250.074.146.341	497	580	698	774	776	946	823	299	265	786
153	254.343.400.401	748	819	839	070	736	667	177	683	282	737
154	013.441.134.441	535	495	254	904	975	227	692	622	870	363
155	262.531.272.501	932	875	154	312	065	131	430	987	327.	000
156	316.564.132.541	386	515	433	587	329	816	766	438	889	230
157	364.535.574.601	884	922	122	360	950	359	908	254	537	848
158	121.606.140.641	113	127	686	758	694	259	242	266	411	479
159	343.541.506.701	171	236	609	539	425	418	579	524	545	124
160	270.750.156.741	911	881	622	959	400	833	742	583	751	070
161	071.026.031.001	915	914	396	698	285	065	401	806	034	249
162	164.553.205.041	972	972	097	986	400	929	401	961	764	902
163	176.153.763.101	818	830	642	433	352	178	141	053	378	129
164	311.440.243.141	823	871	748	338	205	658	158	044	306	335
165	224.624.325.201	844	773	716	738	224	367	930	011	099	062
166	130.130.311.241	625	595	031	176	997	276	550	558	446	725
167	076.400.277.301	045	061	829	520	873	132	973	166	570	045
168	245.244.367.341	220	307	234	042	450	573	242	635	585	830
169	245.140.661.401	489	633	869	774	600	818	608	888	630	373
170	241.725.455.441	837	881	068	484	400	732	829	508	144	924
171	342.646.653.501	762	729	017	024	188	850	895	037	960	541
172	061.774.553.541	266	370	592	511	156	918	034	153	739	486
173	211.403.255.601	467	414	157	360	956	242	637	051	352	311
174	271.552.661.641	997	997	924	266	224	574	344	975	213	446
175	053.347.267.701	589	680	365	262	425	789	665	818	597	532

Tablice Generatora RNB

ciąg dalszy Tablicy 2

L.p.	Wartość początkowa	A	B	C	D	E	F	G	H	I	K	L	M
176	233.260.777.741	175	302	117	630	600	411	718	898	216	609	312	339
177	243.003.712.001	200	235	016	058	873	302	809	644	197	393	218	660
178	311.240.126.041	984	978	705	634	575	985	317	932	276	468	166	426
179	332.511.744.101	763	541	587	104	376	674	460	323	191	316	216	128
180	111.111.264.141	862	825	102	758	859	743	637	989	499	160	809	153
181	324.752.406.201	191	141	617	442	935	205	948	174	622	009	588	060
182	274.375.432.241	162	171	954	415	693	840	579	747	570	842	476	444
183	264.526.460.301	244	299	592	804	205	195	266	755	858	159	257	644
184	351.115.610.341	691	627	960	663	525	729	550	998	938	429	809	372
185	004.477.142.401	958	898	941	937	648	217	880	533	176	530	381	096
186	101.012.776.441	283	270	786	429	400	399	766	732	716	027	363	453
187	177.625.234.501	399	354	905	102	306	600	242	662	743	493	107	326
188	152.106.174.541	932	946	741	744	897	431	987	123	830	041	447	077
189	231.211.736.601	795	736	488	258	306	918	219	548	950	825	814	256
190	306.520.402.641	704	710	215	154	757	676	401	618	715	955	767	960
191	004.216.050.701	420	336	915	882	987	120	919	287	602	959	339	018
192	374.672.420.741	834	872	959	907	694	570	718	909	922	639	695	998
193	074.122.573.001	107	124	190	755	575	754	344	234	936	486	798	190
194	177.126.047.041	409	281	678	387	575	774	579	972	627	961	769	890
195	014.310.725.101	196	258	492	888	425	671	692	750	960	637	848	147
196	024.063.305.141	103	097	984	208	082	753	809	100	014	633	366	814
197	030.441.467.201	186	091	709	705	141	884	097	379	374	821	579	611
198	136.243.553.241	693	703	142	659	205	938	034	064	597	626	154	676
199	144.315.041.301	431	466	270	383	205	010	490	033	242	123	165	983
200	144.470.031.341	778	789	675	525	003	087	344	526	622	590	067	437

Tablica 3.

Empiryczne rozkłady badanych statystyk

Przedział	Statystyka											
	A	B	C	D	E	F	G	H	I	K	L	M
0.0-0.1	15	18	24	19	14	18	15	30	23	18	13	17
0.1-0.2	27	20	27	19	9	24	19	23	19	21	28	25
0.2-0.3	15	18	15	14	22	16	21	16	22	21	17	25
0.3-0.4	13	17	14	26	25	18	15	22	20	24	24	28
0.4-0.5	22	21	11	23	21	20	17	15	25	22	21	23
0.5-0.6	17	15	15	19	19	17	13	21	16	19	23	15
0.6-0.7	16	16	26	21	23	13	26	19	22	19	16	24
0.7-0.8	21	19	21	15	25	25	31	15	18	17	17	11
0.8-0.9	31	30	20	15	20	27	22	20	19	25	22	16
0.9-1.0	23	26	27	29	22	22	21	19	16	14	19	16

Tablica 4.

Zgodność rozkładu obserwowanych statystyk z rozkładem teoretycznym

Statystyka	Wartości $P(\chi^2 < \chi^2_e)$ dla pierwszych 100 wartości statystyk	Wartości $P(\chi^2 < \chi^2_e)$ dla pierwszych 200 wartości statystyk
A	0.581	0.919
B	0.054	0.633
C	0.599	0.931
D	0.897	0.710
E	0.665	0.744
F	0.616	0.544
G	0.787	0.863
H	0.425	0.572
I	0.544	0.466
K	0.486	0.157
L	0.798	0.553
M	0.945	0.888

Uzyskane wyniki nie pozwalają na odrzucenie hipotezy, że pierwsze 200,000 liczb generatora RNB stanowią próbę prostą z populacji o rozkładzie równomiernym.

## Generatory RNC i RNC1

Empiryczny rozkład statystyk G i H w próbce stuelementowej /dla pierwszych 100,000 liczb/ z generatorów RNC i RNC1 podano w tab. 5.

Empiryczny rozkład statystyk Gi H w generatorach RNC i RNC1

Przedział	Generator RNC		Generator RNC1	
	G	H	G	H
0.0-0.1	97	97	74	93
0.1-0.2	2	2	15	3
0.2-0.3	1	-	5	1
0.3-0.4	-	1	1	3
0.4-0.5	-	-	2	-
0.5-0.6	-	-	2	-
0.6-0.7	-	-	1	-
0.7-0.8	-	-	-	-
0.8-0.9	-	-	-	-
0.9-1.0	-	-	-	-
$\chi_e^2$	841.4	841.4	473.6	766.8
$P(\chi^2 > \chi_e^2)$	$<10^{-7}$	$<10^{-7}$	$<10^{-7}$	$<10^{-7}$

W tab. 5 podano obliczoną wartość statystyki  $\chi^2$  w teście zgodności rozpatrywanych rozkładów empirycznych z rozkładem równomiernym oraz wartość  $P(\chi^2 > \chi_e^2)$  dla dziewięciu stopni swobody.

