

Запропоновано ітераційні алгоритми розв'язання зворотної задачі, представлені у вигляді задачі квадратичного програмування, розроблені шляхом модифікації алгоритмів, заснованих на механізмі зворотних обчислень. Ітераційні алгоритми полягають у послідовній зміні значень аргументів за допомогою ітераційних формул до досягнення функцією величини, найбільш відповідної обмеженню. При цьому розглянуто два варіанти вирішення задачі: шляхом визначення найкоротшої відстані до лінії заданого рівня, що визначається обмеженням, і шляхом руху вздовж градієнта. Даний підхід також був адаптований для вирішення оптимізаційних завдань нелінійного програмування більш загального вигляду. Розглянуто вирішення чотирьох завдань: формування випуску продукції та складських витрат, оптимізація портфеля цінних паперів та складських витрат при заданому обсязі закупівель. Показано, що одержувані при використанні ітераційних алгоритмів рішення узгоджуються з результатом використання класичних методів (множників Лагранжа, штрафів), стандартної функції математичного пакету MathCad. При цьому найбільша ступінь відповідності була отримана за допомогою методу на основі побудови лінії рівня, метод на основі руху вздовж градієнта є більш універсальним.

Перевагою алгоритмів є більш проста комп'ютерна реалізація ітераційних формул, можливість отримати рішення за менший час в порівнянні з відомими методами (наприклад, методом штрафів, що вимагає багаторазової оптимізації модифікованої функції зі змінною штрафного параметра). Алгоритми можуть бути також використані для вирішення інших завдань нелінійного програмування представленого виду.

Стаття може бути корисна для фахівців, які здійснюють вирішення завдань в області економіки, а також розробку програмних систем підтримки прийняття рішень

Ключові слова: зворотні обчислення, оптимізація функцій, нелінійне програмування, градієнтний метод, зворотна задача

DEVELOPMENT OF ITERATIVE ALGORITHMS FOR SOLVING THE INVERSE PROBLEM USING INVERSE CALCULATIONS

E. Gribanova

PhD, Associate Professor
Department of Automated Control System
Tomsk State University
of Control Systems and Radioelectronics
Lenina ave., 40, Tomsk, Russia, 634050
E-mail: geb@asu.tusur.ru

Received date 10.05.2020

Accepted date 03.06.2020

Published date 30.06.2020

Copyright © 2020, E. Gribanova

This is an open access article under the CC BY license
(<http://creativecommons.org/licenses/by/4.0>)

1. Introduction

In the study of socioeconomic systems, there is a need to solve both direct and inverse problems. If the solution of direct problems allows you to evaluate the performance of an object based on the available characteristics, the solution of inverse problems provides an opportunity to determine a set of characteristics to achieve a given performance. So, for example, organizations have the problems of determining a set of indicators for generating profit, revenue, sales, integral characteristic of the enterprise activity [1]. The relationship of indicators can form a tree, at each level of which a solution to a separate inverse problem is required. In this case, the solution of inverse problems due to their instability requires the determination of additional conditions (regularization), which determines the variety of approaches to solving such problems, the development of which modern research is also devoted to.

The relevance of this research area is associated with the wide spread of inverse problems in various fields (economics, physics, astronomy, etc.), as well as their high applied significance. Thus, in the field of economics, the solution of inverse problems allows determining control actions to achieve a given state of an economy object and thus forming optimal management decisions.

2. Literature review and problem statement

In [2], as additional conditions for solving economic analysis problems, expert information is used: coefficients of the relative priority of indicators, directions of changes in indicators. The use of expert information requires the involvement of a specialist, which leads to additional costs of time and financial resources. In addition, the resulting decision will be subjective and determined by the degree of the expert's professionalism. The most common types of regularization based on the deviation of the obtained solution from the initial one are Tikhonov regularization [3–5] and Manhattan distance regularization [6, 7].

Let x_i be the i -th performance indicator of an economic object, y – the resulting performance indicator of the object, $h(x_i)$ – the dependence function of the indicators x_i and the resulting indicator y ($y=h(x)$). The problem is to determine changes in the initial characteristics Δx_i to achieve the given value of the resulting indicator $y+\Delta y$.

When applying Tikhonov regularization, the problem can be represented as follows (μ is the regularization parameter):

$$Q(\Delta x) = (h(x + \Delta x) - y - \Delta y)^2 + \mu \sum_{i=1}^n \Delta x_i^2 \rightarrow \min. \quad (1)$$

In the case of Manhattan distance regularization, instead of the sum of squares of argument changes, in the formula (1) the sum of modules of argument changes is used.

The solution of the problem (1) requires finding the regularization parameter μ , which is a separate problem that requires choosing the method of searching for μ [8], which determines the result. In this regard, consideration may be given to representation as a constrained optimization problem. In this case, the options of the objective function can be considered: minimization of the sum of modules of argument changes, minimization of the sum of squares of argument changes.

In the case of minimization of the sum of modules of argument changes, the problem has the following form [9]:

$$f(\Delta x) = \sum_{i=1}^n |\Delta x_i| \rightarrow \min, \quad h(x + \Delta x) = y + \Delta y. \quad (2)$$

As a result of solving this problem, the values of some argument changes are equal to zero, so the best features for changes can be selected.

