About the Dialog method for solving optimization problems.

[Cybernetics.- 1975.-№ 4.]

In many modern optimization problems, the optimum point is found to lie on the boundary of the domain of the function under study. Especially many examples of such problems are provided by Economics, where optimization problems are often set and solved using linear models.

Even in the linear case, it is usually not an easy task to accurately define the boundaries of a domain. Difficulties arise here primarily due to the high dimensionality inherent in most practically important economic problems (for example, the problem of optimizing the inter-industry balance). In addition, the coefficients of linear equations that define the boundaries of the domain can often be obtained only on the basis of statistical processing and averaging a huge number of primary data obtained as a result of experiments or calculations of a technological nature. This is the case, for example, when determining technological coefficients in linear macroeconomic models.

In classical statements of optimization problems, the work that must be spent to accurately define the boundaries of the domain is not taken into account. However, this work is far from fully amenable to the process of formalization and automation and often turns out to be much more cumbersome and lengthy than subsequent computer solutions to the optimization problem itself.

The situation is compounded by the fact that the initial data for optimization problems are usually prepared by specialists who have approximate intuitive ideas about the border area where the optimum should be located. Meanwhile, the optimization mathematician does not have the intuition that follows from a meaningful (rather than formal) statement of the problem, the initial data of which would equally accurately determine the entire boundary. However, it is obvious that, for example, in a classical linear programming problem, it is pointless to define hyperplanes that define sections of the boundary far from the area of the expected optimum. Feeling that their time is being spent on meaningless work, problem-setters lose interest in it and, as a result, perform their part of the work carelessly. As a result, the original data is not accurately calculated, and new technological opportunities (especially the most new and promising ones, since they usually require more labor) are simply not taken into account. In other words, the boundary of the domain, which is immutable from the point of view of the optimization mathematician, can actually undergo significant variations. The size of these variations means that the effect obtained with their help can significantly exceed the effect obtained by primary optimization. It is not surprising that problem-setters in such conditions are sometimes skeptical of the very idea of optimization, not to mention the results of specific optimization calculations.

In most cases, in order to practically effectively solve optimization problems, it is necessary that the methods used satisfy two additional conditions. First, they should minimize (to a reasonable extent) the work of problem-setters, avoiding, if possible, wasting it on the preparation of unnecessary primary information. Second, they should stimulate and support the creative initiative of problem-setters to search for fundamentally new informal opportunities to improve the solution by changing the boundary of the area (usually only in the area of the previously found optimal point). In the economy, such opportunities arise most often due to fundamentally new technological ideas and solutions.

The method that meets the conditions put forward, due to the nature of these conditions, must necessarily be interactive: iterative improvement of the solution should occur as a result of a dialogue between the computer and a collective of experts problem-setters who have special meaningful knowledge about the problem being solved. The solution process goes through the following stages: initially, the problem-setters roughly delineate the boundaries of the area in which the solution is sought, allowing them to be more refined only in the area in which the optimum should be located in accordance with their view; then the computer solves the optimization problem by finding the exact position of the optimum point on the specified boundary. Problem-setters are given a new (updated) border area where the obtained point is located, as well as, possibly, other information that guides them in the direction of informal attempts to further improve the obtained solution. Proposals received from them are evaluated by the computer and either accepted or

rejected (It is possible that previously rejected proposals will be accepted if they are repeated in a few subsequent steps.). The process ends when the flow of suggestions runs out, or when the time allotted for solving the problem runs out. The amount of time allocated to optimization is usually determined quite naturally when considering the content meaning of the task, which is usually a task of planning and managing in real time cycles.

Practice shows that in order to effectively engage experts problem-setters in a dialogue with computer, the time for which the computer evaluates each incoming proposal should not exceed a certain threshold.

The value of this threshold is determined by the time during which the person who gave the proposal retains interest in it and does not switch to another job. In a specific macroeconomic task, which will be discussed below, this threshold can be estimated at 15-20 minutes. It goes without saying that the ideal would be an instantaneous computer response, but for any serious optimization problems that are on the border of the capabilities of the computers used, this ideal is unattainable.