Uzyskane wyniki prowadzą do odrzucenia hipotezy, że pierwsze stu tysięcy liczb z generatorów RNC i RNC1 tworzy próbę prostą z populacji o rozkładzie równomiernym.

Dalszych studiów nad generatorami mieszanymi dla maszyny ZAM-2 nie prowadzono, gdyż wybór takiej stałej A, dla której uzyskanoby zadowalające wyniki wymaga dużej ilości pracy eksperimentalnej. Badania dla innej wartości  $\lambda$  są nieopłacalne, gdyż generator z innym niż podano  $\lambda$  byłby zbyt wolny.

Wobec pozytywnych wyników testowania generatora RNB prace nad generatorami typu RNC wydają się niecelowe. Należy zaznaczyć, że chociaż w generatorze RNC występują operacje dodawania i przesunięcia podczas gdy w generatorze RNB operacja mnożenia, to czas generowania liczb losowych jest w obu generatorach tego samego rzędu /stwierdzono to eksperymentalnie/. Wynika to stąd, że chociaż dodawanie trwa krócej niż mnożenie, to w generatorze RNB tylko trzy razy pobiera się liczby z pamięci, podczas gdy w generatorze RNC - pięć razy, a czas dostępu do pamięci w maszynie ZAM-2 jest długi w porównaniu z czasem wykonywania operacji arytmetycznych. Ponadto generator RNC realizuje się za pomocą ciągu 7 rozkazów /ostatni rozkaz - SN+1 - "gasi" ewentualny nadmiar powstały w wyniku realizacji rozkazów D0 i LW/, podczas gdy generator RNB realizuje się za pomocą tylko trzech rozkazów.

Przedstawione wyżej wyniki wskazują, że jako standartowy generator liczb losowych o rozkładzie równomiernym dla maszyny ZAM-2 powinien być przyjęty generator RNB.

#### Literatura

- [1] CLARK Ch. E., HOLZ B.W.: Exponentially Distributed Random Numbers, Published for Operations Research Office, 1960.
- [2] GAJEWSKI H., ZIELIŃSKI R.: Uwagi o pewnym generatorze kwadratowym, Prace IMM, Algorytmy 1965:2,4.
- [3] HALD A.: Statistical Theory with Engineering Applications, 1957.
- [4] HAMMERSLEY J.M., HANDSCOMB D.C.: Monte Carlo Methods, 1964.
- [5] MARSAGLIA G.: Random Variables and Computers, Transactions of the Third Prague Conference 1962 "Information Theory, Statistical Decision Functions, Random Processes".
- [6] PLESZCZYŃSKA E.: Technika stosowania metod Monte-Carlo na ZAM-2, Prace IMM, Algorytmy 1965, Vol. III, No 5.
- [7] PLESZCZYŃSKA E.: Niektóre metody generowania realizacji procesu Poissona, Prace IMM, Algorytmy 1963, Vol. I, N° 2.
- [8] STOCKMAL F.: Calculation with Pseudo-Random Numbers, Journal of the Association for Computing Machinery, Vol. 11, No 1, January, 1964.
- [9] TAUSKY O., TODD J.: Generation and Testing of Pseudo-Random Numbers Symposium on Monte Carlo Methods, ed.H.A. Meyer 1956.

## RANDOM NUMBER GENERATOR WITH UNIFORM DISTRIBUTION FOR ZAM-2 COMPUTERS

Summary

The random number generator with uniform distribution is basic for obtaining random numbers with an arbitrary distribution, to realize random processes and to solve any kind of task of simulation under random conditions. The paper presents the results carried out on some random number generators with uniform distribution, full results of testing the generator RNB /chosen as the best one for the ZAM-2 computer/ and this generator tables enabling its application to more complex cases of sampling e.g. for the stratified one.

D O D A T E KPorównanie czasu pracy generatorów

Czas pracy generatorów RNB i F /Fibonacciego/ badano za pomocą następujących programów:

Program			
P1	P2	P3	P4
BLOK(0):X,C X 234.642.457.001 C 145.361.406.635 JĘZYK SAS: 0 SK a a/.UB+2 .SK+2 SS 10000			
1 UM.X MN.C PM.X	1 UA.X DO.C PA.X DO.C PA.C .SN+1	1 UM.X MN.C PM.X DO.C PA.C MN.C PM.X	1 UA.X DO.C PA.X DO.C PA.C .SN+1 UA.X DO.C PA.X DO.C PA.C .SN+1
2 .SB+4 .BB+4			
.SK-5 .SK+0 SS0 SS1	.SK-8	.SK-8	.SK-14

/Rozkazy poza obszarami zamkniętymi liniami przerywanymi są identyczne we wszystkich czterech programach/.

Miejsca dla zmiennych X, C zostają ustalone pierwszym rozkazem, natomiast zmieniając a od pewnej liczby  $a_0$  do  $a_0 + 15$  /w obliczeniach przyjęto  $a_0 = 32$ / uzyskuje się wszystkie możliwe po-

żenja w pamięci początku wykonywanego ciągu rozkazów względem ustalonych miejsca dla X i C. Przedmiotem badania jest czas wykonywania sekwencji rozkazów oznaczonych numerem 1, natomiast sekwencje rozkazów oznaczone numerem a oraz 2 służą do zasygnalizowania, że określona ilość /w naszym przypadku 10000/ liczb losowych została już wygenerowana. Zmierzony czas będzie oczywiście czasem wykonania wszystkich rozkazów od numeru a do numeru 2. Do ustalenia czasu wykonywania tylko rozkazów generatora RNB przeznaczone są programy P1 i P3. Jeżeli czas wykonania programu  $P_1$  wynosi  $t_1$ , to za czas wykonania 10000 razy sekwencji /oznaczamy tą sekwencję przez S/:

UM.X  
MN.C  
PM.X

przyjmiemy  $t_3 - t_1$ . Analogicznie dla czasu generowania 10000 liczb w generatorze F otrzymujemy czas  $t_4 - t_2$ . Ponieważ dla różnych a otrzymujemy różnych czas pracy poszczególnych programów, za czas  $t_1$  przyjmiemy czas średni. Wszystkie zaobserwowane czasy oraz obliczone czasy średnie przedstawiono w następującej tabelce:

a	Czas wykonywania programu /w sek/			
	P1	P2	P3	P4
32	80	64	129	133
33	71	64	120	132
34	71	70	121	141
35	71	71	127	135
36	71	70	135	140
37	71	77	134	134
38	79	85	140	142
39	85	84	154	141
40	86	85	147	143
41	85	77	153	127
42	91	85	143	134
43	84	72	135	120
44	85	71	140	126
45	84	70	129	127
46	77	64	127	121
47	78	64	127	126
Razem	1269	1173	2161	2122
Srednia	79.3125	73.3125	135.0625	132.6250

Otrzymujemy czas generowania 10000 liczb:

dla generatora RNB - 55.75 sek

dla generatora F - 59.3125 sek.