In the case of minimization of the sum of squares of argument changes, the problem has the following form:

$$f(\Delta x) = \sum_{i=1}^n \Delta x_i^2 \rightarrow \min, \quad h(x + \Delta x) = y + \Delta y. \quad (3)$$

Representation of the problem in this form can be determined by the need to achieve the given value of the resulting indicator so that changes in input parameters are as close to zero as possible. This method of solution reflects the desire to minimize the adjustment of input controlled indicators, and, consequently, to reduce resource expenditures for activities associated with changes in indicators compared to their CUrrent state.

The problem (2) can be represented as a linear programming problem, the solution of which is reduced to forming an equation for the arguments with the largest absolute numerical values in the constraint [9]. Solving the problem (3) is a more complex problem. If, with a small number of elements, the problem can be solved analytically using the Lagrange multiplier method, then when increasing the dimension and implementing software applications, it becomes necessary to implement numerical solution algorithms.

The classical methods for solving the nonlinear optimization problem (3) are the penalty method and Lagrange multiplier method. In the Lagrange multiplier method, the modified function includes unknown parameters – Lagrange multipliers λ [12]:

$$Z(\Delta x, \lambda) = \lambda(h(x + \Delta x) - y - \Delta y) + \sum_{i=1}^n \Delta x_i^2 \rightarrow \min. \quad (4)$$

To optimize the function (4), a system of equations is formed in which the partial derivatives are equal to zero, and the conditions of complementary nonrigidity are also included. Due to the definition of additional variables λ , the dimension of the problem increases, which is a drawback of this method.

In the penalty method, there are multiple optimizations of the modified function with a sequential change in the penalty parameter R :

$$L(\Delta x) = R(h(x + \Delta x) - y - \Delta y)^2 + \sum_{i=1}^n \Delta x_i^2 \rightarrow \min. \quad (5)$$

This classical solution scheme can be modified taking into account the specifics of the problem being solved. For example [10] addresses the solution to the multicriteria optimization problem. The authors of [11] present a solution to the two-level optimization problem using the penalty method.

The main disadvantage of the penalty method is the need to perform multiple unconstrained optimizations of the function. As the modified function (5) includes two components (the sum of increment squares and the compliance of the function h with the given value of the resulting indicator), optimization may take a long time, and gradient methods may be ineffective.

As a way to overcome this difficulty, the authors propose algorithms for solving the problem without applying the penalty parameter, based on the Kuhn-Tucker conditions. As a result, the solution of the problem is reduced to solving systems of equations. So, in [13], three systems of linear equations are solved at each iteration to search for the direction of argument changes, after which a linear search is performed in the given direction. In [14], the solution of the nonlinear programming problem is reduced to solving the linear programming problem by the simplex method. However, the proposed method can be used only with a linear constraint. Also, the Zoutendijk method [15] is used to solve nonlinear optimization problems with inequality constraints, which includes solving a linear programming problem to determine the search direction, followed by optimization of the function by moving along the selected direction.

Another area of research in the field of solving nonlinear programming problems is the use of evolutionary algorithms [16]. In particular, the use of recurrent neural networks for solving the nonlinear optimization problem is considered [17]. However, such algorithms require generating a large number of population agents, performing multiple operations to select them and forming new individuals. The use of neural networks requires the implementation of network learning algorithms. Therefore, the development of the algorithm, the implementation of multiple optimizations of the function can also take a significant amount of time and computing resources.

Some authors also consider a combination of two methods, for example, in [18], the Zoutendijk method was used in conjunction with the heuristic method.

To eliminate the indicated drawbacks of the methods, a method for solving problems (3) using inverse calculations was developed. Two approaches to problem solution were identified [19, 20]:

1. Solving the problem by determining the minimum distance to the line of the given level. The essence of this method is to move from the starting point, the coordinates of which are determined by the values of the variables x , to the point on the line of the given level by the shortest path. The length of this path is equal to the length of the perpendicular lowered from the starting point to the line of the given level. So, point A in Fig. 1 corresponds to the initial values of profit (equal to 2 CU) and cost (equal to 15 CU) (output value is profit). Fig. 1 also presents the line of the given profit level (0.2). Fig. 2 shows the options of argument changes, providing a profit value of 0.2. The points forming a Pareto effective set are connected by a line. The solution to the problem is an element that provides the minimum sum of two criteria. In Fig. 1, the solution to the problem is represented by point B obtained by crossing the line of the given profit level and the perpendicular lowered from point A to that line.