The need to have a small delay in the system's response to incoming proposals requires the development of optimization methods that make it much easier and faster to refine (by changing the boundary) the solution that has already been obtained than to find the solution again. At first glance, this condition is satisfied by a number of classical optimization methods, such as gradient optimization. However, it should be emphasized that in most modern optimization problems, the concept of "area of the optimum point" used above is by no means equivalent to the classical concept of "small area". For example, for the task of optimizing the inter-industry balance, it is natural to assume that arbitrarily large changes in the technological coefficients of one of the technologies under consideration do not lead us out of the area of the previously found optimum. Therefore, creating effective dialog methods for solving optimization problems requires some improvement of classical methods.

Let's take the classical linear programming problem as an example.

Find the minimum (maximum) of a linear function $z=c_1x_1+c_2x_2+\ldots+c_nx_n+d$ under constraints:

$y_1=a_{11}x_1+a_{12}x_2+\ldots+a_{1n}x_n+b_1\geqslant 0,$

$y_2=a_{21}x_1+a_{22}x_2+\ldots+a_{2n}x_n+b_2\geqslant 0,$

$y_m=a_{m1}x_1+a_{m2}x_2+\ldots+a_{mn}x_n+b_m\geqslant 0,$

where $m\geqslant n$. As is known, the desired minimum (maximum) point, if it exists, coincides with the intersection point of some n hyperplanes $y_{i_1}=0, y_{i_2}=0,\ldots, y_{i_n}=0$. Let it be hyperplanes $y_1=0, y_2=0,\ldots, y_n=0$. Let's perform an affine transformation with the matrix

$$A=\begin{Vmatrix} a_{11} & a_{12} & \ldots & a_{1n} & b_1 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & \ldots & a_{nn} & b_n \\ 0 & 0 & \ldots & 0 & 1 \end{Vmatrix}$$

so that vector $x=\begin{Vmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ 1 \end{Vmatrix}$

transforms into a vector $x'=\begin{Vmatrix} x_1' \\ x_2' \\ \vdots \\ x_n' \\ 1 \end{Vmatrix}=Ax$

The desired optimum point lies at the new origin of coordinates $x_1=x_2=\ldots=x_n=0$.

The transformation $x \to x' = Ax$ translates an arbitrary linear function $p'x = p_1 x_1 + p_2 x_2 + \ldots + p_n x_n + p_0$ into a linear function $p'x' = px = pA^{-1}x'$. Thus, the vector p of the coefficients of a linear function is transformed into a vector $p' = pA^{-1}$.

The value of the optimized function $z = cx$ at the optimum point will be, obviously, equal to CA^{-1},

$$\begin{Vmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{Vmatrix}$$

or, what is the same, the last (n+1)-th element of the vector cA^{-1}, where $c = \|c_1, c_2, \ldots, c_n d\|$ - is the vector of coefficients of the linear function z.

In accordance with the general idea of the dialog method, initial constraints $y_1 \geqslant 0, \ldots, y_m \geqslant 0$ can be set with certain errors (approximately). After finding the optimum point under these constraints, we begin to analyze the possibility of changing them in order to improve the achieved optimum. Note that this can only be achieved by changing the constraints that create a new coordinate system (in this case, the constraints $y_1 \geqslant 0, \ldots, y_n \geqslant 0$), since all possible solutions lie in a cone $y_1 \geqslant 0, \ldots, y_n \geqslant 0$.

Suppose that for some $i (1 \leqslant i \leqslant n)$, the constraint $y_i = a_i x \geqslant 0$ was replaced by the constraint $y_i + \Delta y_i = (a_i + \Delta a_i)x \geqslant 0$, where a_i - is a vector $\|a_{i1}, a_{i2}, \ldots, a_{in}, b_n\|$, and Δa_i - is its arbitrary increment. By replacing the coordinate hyperplane $y_i = 0$ with a new hyperplane $y_i + \Delta y_i = 0$, we move from the affine transformation $x \to x'$ with the matrix A_i to the affine transformation with the matrix $A + \Delta_i A$. Here by $\Delta_i A$ is denoted the "single-line" matrix

$$\begin{Vmatrix} 0 & 0 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \Delta a_{i_1} & \Delta a_{i_2} & \ldots & \Delta a_{i_n} & \Delta b_i \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 0 & 0 \end{Vmatrix}$$

where all the rows except the i-th are filled with zeros.