Zwracamy uwagę, że:

- 1/ przy obliczeniu średnich  $t_1$  uwzględniono tylko położenie początku generatora względem miejsc roboczych, nie uwzględniono natomiast wzajemnego położenia miejsc X i C względem siebie;
- 2/ przyjmując, że czas  $t_3-t_1$  /lub odpowiednio  $t_4-t_2$ / jest czasem generowania 10000 liczb założono, że zależność między czasem wykonywania programu a liczbą sekwencji S jest liniowa. Jest to tylko w przybliżeniu słuszne, gdyż różnica między programem P1 i P3 polega nie tylko na różnej ilości sekwencji S, ale również na tym, że wszystkie rozkazy poczynając od rozkazu oznaczonego numerem 2, umieszczone są w obu programach w innych miejscach pamięci.

Z tych powodów otrzymany wynik należy traktować jako przybliżony.



STOCHASTYCZNE ALGORYTMY W ZAGADNIENIACH OPTYMIZACJI

Komentowany przegląd bibliograficzny

Ryszard ZIELIŃSKI

Pracę złożono 14.5.1966 r.

W pracy podano krótki przegląd podstawowych idei metod probabilistycznych w zagadnieniach optymizacji.

Każde zagadnienie optymizacji można przedstawić w następującej postaci: dany jest pewien zbiór  $X$  oraz funkcja  $F$  odwzorowująca ten zbiór w zbiór liczb rzeczywistych. Należy znaleźć taki punkt  $x^* \in X$ , żeby dla każdego  $x \in X$  było

$$F(x^*) \leq F(x).$$

/1/

W zagadnieniach optymizacji funkcja  $F$  nazywa się funkcją celu, funkcją-kryterium lub krótko kryterium. Każdy element zbioru  $X$  nazywa się decyzyją, element  $x^*$  spełniający dla każdego  $x \in X$  związek /1/ decyzyją optymalną. Czasami funkcja  $F$  ma sens również poza zbiorem  $X$ ; wtedy zbiór  $X$  przyjęto nazywać zbiorem decyzji dopuszczalnych lub zbiorem rozwiązań dopuszczalnych.

Zagadnienia optymizacji znane są również jako zagadnienia programowania, przy czym:

- a/ jeżeli zbiór  $X$  jest podzbiorem w przestrzeni euklidesowej wyznaczonym przez układ nierówności liniowych oraz  $F$  jest funkcją liniową, otrzymujemy znane zadanie programowania liniowego;
- b/ jeżeli zbiór  $X$  jest wyznaczony tak jak w punkcie a/ oraz kryterium  $F$  ma postać

$$F(x) = xBx + Ax$$

gdzie A, B - macierze kwadratowe odpowiedniego stopnia, otrzymujemy zadanie programowania kwadratowego;

c/ jeżeli zbiór X składa się tylko z takich punktów

$x = (x_1, x_2, \dots, x_n)$ , że każde  $x_i$  może być tylko liczbą całkowitą, otrzymujemy zadanie programowania w liczbach całkowitych;

d/ jeżeli elementami zbioru X są funkcje, wkraczamy w zagadnienia rachunku wariacyjnego i programowania dynamicznego.

Będziemy rozpatrywali tylko takie przypadki, gdy X jest zbiorem w przestrzeni euklidesowej. O zbiorze X będziemy zakładali, że zawiera punkt  $x^*$  spełniający dla każdego  $x \in X$  warunek /1/, to znaczy że istnieje rozwiązanie optymalne.

Rozwiązanie zadania optymizacji okazuje się zwykle trudne, a nieraz nawet niemożliwe. Nawet w najprostszym przypadkach, na przykład w przypadku programowania liniowego, zadowalamy się znalezieniem takiego  $x_\varepsilon$ , że /przy ustalonym  $\varepsilon$ /:

$$|x_\varepsilon - x^*| < \varepsilon, \quad /2/$$

to znaczy znalezieniem decyzji dostatecznie bliskiej decyzji optymalnej, lub takiego  $x_\delta$ , że /przy ustalonym  $\delta$ /:

$$|F(x_\delta) - F(x^*)| < \delta, \quad /3/$$

to znaczy takiej decyzji, której skutki dostatecznie mało różnią się od skutków decyzji optymalnej.

Praktycznie wystarcza często przekonanie, że algorytm z dostatecznie dużym prawdopodobieństwem prowadzi do wyniku  $x_\varepsilon$  lub  $x_\delta$  spełniającego /2/ lub odpowiednio /3/. Prowadzi to do zastąpienia podanego na wstępie sformułowania zadania optymizacji jednym z następujących sformułowań:

#### Zadanie A:

W ustalonym zbiorze X znaleźć taki punkt  $x_\varepsilon$ , aby

$$\Pr \{ |x_\epsilon - x^*| < \epsilon \} > 1 - \epsilon_1, \quad /4/$$

gdzie  $\epsilon, \epsilon_1$  - ustalone stałe dodatnie.

#### Zadanie B:

W ustalonym zbiorze  $X$  znaleźć taki punkt  $x_\delta$ , aby

$$\Pr \{ |F(x_\delta) - F^*| < \delta \} > 1 - \delta_1, \quad /5/$$

gdzie  $\delta, \delta_1$  - ustalone stałe dodatnie oraz  $F^* = F(x^*)$ .

Sformułowanie zadań A i B upoważnia nas do traktowania  $x_\epsilon$  oraz  $x_\delta$  jako zmiennych losowych.

Pokażemy dwa przykłady, w których wynik rozwiązania jest rzeczywiście zmienną losową.

#### Przykład 1

Pewien wskaźnik  $V$  powinien w procesie sterowania produkcją być doprowadzony do pewnego poziomu  $V_0$  takiego, żeby wskaźnik  $W$  mierzący efekt produkcyjny osiągał maksimum /sterowanie ekstremalna/. Zwykle efekt produkcyjny, a więc i wartość wskaźnika  $W$ , zależy nie tylko od wielkości regulujących  $V$ , lecz również od całego szeregu przyczyn, których sumaryczny wpływ ma charakter losowy. Ustawiając wskaźnik  $V$  na różnych poziomach  $V_1, V_2, \dots$  obserwujemy wyniki  $W_1, W_2, \dots$ . Za wartość optymalną  $V_0$  przyjmujemy na przykład tę wartość, dla której zaobserwowano najlepszy wynik. Ponieważ  $W_1, W_2, \dots$  są realizacjami pewnych zmiennych losowych, więc tak określone  $V_0$  jest również realizacją pewnej zmiennej losowej. Za zadowalającą uznamy taką procedurę ustalania wartości  $V_0$ , która prowadzi do spełnienia warunków typu /4/ lub /5/.