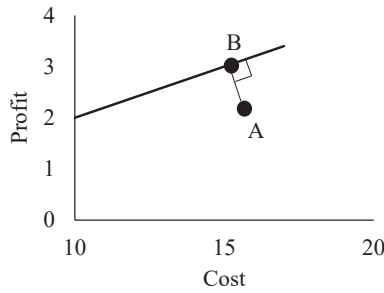


Fig. 1. Solution of the problem by crossing the perpendicular and the level line

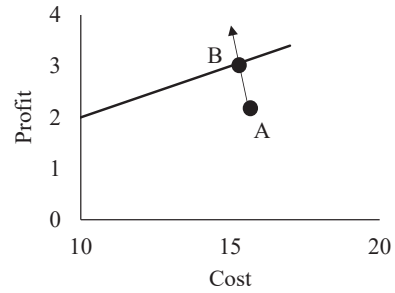


Fig. 3. Solution of the problem by moving along the gradient

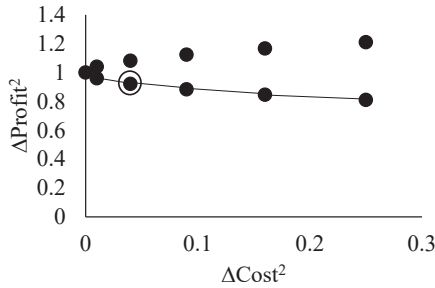


Fig. 2. Options of argument changes

The algorithm for problem solution includes expressing one of the arguments of the constraint function and equating the value of the partial derivative of the dependence function of the arguments to the relation of argument changes. As a result, a system that includes equations for argument relations and the resulting indicator using the dependence function $h(x)$ are formed. So, for the problem in Fig. 1, the system of equations has the following form (the expression of the *Profit:Profit=0.2·Cost* variable is satisfied):

$$\begin{cases} \frac{\Delta Cost}{\Delta Profit} = -0.2; \\ \frac{2 + \Delta Profit}{15 + \Delta Cost} = 0.2. \end{cases}$$

The solution of the system: $\Delta Cost = -0.192, \Delta Profit = 0.962$.

In the case the dependence function of the arguments is nonlinear, the problem is solved iteratively, the obtained solution is used to calculate the new value of the partial derivative.

The main disadvantage of this algorithm is the need to form a dependence function of the arguments, which is not possible for some problems. In addition, when solving a problem, it is necessary to take into account the range of admissible values of the arguments of the generated function (for example, the radicand cannot be negative).

2. Solving the problem by moving along the gradient of a function (gradient method). The main idea is to change the arguments of the function according to the values of the elements of the gradient vector of the constraint function until the specified value is reached (Fig. 3).

The system of equations in this case is as follows:

$$\begin{cases} \frac{\Delta Cost}{\Delta Profit} = \frac{2}{15^2}; \\ \frac{2 + \Delta Profit}{15 + \Delta Cost} = 0.2. \end{cases}$$

The solution of the system: $\Delta Cost = -0.13, \Delta Profit = 0.974$.

With a linear constraint function, analytical formulas can be obtained that will be identical for the two approaches considered. At the same time, high compliance of the solution obtained using the given methods with that using mathematical packages is achieved [9, 19]. However, under nonlinear constraints, the following disadvantages of the methods were revealed:

1. For some functions, there was a significant difference between the obtained solution and the optimal one, or the solution was not found (for example, the direction of the gradient vector in the initial point allows reaching the specified value of the constraint function).

2. Multiple solutions of the system of equations, and, accordingly, implementation of the corresponding numerical methods (for example, the Newton method) are required, which complicates the solution process and also increases the solution time.

An example is the formation of the total marginal profit y with a quadratic dependence function of the marginal profit of the i th products and the set price x_i . The dependence function has the following form (initial price values: $x_1 = 4$ CU, $x_2 = 2.7$ CU, $x_3 = 1.5$ CU):

$$y = (120 - (x_1 - 9)^2) + (140 - (x_2 - 10)^2) + (150 - (x_3 - 11)^2). \tag{6}$$

It is necessary to determine the changes Δx that ensure the total marginal profit value of 400 CU.

The results of applying the gradient method, as well as the standard function of the mathematical package, are presented in Table 1. We can note the difference in the values of the objective function by more than two times.

Table 1

Solution of the problem of marginal profit generation

Method	Δx_1	Δx_2	Δx_3	$f(\Delta x)$
Gradient	6.218	9.078	11.814	260.647
Using the MathCad function	3.782	5.522	7.186	96.433

Thus, the identified shortcomings indicate the feasibility of conducting a study on the development of algorithms different from the known ones in simpler computer implementation and faster problem solution. This paper discusses the development of iterative algorithms for solving the optimization problem based on existing algorithms using the inverse calculation apparatus. This will simplify the implementation of the methods, increase the solution accuracy and expand the range of problems to be solved.

3. The aim and objectives of the study

The aim of this work is to investigate the possibility of using iterative algorithms to solve inverse problems while minimizing the sum of squares of argument changes, as well as optimization problems of a more general kind. This will allow determining the values of arguments with less computational resources and higher accuracy compared to existing methods based on inverse calculations.

To achieve the aim, the following objectives were set:

- to develop iterative algorithms for solving inverse problems while minimizing the sum of squares of argument changes;
- to solve inverse problems using iterative algorithms and compare the results with solutions to problems in the MathCad package;
- to modify the algorithm to solve nonlinear programming problems of a more general kind;
- to solve optimization problems using iterative algorithms and compare the results with solutions of problems in the MathCad package.