In order for the beginning of the new coordinate system to coincide with the new optimum point, it is sufficient that two conditions are met: 1) the free terms of linear functions $y_{n+1}, \ldots, y_m$ in the new coordinate system must be non-negative; 2) all coefficients for unknowns of function $z = cx$ in the new coordinate system have the same sign (plus for the minimum and minus for the maximum). But since the vectors of coefficients of linear functions are transformed using the inverse matrix, the verification of both conditions is reduced to finding a new inverse matrix $(A - \Delta_i A)^{-1} = A^{-1} + \Delta_i (A^{-1})$.

Using the fact that matrix $\Delta_i A$ is "single-line", we can show that

$$\Delta_i (A^{-1}) = A^{-1} \frac{1}{1-\alpha} D_i = \frac{1}{1-\alpha} A^{-1} \Delta_i A A^{-1} \qquad (1)$$

Here, D_i denotes the "single-line " matrix $\Delta_i A A^{-1}$, and α denotes the element of this matrix that stands at the intersection of the i-th row and the i-th column. Calculating the matrix $\frac{1}{1-\alpha} D_i$ requires performing $(n+1)^2$ multiplications, $n(n+1)$ additions and $n+1$ divisions, i.e. $2(n+1)^2$ arithmetic operations. When multiplying matrices A^{-1} and $\frac{1}{1-\alpha} D_i$, $(n+1)^2$ more multiplications are added to them, so that the total number of operations when calculating the increment of $\Delta_i (A^{-1})$ is $3(n+1)^2$, not counting the operation of subtracting α from 1.

Knowing the increments of the inverse matrix, using additional $2n+1$ operations we can also calculate the increment of the vector of coefficients of any linear function. This makes it possible to check the above two conditions. If they are met, we find the increment Δz of the value of the criterion

$$z: \Delta z = c\,\Delta_i(A^{-1}) \begin{Vmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{Vmatrix} .$$

If $\Delta z<0$ - in the case of searching for the minimum and $\Delta z>0$ - in the case of searching for the maximum, we achieve the desired improvement of the optimum, otherwise it either worsens or the value of the optimum remains unchanged.

In the case when at least one of the above two conditions is not met, in accordance with the general idea of the simplex method, we make the transition to a new coordinate system (with the beginning at the optimum point) by successive modified jordan transformations. Since any such transformation is defined by a "single-line" matrix, the formula (1) can be used for calculating the inverse matrix at each of the steps. A similar sequential method for calculating the inverse matrix can also be used for finding the initial value of the optimum (with unspecified boundary conditions).

A similar method of refining the solution can also be used in the case when the limiting hyperplanes of $y_{n+1},...,y_m$ are subjected to refinement, although, of course, in this case, the initial value of the optimum found does not obviously improve. When solving specific practical problems, the intuition of problem setters usually avoids unnecessary work on clarifying conditions that do not change the value of the optimum.

It should be emphasized that in a large number of applications of linear programming to economic problems, many constraints are determined by the features of the technology used and can therefore be changed purposefully, and not just refined. Therefore, due to the choice (or invention) of a suitable technology, the problem setter can often carry out a conscious multi-step improvement of the initially found optimum. At the same time, a simple refinement of the constraints, being an objective process that does not depend on the will of the problem setter, can change the initial value of the optimum in any direction. Conscious management of the process of improving the optimum should also rely on the intuition of the problem setters, due to which it is possible to avoid many obviously useless attempts to change the initial constraints.

A good illustration of the advantages of the dialog method can be the problem of optimizing the intersectoral balance. In each of the specified n industries, one specific technology is initially fixed. These technologies determine the matrix $A=\|a_{ij}\|$ of direct product costs (element a_{ij} of the matrix A determines the amount of product produced by the $i-th$ industry and which is spent in the $j-th$ industry on the production of a unit of product produced by it). We denote by $b=\|b_1,b_2,...,b_n\|$ the vector of direct costs of some resource external to the system of products under consideration, for example, labor costs. b_j here denotes the costs of this resource in the $j-th$ industry for the production of a unit of product produced by this industry. Like the elements a_{ij} of the matrix A, the elements b_j of the vector b are determined by a given set of technologies.