#### Przykład 2

Z obszaru  $X$  wybieramy w pewien losowy sposób ciąg punktów  $x_1, x_2, \dots$  i w każdym z nich obliczamy wartość funkcji-kryterium. Otrzymujemy ciąg  $F_1, F_2, \dots$ . Pewną funkcję określoną na tych ciągach przyjmujemy za estymator decyzji optymalnej. Taki sposób

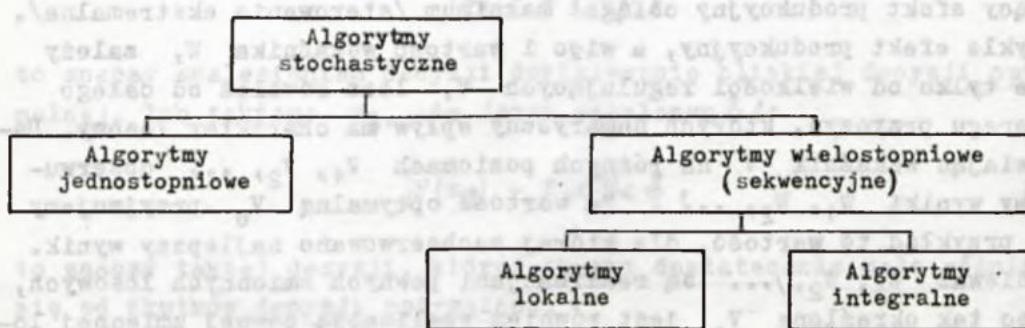
postępowania jest typowy dla metod Monte Carlo. Algorytmy obliczeniowe zbudowane na tej zasadzie będziemy nazywali algorytmami stochastycznymi.

W szczególnym przypadku, gdy zastosowany algorytm i technika obliczeniowa prowadzą w ścisłe zdeterminowany sposób do jednego wyniku  $x_\varepsilon$ , mamy do czynienia ze zmienną losową jednopunktową, przy czym jeśli dla ustalonego  $\varepsilon$  jest  $|x_\varepsilon - x^*| < \varepsilon$ , to warunek /4/ jest zawsze spełniony, natomiast w przypadku, gdy  $|x_\varepsilon - x^*| \geq \varepsilon$  warunek ten nigdy nie jest spełniony.

Metody optymizacji oparte na eksperymencie statystycznym /jak w Przykładzie 1/ lub wykorzystujące algorytmy stochastyczne /co jest równoważne pewnym eksperimentom statystycznym/, będące nazywane statystycznymi metodami optymizacji. W dalszym ciągu zajmiemy się głównie algorytmami stochastycznymi, poświęcając innym zagadnieniom optymizacji statystycznej końcową część opracowania.

#### Klasyfikacja algorytmów stochastycznych

Przyjmiemy następującą klasyfikację algorytmów stochastycznych:



Uzasadnienie poszczególnych nazw użytych w powyższej klasyfikacji wyniknie w sposób oczywisty z dalszej części opracowania. Podawane jako przykłady algorytmy będące oznaczali kolejno symbolami A<sub>1</sub>, A<sub>2</sub>, ...

### Algorytmy jednostopniowe

Algorytmy jednostopniowe znane są w literaturze jako metody Monte Carlo poszukiwania ekstremum funkcji.

/A1/ Najprostszym algorytmem jednostopniowym jest algorytm rozpatrywany w pracy [74] i wózniej w pracy [8]: w zbiorze decyzji  $X$  wybieramy losowo  $n$  punktów  $x_1, x_2, \dots, x_n$  i w każdym z nich obliczamy wartość kryterium  $F$ . Za najmniejszą wartość funkcji  $F$  na zbiorze  $X$  przyjmujemy:

$$\hat{F} = \min_{1 \leq i \leq n} F(x_i).$$

Za optymalną decyzje  $x^*$  przyjmujemy jedną z tych decyzji, które realizują  $\hat{F}$ . Losowanie punktów w zbiorze  $X$  przeprowadza się zwykle zgodnie z równomiernym rozkładem prawdopodobieństwa. Jako ocenę rozwiązania przy ustalonym  $n$  przyjmuje się prawdopodobieństwo  $S$ , że miara  $\mu$  podzbioru  $X$ , na którym funkcja przyjmuje wartości mniejsze od  $\hat{F}$ , równa jest co najwyżej ustalonej liczbie  $\alpha$  /zakłada się, że  $\mu(X) = 1$  oraz  $0 < \alpha \leq 1$ .

Algorytm A1, a w szczególności schemat wnioskowania, jest typowy dla zagadnień znanych pod nazwą "nieparametryczne przedziały ufności". Prace w tym kierunku zostały zapoczątkowane przez S.S. Wilks'a [71] w 1941 roku oraz A. Walda [69] w 1943 roku. Nowe informacje na ten temat oraz dalsze pozycje bibliograficzne znaleźć można w pracy [32].

/A2/ Pewna inna klasa algorytmów jednostopniowych związana jest z teorią wartości ekstremalnych czyli skrajnych statystyk pozycyjnych. Wiadomo /por. np [65]/, że przy dostatecznie dużych liczbach  $n$  próbek rozkład wartości największej z próbki ma jedną z następujących trzech postaci:

$$\Phi_\alpha(x) = \begin{cases} 0 & , \text{ dla } x \leq 0 \\ e^{-x^{-\alpha}} & , \text{ dla } x > 0 \end{cases}$$

$$\Psi(x) = \begin{cases} e^{-(x)^a}, & \text{dla } x \leq 0 \\ 1, & \text{dla } x > 0 \end{cases}$$

$$\Lambda(x) = e^{-e^{-x}}.$$

Ponieważ algorytmy stochastyczne stosowane są zwykle na maszynach matematycznych, nie ma zasadniczych trudności w uzyskaniu dostatecznie dużych  $n$ . Wykonując  $N$  prób  $n$ -elementowych i rejestrując w każdej z nich wartość największą otrzymujemy ciąg wartości  $\hat{F}_1, \hat{F}_2, \dots, \hat{F}_N$ . Oszacowanie na podstawie tego ciągu pewnego parametru odpowiedniego rozkładu / $\Phi, \Psi_a$  lub  $\Lambda$ / jest równoważne oszacowaniu maksymalnej wartości funkcji  $F$  na zbiorze  $X$ . Budowa takich estymatorów jest rozważana na przykład w [65].