4. Development of iterative algorithms for solving inverse problems

The initial data of the algorithms: the initial values of the arguments x , the given value of the resulting indicator $y+\Delta y$, α is a small positive number that provides movement towards the given value of the constraint $y+\Delta y$.

An iterative search based on the gradient method can be represented as follows:

Step 1. Using the initial data, calculate the value of the constraint function $h(x)$ and compare with the given value $y+\Delta y$:

- if $h(x) < y+\Delta y$, the arguments must be changed towards increasing the value of the constraint function (gradient vector direction): $t=1$;
- if $h(x) > y+\Delta y$, the arguments must be changed towards decreasing the value of the constraint function (antigradient vector direction): $t=-1$.

Step 2. Determine the absolute difference between the value of the constraint function and the given value $y+\Delta y$:

$$d_0 = |h(x) - y - \Delta y|.$$

Step 3. Determine new argument values by moving towards the gradient/antigradient:

$$x_i^* = x_i + t \cdot \alpha \cdot \frac{\partial h(x_i)}{\partial x_i}, \tag{5}$$

where $i=1...n$, n is the number of arguments.

Step 4. Calculate the value of the constraint function $h(x_i^*)$ and the deviation d_1 from the given value $y+\Delta y$.

Check: if $d_1 > d_0$, the algorithm ends. Otherwise, $d_0=d_1$, $x=x^*$, go to step 3.

The solution to the problem is x .

The algorithm based on the formation of the line of the given level includes the following steps (k is the number of the expressed variable, ϵ is the given accuracy, s is the implementation number):

Step 1. Set the initial values of the variables: $s=0$, $\hat{x} = x$.

From the constraint function $h(x)$ express the k -th variable:

$$x_k = g(x_l), \quad l \neq k.$$

Using the initial data, calculate the constraint value $h(x)$ and compare with the given value $y+\Delta y$:

- if $h(x) < y+\Delta y$, the arguments must be changed towards increasing the value of the constraint function (gradient vector direction): $t=1$;
- if $h(x) > y+\Delta y$, the arguments must be changed towards decreasing the value of the constraint function (antigradient vector direction): $t=-1$.

Step 2. Determine the absolute difference between the value of the constraint function and the given value $y+\Delta y$:

$$d_0 = |h(x) - y - \Delta y|.$$

Step 3. Determine the value of the partial derivatives of the function g :

$$r_i = \frac{-\partial g(x_i)}{\partial x_i}.$$

Step 4. Determine new argument values:

$$x_k^* = x_k + t \cdot \alpha, \quad x_i^* = x_i + t \cdot \alpha \cdot r_i, \quad i \neq k. \tag{6}$$

Step 5. Calculate the value of the constraint function $h(x_i^*)$ and the deviation d_1 from the given value $y+\Delta y$.

Check: if $d_1 > d_0$, go to step 6. Otherwise, $d_0=d_1$, $x=x^*$, go to step 4.

Step 6. Calculation of the objective function value: $s=s+1$, $f_s=f(x)$.

If $s > 1$, the algorithm is checked for completion; if $|f_s - f_{s-1}| < \epsilon$, the algorithm ends.

Step 7. Calculation of new values of partial derivatives:

$$r_i = \frac{-\partial g(x_i^*)}{\partial x_i}, \quad x = \hat{x},$$

go to step 4. The solution to the problem is x .

5. Results of solving inverse problems using iterative algorithms

Consider the use of iterative algorithms to solve inverse problems while minimizing the sum of squares of argument changes. The dependence of production volume on production factors (labor and capital costs) is described by the Cobb-Douglas function [21]:

$$y = A \cdot K^\alpha L^\beta,$$

where y is production volume; K is capital costs; L is labor costs; A, α, β are parameters.

The initial values of K, L are equal to 2 and 1.15, the parameters A, α, β are equal to 7, 0.5, 0.3, respectively. It is necessary to identify changes in these arguments in order to achieve a production volume of 17.

For an algorithm based on constructing the line of the given contour level, the first iterative formula is

$$\left(K = \left(\frac{17}{7L^{0.3}} \right)^2, \quad \frac{\partial K}{\partial L} = \frac{-3.539}{L^{1.6}} \right):$$

$$K = K + \alpha, \quad L = L + \alpha \frac{3.539}{L^{1.6}}.$$

For the gradient algorithm, the first iterative formula is $\left(\frac{\partial y}{\partial K} = \frac{3.5L^{0.3}}{K^{0.5}}, \frac{\partial y}{\partial L} = \frac{2.1K^{0.5}}{L^{0.7}}\right)$:

$$K = K + \alpha \frac{3.5L^{0.3}}{K^{0.5}},$$

$$L = L + \alpha \frac{2.1K^{0.5}}{L^{0.7}}.$$

Tables 2 and 3 show changes in the arguments in the process of solving the problem for $\alpha=0.01$ (the algorithm is implemented in Excel using VBA).