From the theory of static macroeconomic models, the vector $b^*=\|b_1^*,b_2^*,...,b_n^*\|$ of the total costs of the resource under consideration is determined by the formula

$$b^*=bA^*=b(E-A)^{-1} . \qquad (2)$$

The component b_j^* of the vector b^* is equal to the total cost of the resource (taking into account all its costs in other industries) required to produce a unit of the $j-th$ product. Denote by

$$c=\begin{Vmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{Vmatrix}$$

the vector of the final (or net) output. The component c_i of this vector determines the total amount of the product produced by the $i-th$ industry, minus the part of it that is consumed in the production of all other products.

The amount of external resource that must be consumed to ensure a net output of c is determined by the formula

$$z=b^* c=b(E-A)^{-1}c \quad . \qquad (3)$$

Now let's assume that there are not one, but several possible technologies for each industry. The replacement of technology in the $j-th$ industry leads to a change in the $j-th$ column of the matrix A and the $j-th$ element of the vector b and causes a change in the value of the criterion z. The task is to determine such a choice of technologies at which the value of the z criterion reaches a minimum. It can be reduced to a linear programming problem [1], however, not in the original space of dimension n, but in a space whose dimension is equal to the total number of all possible technologies. In the original space, as can be seen from formula (3), the optimization problem is not linear.

The author [2] proved that the desired minimum can be obtained as a result of successive replacement of technologies (one for each step) so that each replacement reduces the value of the z criterion. The impossibility of its further reduction by any change in one of the technologies means that the absolute minimum has been reached.

Since the increments of the matrix A, and hence the matrix $E-A$, are "single-column" (all the columns of the increment ΔA of the matrix A, with the exception of one, are zero), then by analogy with the formula (1) we get

$$\Delta A^*=\frac{1}{1-\alpha}DA^*=\frac{1}{1-\alpha}A^*\cdot\Delta A\cdot A^* \quad , \qquad (4)$$

where α - is an element in a non-zero column of the matrix D, standing at the intersection of this column with the main diagonal of the matrix.

Increment of the criterion value

$$\Delta z=(\Delta b+bA^*\Delta A)(A^*+\Delta Ab)c \quad . \qquad (5)$$

The method of solving the problem is as follows. After selecting the initial set of technologies and defining matrices $A, A^*=(E-A)^{-1}$ for it, as well as vector b, we organize a dialogue with the problem setters. They make one after another elementary proposals aimed at saving the resource under consideration. The elementary nature of the proposal means that only one technology is being replaced. Each of the submitted proposals is checked on the computer according to the formula (5). If $\Delta z<0$, the proposal is accepted, otherwise it is rejected. In principle, it is possible that some technology, replaced by another, can be introduced again at the next steps. However, there is no need to store all the steps taken earlier in the computer's memory, it is enough to remember only the values A, A^*, b and z obtained at the last step.

Calculations using the formulas (4) and (5) are performed much faster than finding the initial value of the matrix A^*, since the number of operations performed has the order of the square of the dimension of the problem (n), and not the cube of the dimension, as is the case when calculating the matrix A^*. Therefore, even with a sufficiently high dimension (n = 1200), the verification of any elementary proposal takes only 15-18 minutes of work, even for such a relatively low-performance machine as Minsk-32. This indicator is quite consistent with the above requirements for the response time of the system, which ensure effective dialogue.

Note that the general approach described above to solving linear programming problems in the interactive mode is not applicable for the problem just considered. In fact, when linearizing this problem, the introduction of each new technology does not mean an increase in the number of constraints, but an increase in the dimension of the space (the number of variables) in which the

optimization is performed. The case of dimension growth is not accounted for in the method described above. It requires additional consideration.

BIBLIOGRAPHY

1. Lancaster K. Mathematical Economics. - Moscow: Sov. Radio, 1972.
2. Glushkov V. M. On sequential optimization in linear macroeconomic models / / Control systems and machines.— 1973.— N° 4.