Charakterystyczną cechą wymienionych wyżej algorytmów jest to, że otrzymany wynik nie zależy od kolejności, w jakiej wylosowane zostały poszczególne punkty ze zbioru  $X$ , a prawdopodobieństwo wylosowania punktu z dowolnego podzbioru zbioru  $X$  nie zależy od wyników poprzednich losowań. Właściwość ta wyróżnia algorytmy jednostopniowe.

#### Algorytmy wielostopniowe

Algorytmy wielostopniowe znane są w literaturze jako stochastyczne metody poszukiwania ekstremum /termin angielski: random search lub random seaking, termin rosyjski: случайный поиск/.

Ogólny schemat postępowania jest następujący: w sposób losowy, zgodnie z pewnym rozkładem prawdopodobieństwa  $P_0$ , wybiera się w zbiorze  $X$  punkt  $x_0$ . Jeżeli już wybrano  $n$  punktów, to bądź wartość pewnej funkcji zależącej od wylosowanych punktów  $x_1, x_2, \dots, x_{n-1}$  i wartości  $F(x_0), F(x_1), \dots, F(x_{n-1})$  przyjmuje się za decyzję optymalną, bądź losuje się następny punkt  $x_n$  zgodnie z rozkładem prawdopodobieństwa  $P_n$  zależnym od wyników dotychczasowych losowań. W ogólniejszym przypadku na każdym  $i$ -tym etapie można losować nie jeden, lecz  $n_i$  punktów.

Funkcja  $F$  może mieć na zbiorze  $X$  więcej niż jedno ekstremum lokalne. Zgodnie z podanym na wstępie sformułowaniem interesuje nas najmniejsze z nich. Niektóre algorytmy wielostopniowe są zbudowane w taki sposób, że prowadzą do uzyskania ekstremum lokalnego /podobnie jak metody gradientowe/ zamiast ekstremum absolutnego. Algorytmy takie będziemy nazywali algorytmami lokalnymi. Algorytmy prowadzące do uzyskania wartości najmniejszej w zbiorze  $X$  będą nazywali algorytmami integralnymi.

### Algorytmy lokalne

/A3/ Jednym z najprostszych algorytmów lokalnych jest następujący algorytm: wybieramy w  $X$  pewien punkt  $x_0$  i obliczamy w tym punkcie wartość kryterium  $F$ . Niech  $X$  będzie zbiorem w przestrzeni  $m$ -wymiarowej i oznaczmy przez  $\xi$   $m$ -wymiarowy wektor jednostkowy z początkiem w punkcie  $0$ . Koniec wektora  $\xi$  opisuje sferę. Niech będzie ustalony rozkład prawdopodobieństwa na tej sferze, na przykład rozkład równomierny. Przez  $\xi_i$  będziemy oznaczać  $i$ -tą reakcję wektora  $\xi$ . Jeżeli już wylosowano punkty  $x_1, x_2, \dots, x_{k-1}$ , to bądź kończymy poszukiwanie, bądź wybieramy następny punkt w następujący sposób:

$$x_k = \begin{cases} x_{k-1}, & \text{gdy } F(x_{k-1} + \delta \cdot \xi_k) \geq F(x_{k-1}) \\ x_{k-1} + \delta \cdot \xi_k, & \text{w przeciwnym przypadku} \end{cases}$$

gdzie  $\delta > 0$  jest pewną stałą, zwaną długością kroku. Proces kończymy, gdy po dostatecznie dużej liczbie kroków pozostajemy w ustalonym punkcie. Algorytm ten rozważany jest w pracach [50], [51], [59], natomiast w pracy [52] porównywany jest z innymi algorytmami. /A4/ W pracy [53] rozpatruje się następujący algorytm znajdowania punktu  $x_k$ :

$$x_k = x_{k-1} + \Delta x_k$$

$$\Delta x_k = \begin{cases} \Delta x_{k-1}, & \text{gdy } F(x_{k-1} + \Delta x_{k-1}) < F(x_{k-1}), \\ \delta \cdot \xi, & \text{gdy } F(x_{k-1} + \Delta x_{k-1}) \geq F(x_{k-1}) \quad \text{oraz} \\ & F(x_{k-1} + \delta \cdot \xi) < F(x_{k-1}), \\ 0, & \text{poza tym,} \end{cases}$$

gdzie  $\delta$  oraz  $\xi$  mają takie samo znaczenie jak w poprzednio rozpatrywanym algorytmie A3.

Podany wzór opisuje następujący sposób postępowania: jeżeli w  $k$ -tym kroku znaleźliśmy taki punkt  $x_k$ , że  $F(x_k) < F(x_{k-1})$ , to wykonujemy dalsze kroki o długości  $\delta$  w kierunku wyznaczonym przez wektor  $x_k - x_{k-1}$  tak dugo, aż w pewnym punkcie  $x_s$  będzie  $F(x_s) \geq F(x_{s-1})$ ; osiągnąwszy punkt  $x_{s-1}$  jako następnego poszukujemy losowego takiego punktu  $x_s$  na powierzchni sfery o środku w punkcie  $x_{s-1}$  i o promieniu  $\delta$ , w którym  $F(x_s) < F(x_{s-1})$ .

W cytowanej już pracy [53] podaje się dla algorytmu A4 zależność między odległością punktu początkowego  $x_0$  od ekstremum a średnią liczbą kroków potrzebnych do osiągnięcia ekstremum. Algorytm ten omawiany jest również w [51].

W literaturze znaleźć można szereg innych odmian algorytmów lokalnych. Obszerny przegląd różnych algorytmów oraz dalsze pozycje bibliograficzne znaleźć można w pracy [51], a pewien pomysł związany z ustalaniem wielkości  $\delta$  w pracy [59].

### Algorytmy integralne

Cechą wyróżniającą algorytmy integralne jest to, że prowadzą do uzyskania absolutnego minimum funkcji  $F$  w zbiorze  $X$ . Wynika stąd, że wszystkie algorytmy jednostopniowe są algorytmami integralnymi. Różnorodność sposobów konstrukcji wielostopniowych algorytmów integralnych ilustrują podane niżej przykłady.

/A5/ Jeden ze sposobów konstrukcji algorytmów integralnych polega na wprowadzeniu do algorytmu lokalnego pewnego dodatkowego wektora  $w_k$  zwanego wektorem pamięci. Dzięki temu algorytmy powinny być "mało czułe" na lokalne minima funkcji. Oznacza to, że proces poszukiwania najmniejszej wartości funkcji nie powinien "zatrzy-

mywać się" w minimum lokalnym, ale powinien do niego wrócić, jeżeli minimum to okaże się wartością najmniejszą. Schemat algorytmu jest następujący: w  $m$ -wymiarowym zbiorze  $X$  wybieramy pewien punkt początkowy  $x_0$  oraz ustalamy wektor  $w_0 = 0$ . Jeżeli już osiągnięto punkt  $x_k$  postępujemy według następującego procesu iteracyjnego:

$$\Delta x_{k+1} = \delta \cdot g(\xi_1, \dots, \xi_k, F, w_k)$$

$$x_{k+1} = x_k + \Delta x_{k+1}$$

$$w_{k+1} = h(w_k, x_1, \dots, x_{k+1}, F),$$

gdzie:  $\delta$  oraz  $\xi_i$  mają znaczenie takie jak w A3,

$g = [g_1, \dots, g_m]$  - wektor-funkcja taka, że  $|g| = 1$ ,

$w_k$  = wektor pamięci.