Table 2

Results of iterations using the gradient method

Iteration number	<i>K</i>	<i>L</i>	<i>d</i>	<i>f(x)</i>
1	2.026	1.177	6.537	0.001
2	2.052	1.204	6.399	0.006
3	2.077	1.231	6.263	0.012
...
57	3.449	2.468	0.046	3.835

Table 3

Results of implementations using the method based on forming the line of the given level ($\epsilon=0.02$)

Implementation number, <i>s</i>	<i>R</i>	<i>K</i>	<i>L</i>	<i>d</i>	<i>f(x)</i>
1	2.83	2.810	3.442	0.002	5.910
2	0.489	3.840	2.051	0.016	4.198
3	1.121	3.310	2.619	0.0003	3.873
4	0.758	3.550	2.326	0.011	3.784
5	0.917	3.430	2.461	0.014	3.765

According to the results obtained, it can be concluded that using the method based on forming the line of the given level, a smaller difference from the given constraint value and a smaller value of the objective function were achieved. However, the number of iterative calculations was higher and amounted to 699. Greater compliance with the specified constraint value can be achieved by decreasing the parameter α . So, Table 4 presents the results of solving the problem using two algorithms for $\alpha=10^{-8}$ ($\epsilon=0.001$). The last column shows the value *u* – the difference between the value of the objective function *f(x)*, obtained using this method and the value of the objective function using the standard MathCad function. Also Table 4 shows the results of applying classical methods of problem solution (penalty and Lagrange multiplier methods), the gradient method, and the method based on forming the line of the given level (description is given in Section 2). In the penalty method, the step of changing the penalty parameter is 10, and the accuracy is 10^{-8} . The greatest value of the difference *u* was obtained using the gradient method, and the difference *d* – the penalty method. Considering the parameters *d* and *u* as minimized values, it can be noted that the Pareto effective results will be those obtained using the Lagrange multiplier method, penalty method, and the standard MathCad function. In this case, the best result among the algorithms based on inverse calculations was obtained using the iterative algorithm based on forming the line of the given level.

Table 4

Solution of the problem of production volume formation ($\alpha=10^{-8}$)

Method	<i>K</i>	<i>L</i>	<i>d</i>	<i>f(x)</i>	<i>u</i>
Iterative gradient	3.441	2.455	$4.1 \cdot 10^{-7}$	3.779	$3.4 \cdot 10^{-3}$
Iterative based on forming the line of the given level	3.463	2.429	$5.6 \cdot 10^{-8}$	3.776	$2.8 \cdot 10^{-4}$
Gradient	3.412	2.562	$5 \cdot 10^{-7}$	3.827	0.051
Based on forming the line of the given level	3.463	2.429	$2.4 \cdot 10^{-7}$	3.776	$2.8 \cdot 10^{-4}$
Lagrange multiplier	3.472	2.418	0	3.776	$2.8 \cdot 10^{-6}$
Penalty	3.472	2.418	$3.9 \cdot 10^{-4}$	3.775	$-4.6 \cdot 10^{-4}$
Using the MathCad function	3.472	2.418	$2.3 \cdot 10^{-6}$	3.776	–

The gradient method was also used to solve the problem (6). The obtained values of argument increments: $\Delta x_1=3.782$, $\Delta x_2=5.522$, $\Delta x_3=7.186$. The value of the objective function is 96.433, the values of *d* and *u* are equal to $1.3 \cdot 10^{-6}$ and $-4.8 \cdot 10^{-4}$. Thus, the iterative algorithm made it possible to obtain a solution with a significantly lower value of the objective function (Table 1).

As an example of a problem for which the method based on constructing the line of the given level cannot be applied, the formation of storage costs can be considered (according to the classical inventory management model [22]). The cost-volume function of the first, second and third kind of products is presented as follows:

$$y = \frac{w_1 q_1}{x_1} + \frac{s_1}{2} x_1 + \frac{w_2 q_2}{x_2} + \frac{s_2}{2} x_2 + \frac{w_3 q_3}{x_3} + \frac{s_3}{2} x_3, \tag{7}$$

where *x* is the order size; *s* is the cost of storing a unit of products per unit of time; *w* is the cost per order; *q* is the intensity of demand.

The values of the variables are presented in Table 5. It is necessary to determine the order size of each type of products so that the total cost is 10 CU. The results of solving the problem are presented in Table 6.

Table 5

Input data of the cost formation problem

Indicator	Product number		
	1	2	3
Cost of storing a unit of products per unit of time, <i>s</i>	0.3	0.1	0.1
Cost per order, <i>w</i>	10	5	5
Demand intensity, <i>q</i>	2	4	5
Initial order size, <i>x</i>	7	5	4

Table 6
Solution of the cost formation problem for $\alpha = 10^{-8}$

Method	x_1	x_2	x_3	d	$f(x)$
Iterative gradient	8.347	7.986	8.233	$1.2 \cdot 10^{-9}$	28.649
Gradient	7.854	7.48	9.001	$9.28 \cdot 10^{-8}$	31.888
Using the standard MathCad function	8.525	8.102	8.069	$3.75 \cdot 10^{-6}$	28.508

The results obtained also suggest that the use of the iterative algorithm made it possible to obtain a solution with a lower value of the objective function.

6. Modification of iterative algorithms for solving nonlinear programming problems

The inverse calculation approach can be used to solve a wider class of optimization problems, in particular, nonlinear programming problems with one constraint in the form of equality [19]. The partial derivatives of the objective function must be one-dimensional functions. In this case, the gradient method can be used. For the iterative algorithm, it is necessary to perform the following modification:

1. To carry out unconstrained optimization of the objective function $f(x)$, subsequent use of iterative algorithms adjusts the obtained values of the arguments x . That is, instead of the initial values of x used in the inverse problem, the values obtained from the unconstrained optimization of the objective function $f(x)$ are used.

2. In iterative calculation formulas, it is necessary to make an adjustment that reflects the effect of argument changes on the objective function (if the second partial derivatives are neither constant nor equal). This operation is performed by dividing the first-order partial derivatives of the constraint function by the second-order partial derivatives of the objective function:

$$x_i^* = x_i + t \cdot \alpha \cdot \frac{\frac{\partial h(x_i)}{\partial x_i}}{\frac{\partial^2 f(x_i)}{\partial x_i^2}}$$

6.1. Results of solving optimization problems using iterative algorithms

Consider solving two classical problems of operations research using iterative algorithms: optimization of the securities portfolio and cost formation for a given total order.

The problem of optimization of the securities portfolio in the absence of their mutual influence and minimizing the risk is as follows [23]:

$$\begin{aligned} f(x) &= \sigma_1 x_1^2 + \sigma_2 x_2^2 + \\ &+ \sigma_3 x_3^2 + \sigma_4 x_4^2 \rightarrow \min, \\ m_1 x_1 + m_2 x_2 + m_3 x_3 + \\ &+ m_4 x_4 = M, \\ x_1 + x_2 + x_3 + x_4 &= 1, \end{aligned} \tag{8}$$

where σ is the risk indicator; m is the profit indicator; M is the profit margin.

The values of risk and profit indicators: $\sigma_1=0.0165$, $\sigma_2=0.0032$, $\sigma_3=0.0008$, $\sigma_4=0.0002$, $m_1=0.291$, $m_2=0.121$, $m_3=0.481$, $m_4=0.381$. The given value of profit M is 0.37.

The problem (8) has two constraints. To use the iterative algorithm, it is necessary to convert them into a single constraint. There are two ways to do this:

1) replacement of variables: expression of a variable from one constraint and substitution of it into the second constraint and objective function (the main advantage is the reduction of the dimension of the problem being solved);

2) formation of the constraint as the sum of squares of the difference between the constraint function and its given value.

Using the second method provided a solution with a lower value of the objective function. The optimization problem in this case is:

$$\begin{aligned} f(x) &= \sigma_1 x_1^2 + \sigma_2 x_2^2 + \sigma_3 x_3^2 + \sigma_4 x_4^2 \rightarrow \min, \\ (m_1 x_1 + m_2 x_2 + m_3 x_3 + m_4 x_4 - M)^2 + \\ &+ (x_1 + x_2 + x_3 + x_4 - 1)^2 = 0. \end{aligned}$$

The first iterative formula for the first variable is (the initial values of the variables x are zero):

$$\begin{aligned} \frac{\partial^2 f(x_i)}{\partial x_i^2} &= 2\sigma_i, \\ \frac{\partial h(x_i)}{\partial x_i} &= 2x_1 + 2x_2 + 2x_3 + 2x_4 + \\ &+ 2m_1(m_1 x_1 + m_2 x_2 + m_3 x_3 + m_4 x_4 - 0.37) - 2, \\ x_1 &= 0 - \alpha \frac{-2.215}{0.033}. \end{aligned}$$

We also consider the problem of minimizing the function of purchase and storage costs (7) for the given volume of purchases, which should be equal to 28:

$$x_1 + x_2 + x_3 = 28.$$

The initial data are presented in Table 5, while the initial values of the arguments x being the values obtained by unconstrained optimization of the function (7): $x_1=11.547$, $x_2=19.999$, $x_3=22.358$.

Table 7 shows the results of solving two optimization problems.

Table 7
Results of solving optimization problems for $\alpha = 10^{-8}$

Optimization problem	Method	x_1	x_2	x_3	x_4	d	$f(x)$
securities portfolio	Iterative gradient	0.011	0.086	0.12	0.782	$6 \cdot 10^{-22}$	$1.5 \cdot 10^{-4}$
	Gradient	0.008	0.04	0.186	0.76	$3 \cdot 10^{-4}$	$1.4 \cdot 10^{-4}$
	Using the MathCad function	0.011	0.086	0.12	0.782	10^{-12}	$1.5 \cdot 10^{-4}$
storage costs	Iterative gradient	7.884	9.497	10.618	–	$4 \cdot 10^{-6}$	9.185
	Gradient	9.389	8.786	9.825	–	$4 \cdot 10^{-15}$	9.29
	Using the MathCad function	7.884	9.497	10.618	–	$6 \cdot 10^{-13}$	9.185

According to the results obtained, the iterative gradient method provided greater compliance with the solution obtained using the mathematical package. In the cost optimization problem, it provided a lower value of the objective function, and in the portfolio optimization problem, a smaller difference between the value of the constraint function and the given value.