Przy odpowiednio dobranej funkcji  $g$  przyrost  $\Delta x_{k+1}$  będzie różny od zera wtedy, gdy w punkcie  $x_k$  funkcja  $F$  osiąga minimum lokalne oraz będzie zerem tylko wtedy, gdy  $F(x_k)$  jest najmniejszą wartością funkcji  $F$  w zbiorze  $X$ . Algorytmy z różnymi wektorami pamięci rozpatrywane są na przykład w pracach [57], [51], [60]. Dla konstrukcji wektora pamięci wykorzystuje się koncepcje stochastycznej teorii uczenia się, której obszerny wykład stanowi praca [10].

/A6/ Inna koncepcja wielostopniowego algorytmu integralnego przedstawiona jest w pracy [30]. Jeżeli już osiągnięto punkt  $x_k$ , to prawdopodobieństwo, że jako następny zostanie wylosowany punkt z podzbioru  $S \subset X$  określa się w następujący sposób:

$$P(S/x_k) = \begin{cases} r(F(x_k)) + \frac{v(S)}{v(X)} [1 - r(F(x_k))] , & \text{gdy } x_k \in S \\ \frac{v(S)}{v(X)} [1 - r(F(x_k))] & \text{gdy } x_k \notin S \end{cases}$$

gdzie  $\nu$  jest pewną miarą /na przykład miarą Lebesgue'a/ oraz  $r(F)$  jest ściśle malejącą funkcją odwzorowującą zbiór wartości funkcji  $F$  w przedział  $(0,1)$ . W pracy [30] dowodzi się zbieżności takiego procesu poszukiwania ekstremum i bada się jego efektywność.

/A7/ W pracy [40] wykorzystuje się teorię statystyk pozycyjnych do lokalizacji ekstremum funkcji. Zbiór  $X$  dzieli się na dwa łączne zbiorы, powiedzmy  $X_1$  i  $X_2 = X - X_1$ . W każdym z tych zbiorów losuje się po  $n$  punktów i na podstawie wartości funkcji  $F$  w tych punktach weryfikuje się hipotezy:

$$H_1: \max_{x \in X_1} F(x) > \max_{x \in X_2} F(x)$$

$$H_2: \max_{x \in X_1} F(x) \leq \max_{x \in X_2} F(x)$$

W przypadku przyjęcia hipotezy  $H_1$  dalsze poszukiwanie ekstremum prowadzi się tylko w zbiorze  $X_1$ .

#### Kryteria oceny algorytmów

W przypadku sformułowania /4/ za lepszy można uznać na przykład ten algorytm, który przy ustalonym  $\epsilon_1$  prowadzi do mniejszego  $\epsilon$  /przy ustalonej liczbie kroków  $n$ / lub który przy ustalonym prowadzi przy ustalonej liczbie kroków  $n$  do mniejszego  $\epsilon_1$  lub który przy ustalonych  $\epsilon$  i  $\epsilon_1$  prowadzi do spełnienia warunku /4/ w mniejszej liczbie kroków iteracyjnych. Analogiczne oceny można sformułować w przypadku sformułowania /5/. Takie oceny algorytmów wymienione są w pracy [8].

Pewnym innym kryterium oceny algorytmu jest wartość prawdopodobieństwa, iż miara zbioru, na którym funkcja przyjmuje wartości mniejsze niż wartość znaleziona stanowi ustalony ułamek miary całego zbioru  $X$  /por. algorytm A1/. Kryterium to rozpatrywano w [8] oraz [74], a krytyka tego kryterium, podkreślająca jego nieprzydatność w przypadku wielowymiarowych zbiorów  $X$ , podana jest w pracy [31].

W pracy [52] jako kryterium oceny algorytmu przyjmuje się średnią liczbę kroków iteracyjnych potrzebną do osiągnięcia punktu w  $\varepsilon$ -otoczeniu /  $\varepsilon$  - ustalone/ punktu ekstremalnego.

W pracy [30] konstruuje się następujące kryterium oceny algorytmów: zapiszmy algorytm stochastyczny w postaci ciągu rozkładów prawdopodobieństwa:

$$P_0(S), P_1(S), \dots, P_n(S), \dots,$$

gdzie  $S$  jest elementem pewnej  $\delta$ -algebra podzbiorów zbioru  $X$ , natomiast  $P_n(S)$  jest prawdopodobieństwem, że w n-tym kroku zostanie osiągnięty punkt ze zbioru  $S$ .

Oznaczmy:

$$m_n(F) = \int F(x) P_n(dx).$$

Jako dopuszczalny definiuje się taki algorytm, dla którego istnieje granica:

$$\lim_{n \rightarrow \infty} m_n(F) = m(F).$$

Z dwóch algorytmów dopuszczalnych A i B za lepszy uważa się algorytm A, gdy

$$m_A(F) < m_B(F).$$

Szczegółowe rozważania w pracy [30] dotyczą cytowanego wyżej algorytmu A6.

Kryterium oceny algorytmu jest ściśle związane z charakterem rozwiązywanego zadania. Z tego powodu zbudowanie jedynego kryterium dla oceny wszelkich algorytmów nie jest ani możliwe, ani celowe.

#### Niektóre zagadnienia optymizacji statystycznej

Typowym zagadnieniem optymizacji statystycznej jest zagadnienie przedstawione w Przykładzie 1. Inne zagadnienia optymizacji statystycznej znane są w literaturze jako zagadnienia:

- optymizacji w warunkach zakłóceń /w warunkach szumów/ [55], [37], [28];
- aproksymacji stochastycznej [17], [64];
- poszukiwania ekstremum /lub zer/ linii regresji [32], [17], [63] lub ekstremum powierzchni regresji /powierzchni odpowiedzi - response surface/ [7], [5], [70].

Zagadnienie optymizacji statystycznej są ściśle związane z zagadnieniami algorytmów stochastycznych /niżej pokazujemy, na czym ten związek polega/. Jest to powodem, dla którego do załączonej bibliografii włączono również prace z tego zakresu.