7. Discussion of the results of the development of iterative algorithms

Iterative algorithms for solving inverse problems of economic analysis are proposed. The first algorithm is based on determining the shortest distance to the line of the given level, determined by the constraint value. The second algorithm (gradient) is based on moving along the gradient until the constraint best complies with the given value. The algorithms are developed on the basis of the approaches discussed in [9, 19], however, their application does not require solving a system of equations. In addition, the results of the calculations show that in the gradient approach, the use of the iterative algorithm can provide greater compliance with the solution obtained using the mathematical package (Tables 1, 4, 6, 7). The highest degree of compliance with the solution obtained using the mathematical package was achieved using the algorithm based on the expression of a variable (Table 4). However, the algorithm based on moving along the gradient is more universal, since the expression of a variable can be performed not for all problems. The paper considers the solution of problems with a nonlinear constraint function: formation of production volume and cost of delivery and storage of products while minimizing the sum of squares of argument changes.

Modification of iterative algorithms, reflecting the effect of argument changes on the objective function, made it possible to obtain the solution of optimization problems (optimization of the securities portfolio and storage costs), consistent with the results of using the mathematical package (Table 7).

When using iterative algorithms, the choice of the argument change parameter plays an essential role. With the large value, there may be a significant difference from the specified value of the constraint (Tables 2, 3); if the value is too small, a large number of iterations will be required for the solution.

Compared to classical methods for solving nonlinear programming problems (Lagrange multiplier method, penalty method), the advantage of the proposed algorithms is that there is no need for repeated optimization of the modified function (including the objective function and constraint). Also, using these methods, no additional variables that increase the problem dimension are determined. In addition, the proposed iterative algorithms are simpler in terms of computer implementation, since they include iterative formulas for changing arguments using partial derivatives of the constraint function and second partial derivatives of the objective function.

The restriction of the algorithms is related to the type of optimization problems they can be used for. So, if the optimization problem has several constraints, they must have an equal sign. Also, partial derivatives of the objective function must be one-dimensional functions. In addition, the use of the algorithm based on constructing the line of the given level, which provided the best results, is limited due to the impossibility of expressing variables in some problems. The directions of further research will be related to the study of the possibility of modifying algorithms to solve optimization problems with constraints in the form of inequalities and multidimensional partial derivatives of the objective function.

8. Conclusions

1. Iterative algorithms for solving the inverse problem, presented as a quadratic programming problem with a single constraint are proposed. A feature of the proposed approach is the use of iterative formulas for changing arguments based on the inverse calculation apparatus. This apparatus allows a transition from the original argument values to those that satisfy the problem constraint. The approach used simplifies the implementation of algorithms as there is no need to implement methods for solving systems of equations. Compared to multiple optimizations of the modified function, the use of the proposed approach reduces the time of problem solution.

2. Using the developed algorithms, solution of two inverse problems with a nonlinear dependence of the resulting indicator on input variables is made. In the gradient approach, iterative change of arguments provided greater compliance with the solution of the problem using the standard function of the mathematical package compared to problem solution using the system of equations. Thus, iterative algorithms can provide a solution for a wider range of problems.

3. Modification of algorithms for solving nonlinear programming optimization problems of the presented type is performed. In this case, unconstrained optimization of the objective function is performed, and iterative formulas are adjusted to take into account the effect of arguments on the objective function.

4. The solution of two optimization problems using iterative algorithms is considered. The results of the numerical solution of problems are consistent with the results of using standard functions of mathematical packages and classical nonlinear optimization methods. With the argument change parameter α of 10^{-8} , the maximum absolute difference between the values of the objective function obtained using the iterative algorithm and mathematical package was $7 \cdot 10^{-7}$. When solving inverse problems, the maximum value of such a difference was obtained using the iterative gradient algorithm and amounted to 0.141. The presented algorithms can be used to create decision support systems for solving inverse and optimization problems.

References

1. Barmina, E. A., Kvyatkovskaya, I. Yu. (2010). Monitoring of quality of work of a commercial organization. Indicators structuring. Application of cognitive maps. *Vestnik Astrakhanskogo gosudarstvennogo tekhnicheskogo universiteta*, 2, 15–20.
2. Odintsov, B. E. (2004). *Obratnye vychisleniya v formirovanii ekonomicheskikh resheniy*. Moscow: Finansy i statistika, 256.
3. Zheng, G.-H., Zhang, Q.-G. (2018). Solving the backward problem for space-fractional diffusion equation by a fractional Tikhonov regularization method. *Mathematics and Computers in Simulation*, 148, 37–47. doi: <https://doi.org/10.1016/j.matcom.2017.12.005>

4. Park, Y., Reichel, L., Rodriguez, G., Yu, X. (2018). Parameter determination for Tikhonov regularization problems in general form. *Journal of Computational and Applied Mathematics*, 343, 12–25. doi: <https://doi.org/10.1016/j.cam.2018.04.049>
5. Bai, Z.-Z., Buccini, A., Hayami, K., Reichel, L., Yin, J.-F., Zheng, N. (2017). Modulus-based iterative methods for constrained Tikhonov regularization. *Journal of Computational and Applied Mathematics*, 319, 1–13. doi: <https://doi.org/10.1016/j.cam.2016.12.023>
6. Wang, H., Yang, W., Guan, N. (2019). Cauchy sparse NMF with manifold regularization: A robust method for hyperspectral unmixing. *Knowledge-Based Systems*, 184, 104898. doi: <https://doi.org/10.1016/j.knsys.2019.104898>
7. Scardapane, S., Comminiello, D., Hussain, A., Uncini, A. (2017). Group sparse regularization for deep neural networks. *Neurocomputing*, 241, 81–89. doi: <https://doi.org/10.1016/j.neucom.2017.02.029>
8. Xu, J., Schreier, F., Doicu, A., Trautmann, T. (2016). Assessment of Tikhonov-type regularization methods for solving atmospheric inverse problems. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 184, 274–286. doi: <https://doi.org/10.1016/j.jqsrt.2016.08.003>
9. Gribanova, E. (2020). Algorithm for solving the inverse problems of economic analysis in the presence of limitations. *EUREKA: Physics and Engineering*, 1, 70–78. doi: <https://doi.org/10.21303/2461-4262.2020.001102>
10. Qi, Y., Liu, D., Li, X., Lei, J., Xu, X., Miao, Q. (2020). An adaptive penalty-based boundary intersection method for many-objective optimization problem. *Information Sciences*, 509, 356–375. doi: <https://doi.org/10.1016/j.ins.2019.03.040>
11. El-Sobky, B., Abo-Elnaga, Y. (2018). A penalty method with trust-region mechanism for nonlinear bilevel optimization problem. *Journal of Computational and Applied Mathematics*, 340, 360–374. doi: <https://doi.org/10.1016/j.cam.2018.03.004>
12. Trunov, A. N. (2015). Modernization of means for analysis and solution of nonlinear programming problems. *Quantitative Methods in Economics*, 16 (2), 133–141.
13. Li, J., Yang, Z. (2018). A QP-free algorithm without a penalty function or a filter for nonlinear general-constrained optimization. *Applied Mathematics and Computation*, 316, 52–72. doi: <https://doi.org/10.1016/j.amc.2017.08.013>
14. Mitsel', A. A., Khvaschevskiy, A. N. (1999). Noviy algoritm resheniya zadachi kvadrachnogo programmirovaniya. *Avtometriya*, 3, 93–98.
15. Morovati, V., Pourkarimi, L. (2019). Extension of Zoutendijk method for solving constrained multiobjective optimization problems. *European Journal of Operational Research*, 273 (1), 44–57. doi: <https://doi.org/10.1016/j.ejor.2018.08.018>
16. Tsai, J.-T. (2015). Improved differential evolution algorithm for nonlinear programming and engineering design problems. *Neurocomputing*, 148, 628–640. doi: <https://doi.org/10.1016/j.neucom.2014.07.001>
17. Hosseini, A. (2016). A non-penalty recurrent neural network for solving a class of constrained optimization problems. *Neural Networks*, 73, 10–25. doi: <https://doi.org/10.1016/j.neunet.2015.09.013>
18. Darabi, A., Bagheri, M., Gharehpetian, G. B. (2020). Dual feasible direction-finding nonlinear programming combined with metaheuristic approaches for exact overcurrent relay coordination. *International Journal of Electrical Power & Energy Systems*, 114, 105420. doi: <https://doi.org/10.1016/j.ijepes.2019.105420>
19. Gribanova, E. (2019). Development of a price optimization algorithm using inverse calculations. *Eastern-European Journal of Enterprise Technologies*, 5 (4 (101)), 18–25. doi: <https://doi.org/10.15587/1729-4061.2019.180993>
20. Demin, D. (2017). Synthesis of optimal control of technological processes based on a multialternative parametric description of the final state. *Eastern-European Journal of Enterprise Technologies*, 3 (4 (87)), 51–63. doi: <https://doi.org/10.15587/1729-4061.2017.105294>
21. Zhang, Q., Dong, W., Wen, C., Li, T. (2020). Study on factors affecting corn yield based on the Cobb-Douglas production function. *Agricultural Water Management*, 228, 105869. doi: <https://doi.org/10.1016/j.agwat.2019.105869>
22. Sarmah, S. P., Acharya, D., Goyal, S. K. (2008). Coordination of a single-manufacturer/multi-buyer supply chain with credit option. *International Journal of Production Economics*, 111 (2), 676–685. doi: <https://doi.org/10.1016/j.ijpe.2007.04.003>
23. Kalayci, C. B., Ertenlice, O., Akbay, M. A. (2019). A comprehensive review of deterministic models and applications for mean-variance portfolio optimization. *Expert Systems with Applications*, 125, 345–368. doi: <https://doi.org/10.1016/j.eswa.2019.02.011>