Zagadnienia optymizacji statystycznej rozwiązuje się zwykle w następujący sposób iteracyjny: w punkcie  $x_k \in X$  rejestruje się wartość  $y(x_k)$  pewnej funkcji  $y(x)$ , przy czym na skutek działania czynników losowych wartość  $y(x_k)$  należy traktować jako realizację pewnej zmiennej losowej. Oznaćmy rozkład tej zmiennej losowej przez  $Q_k(x)$ . Identyczny wynik jak przy opisanej procedurze uzyskanoby stosując algorytm stochastyczny charakteryzujący się ciągiem rozkładów prawdopodobieństwa:

$$Q_0(x), Q_1(x), \dots, Q_n(x), \dots$$

W ten sposób zagadnienie optymizacji statystycznej sprowadza się do zagadnienia algorytmów stochastycznych. Różnica między tymi zagadnieniami polega na tym, że rozkłady  $Q_n(x)$  zwykle nie mogą być dowolnie ustalane, jak to ma miejsce w przypadku algorytmów stochastycznych. Pozostałe zagadnienia, a w szczególności zagadnienia statystycznej oceny uzyskanego wyniku, ustalania liczby eksperymentów, budowy estymatorów, występują zarówno w zagadnieniach optymizacji statystycznej jak i w teorii algorytmów stochastycznych.

Przegląd bibliograficzny

- [1] Бахарев, А.Т. (1965) : Нахождение экстремальных областей методом случайного поиска. Автоматика и вычислительная техника, 10, Рига.
- [2] BLUM, J.R. (1954) : Approximation methods which converge with probability one. Ann.Math.Stat., 25.
- [3] BLUM, J.R. (1954) : Multidimensional stochastic approximation methods. Ann.Math.Stat., 25.
- [4] BOX, G.E.P. (1957) : Evolutionary operation: A method for increasing industrial productivity. Appl. Statist., 6.
- [5] BOX, G.E.P. (1954) : The exploration and exploitation of response surfaces. Biometrics, 10.
- [6] BOX, G.E.P., JENKINS, G.M. (1962) : Some statical aspects of adaptive optimization and control. J.Roy.Stat.Soc., B, 24.
- [7] BOX, G.E.P., Wilson, K.B. (1951) : The experimental attainment of optimal conditions. J.Roy.Stat.Soc., B, 13.
- [8] BROOKS, S.H. (1958) : A discussion of random methods for seeking maxima. OR, 6, 2.
- [9] BROOKS, S.H. (1959) : A comparison of maximum - seeking methods. OR, 1, 4.
- [10] BUSH, R.R., MOSTELLER, F. (1955) : Stochastic models for learning. New York 1955, John Wiley and Sons.
- [11] CHUNG, K.L. (1954) : On a stochastic approximation method. Ann.Math.Stat., 25.
- [12] COCHRAN, W.G., BAVIS, M. (1965) : The Robbins-Monro method for estimating the median lethal dose. J.Roy.Stat.Soc., B, 27.
- [13] COMER, J.P. (1965) : Application of stochastic approximation to process control. J.Roy.Stat.Soc., B, 27.
- [14] COMER, J.P. (1964) : Some stochastic approximation procedures for use in process control. Ann.Math.Stat., 35.
- [15] DIETRICH, M., GOLIŃSKI, J. (1965) : Optymalna synteza maszyn. Przegląd Mechaniczny, 22, 1965.
- [16] DUPAČ, V. (1965) : A dynamic stochastic approximation method. Ann.Math. Stat., 36.
- [17] DVORETZKY, A. (1956) : On stochastic approximation. Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability. JoNeyman ed. . University of California Press, Berkeley and Los Angeles, 1956. Vol. I.
- [18] FABIAN, V. (1961) : A stochastic approximation method for finding optimal conditions in experimental work and in self-adapting systems in Czech . Apl. Mat. 6.

- [19] PAVREAU, R.R., FRANKS, P.N. (1958) : Statistical optimization. Second Intern. Conf. for Analog Computation, Strasbourg, Sept., 1958.
- [20] Герцбах, И.В. (1965) : Стационарный режим работы одномерного экстремального регулятора в условиях случайного дрейфа цели. Автоматика и вычислительная техника, 10, Рига.
- [21] GOLIŃSKI, J. (1965) : "Über eine optimale Maschinensynthese mittels der Monte-Carlo-Methode." Сборник докладов Международной конференции по "Механизмам и машинам", Болгария, Варна, 1965, Том I.
- [22] GOLIŃSKI, J. (1965) : O optymalnej syntezie maszyn metodami Monte-Carlo. Archiwum Budowy Maszyn, T. XII, zeszyt 3.
- [23] Гурин, Л.С. (1963) : Опыт применения метода Монте-Карло для нахождения на ЭВМ экстремальных значений функции. Бюр. вычисл. мат. и вычисл. техн., Машгиз, М. 1963.
- [24] Гурин, Л.С. (1964) : Оптимизация в стохастических моделях. Журнал вычислительной математики и математической физики, Т.4, № 2.
- [25] Гурин, Л.С. (1964) : К вопросу об оптимизации в стохастических моделях. Журнал вычислительной математики и математической физики, Т.4, № 6.
- [26] Гурин, Л.С. (1965) : К вопросу о сравнительной оценке различных методов оптимизации. Автоматика и вычислительная техника, 10, Рига.
- [27] Гурин, Л.С., Лобач, В.П. (1962) : Комбинация метода Монте-Карло с методом скорейшего спуска при решении некоторых экстремальных задач. Журнал вычислительной математики и математической физики, Т 2, № 3.
- [28] Гурин, Л.С., Растрогин, Л.А. (1965) : О сходимости метода случайного поиска в обстановке помех. Автоматика и телемеханика, 1965, №9.
- [29] Харланов, Б.М. (1964) : Некоторые теоремы о вероятностном поиске в детерминированном поле. Вестн. ЛГУ, № 3.
- [30] Харланов, Б.П. (1965) : Об одном алгоритме стохастического поиска максимума в детерминированном поле. Труды Матем.Инст. им. В.А. Стеклова, LXXIX.

- [31] Hooke, R. and Jeeres, T.A. (1958) : Comments on Brooks' discussion of random methods. *QR*, 6.
- [32] Jilek, M., Likar, O. (1960) : Table of random sample sizes needed for obtaining non-parametric tolerance regions. *Zast. Mat.* V, 2.
- [33] Kallianpur, G. (1954) : A note on the Robbins-Monro stochastic approximation method. *Ann. Math. Stat.*, 25.
- [34] Kesten, H. (1958) : Accelerated stochastic approximation. *Ann. Math. Stat.*, 29.
- [35] Kiefer, J., Wolfowitz, J. (1952) : Stochastic estimation of the maximum of a regression function. *Ann. Math. Stat.*, 23.
- [36] Коган, Я.А. (1965) : Нижние и верхние оценки для риска при оптимальном методе поиска минимума параболической функции при наличии помех. В сб.: "Техническая кибернетика", Москва 1965.
- [37] Kushner, H.J. (1962) : Hill climbing methods for the optimization of multi-parameter noise disturbed systems. *Joint Automat. Control Conf.*, NY., 1962.
- [38] Моцкус, И.В. (1963) : О применении метода Монте-Карло для решения многоэкстремальных и комбинаторных задач. В сб. "Доклады научнотехн. совещания: Общие вопросы применения вероятностных и статистических методов". Вып. I. Киев. Гос. изд-во техн. лит. УССР, 1963, 30-4 .
- [39] Моцкус, И.Б. (1964) : Об одном способе распределения случайных испытаний при решении многоэкстремальных задач. *Журнал вычислительной математики и математической физики*, Т 4, № 2.
- [40] Моцкус, И.Б. (1965) : Об одной последовательной процедуре статистического решения экстремальных задач. *Автоматика и вычислительная техника*, 10, Рига.
- [41] Morecki, A., Goliński, J., Wasersztrum, Z. (1965) : Исследование колебательных систем с двумя степенями свободы с сопротивлением пропорциональным квадрату скорости, методом Монте-Карло. Сборник докладов Международной конференции по "Механизмам и машинам", Болгария, Варна, 1965, Т II.
- [42] Музениекс, В.А., Растигин, Л.А. (1964) : Экстремальное регулирование непрерывных многопараметрических систем методом случайного поиска. *Изв. АН СССР, Техническая Кибернетика*, 1964, № .

- [43] Муцениекс, В.А., Растигин, Л.А. (1964) : Продвижение к цели методом случайного поиска. Известия АН Латв. ССР, Сер. физ.- техн., 1964, №1 .
- [44] MUNSON, J.K., RUBIN, A.I. (1959) : Optimization by random search on the analog computer. IRE Trans., EC-8, 1959, N.2, pp. 200-203.
- [45] Новоселов, Ю.Н. (1965) : К вопросу о выборе области поиска. Автоматика и вычислительная техника, 10, Рига.
- [46] ODERFELD, J. 1965 : *Statistische Synthese der Maschinen*. Сборник докладов Международной конференции по "Механизмам и машинам", Болгария, София, 1965, Том II.
- [47] Разникова, Т.Л. (1965) : Об одном алгоритме машинного поиска экстремума функции. Журнал вычислительной математики и математической физики, Т 5, № 4.
- [48] Растигин, Л.А. (1960) : Экстремальное регулирование методом случайного поиска. Автоматика и телемеханика, 1960, № 9.
- [49] Растигин, Л.А. (1962) : Применение метода случайного поиска для оценки напряженного состояния в пластинах. Труды Всесоюзной конференции по теории пластин и оболочек, Киев 1962.
- [50] Растигин, Л.А. (1963) : О сходимости метода случайного поиска при экстремальном регулировании многопараметрических систем. Автоматика и телемеханика, 1963, № 1.
- [51] Растигин, Л.А. (1963) : Метод случайного поиска в задачах многопараметрической оптимизации. Автоматика и вычислительная техника, 6, Рига.
- [52] Растигин, Л.А. (1964) : Сравнение методов Гаусса, Монте-Карло и случайного поиска при решении систем линейных алгебраических уравнений. Автоматика и вычислительная техника, 7, Рига.
- [53] Растигин, Л.А. (1964) : Об одной шаговой системе многопараметрической оптимизации, работающей по методу случайного поиска. Автоматика и вычислительная техника, 7, Рига, 1964

- [54] Растрогин, Л.А. (1965) : Влияние блуждания цели на поведение системы экстремального управления. Автоматика и вычислительная техника, 9, Рига.
- [55] Растрогин, Л.А. (1965) : Блуждание многопараметрической системы, оптимизируемой в обстановке помех. Автоматика и вычислительная техника, 10, Рига.
- [56] Растрогин, Л.А. (1965) : Проблемы поиска. Автоматика и вычислительная техника, 10, Рига.
- [57] Растигин, Л.А. (1965) : Некоторые статистические алгоритмы глобального поиска. Автоматика и вычислительная техника, 10, Рига.
- [58] Растрогин, Л.А. (1965) : Случайный поиск в задачах оптимизации многопараметрических систем. Рига 1965.
- [59] Растрогин, Л.А., Янсон, Б.А. (1964) : О стравнении эффективности работы методов градиента и случайного поиска в районе цели. Автоматика и вычислительная техника, 7, Рига.
- [60] Растрогин, Л.А., Рипа, К.К. (1963) : Моделирование обучения на ЦВМ при экстремальном регулировании методом случайного поиска. Автоматика, 1963, № 3, Киев.
- [61] Растрогин, Л.А., Рипа, К.К. (1964) : Моделирование обучения при экстремальном регулировании многопараметрических систем методом случайного поиска. Автоматика, 1964, № 5, Киев.
- [62] Растрогин, Л.А., Рипа, К.К. (1965) : Непрерывный алгоритм самообучения при многопараметрической оптимизации методом случайного поиска. Автоматика и вычислительная техника, 10, Рига.
- [63] ROBBINS, H., MONRO, S. (1951) : A stochastic approximation method. Ann. Math. Stat., 22.
- [64] SACKS., J. (1958) : Asymptotic distribution of stochastic approximation procedures. Ann.Math.Stat., 29.
- [65] SARHAN, A.F., GREENBERG, B.G. ed (1962) : Contributions to order statistics. New York(1962), John Wiley and Sons.
- [66] SCHMETTERER, L. (1961) : Stochastic approximation. Proc. 4th Berkeley Symp.Math.Statist. and Prob., 1, 587-609.

- [67] Трахтенберг, Л.С. (1965) : Некоторые проблемы связанные с решением экстремальных задач оптимального проектирования поисковыми методами. Автоматика и вычислительная техника 10, Рига.
- [68] WETHERILL, G.B. (1963): Sequential estimation of quantal response curves. J.Roy.Stat.Soc., B, 25.
- [69] WALD, A. (1943): An extension of Wilks' method for setting tolerance limits. Ann.Math.Stat., 14.
- [70] WILDE, D.J. (1964): Optimum seeking methods. Prentice-Hall, Inc. 1964.
- [71] WILKS, S.S. (1941): Determination of sample sizes for setting tolerance limits. Ann.Math.Stat., 12.
- [72] WOLFOWITZ, J. (1952): On the stochastic approximation method of Robbins and Monro. Ann. Math.Stat., 23.
- [73] WOLFOWITZ, J. (1956): On stochastic approximation methods. Ann.Math. Stat., 27.
- [74] ZIELIŃSKI, R. (1965): On the Monte-Carlo evaluation of the extremal value of a function. Algorytmy, Vol. II, N. 4.

#### STOCHASTIC ALGORITHMS IN PROBLEMS OF OPTIMIZATION

##### Summary

In the paper a short review of basic concepts of the stochastic methods in optimization's problems is given.

Cenzit z 15.6. 2