Запропоновано алгоритм вирішення задачі оптимізації ціни за допомогою зворотних обчислень. Алгоритм включає два етапи: рішення задачі безумовної оптимізації і рішення зворотної задачі з допомогою зворотних обчислень при мінімізації зміни аргументів функції. При цьому рішення зворотного завдання може бути виконано багаторазово протягом заданого числа ітерацій для послідовного наближення до встановленого значення обмеження, а для визначення збільшень аргументів використовуються значення елементів вектора градієнта/антиградієнта функції обмеження. Для врахування впливу аргументів на зміну цільової функції використовуються її другі частні похідні. Розглянуто п'ять варіантів завдання оптимізації ціни, які представляють собою завдання нелінійного програмування з одним обмеженням. У завданнях враховується залежність попиту від ціни і передбачається, що вона має лінійний вигляд. Як цільову функцію розглянуто виручку підприємства, відхилення попиту від обсягу виробництва, відхилення шуканої ціни від її поточного значення. Показано, що одержувані при цьому рішення узгоджуються з результатом використання класичних методів (множників Лагранжа, штрафів), також виконано порівняння результатів з рішенням задач за допомогою математичного пакету MathCad. Перевагою методу є більш проста комп'ютерна реалізація, можливість отримати рішення за менше число ітерацій в порівнянні з відомими методами. Метод може бути також використаний для вирішення інших завдань представленого виду з наступними вимогами до цільової функції та обмежень:

1) частні похідні цільової функції першого порядку – лінійні одномірні функції;

2) обмеження має вид рівності;

3) обмеження має лінійний вигляд або обмеження має квадратичний вигляд, а частні похідні першого порядку функції обмеження – лінійні одновимірні функції.

Стаття може бути корисною для фахівців, що здійснюють прийняття рішень в області цінової політики організацій, а також розробку оптимізаційних моделей об'єктів економіки і систем підтримки прийняття рішень

Ключові слова: зворотні обчислення, оптимізація ціни, квадратичне програмування, градієнтний метод, зворотне завдання

UDC 519.866.2

DOI: 10.15587/1729-4061.2019.180993

DEVELOPMENT OF A PRICE OPTIMIZATION ALGORITHM USING INVERSE CALCULATIONS

E. Gribanova
PhD, Associate professor
Department of Automated Control System
Tomsk state university of control systems and radioelectronics
Lenina str., 40, Tomsk, Russia, 634050
E-mail: geb@asu.tusur.ru

Received date 02.09.2019
Accepted date 04.10.2019
Published date 31.10.2019

1. Introduction

The main aim of the organization's pricing policy is improvement of the financial results of its activities: maximizing revenue and profits, ensuring that sales volumes match planned values, etc. When pricing, it becomes necessary to analyze factors such as consumer demand and its dependence on price, interchangeability and complementarity of goods, the number of competitors, limited resources for the production and delivery of products. Pricing also acts as a tool of marketing policy, allowing to perform sales promotion, attracting and retaining customers.

A large number of approaches used to determine prices is explained by the variety of types of trade and production processes and factors affecting the formation of prices, the challenges facing enterprises, as well as the specifics of their activities.

One of the main factors determining price formation is customer demand. To predict its value depending on the price, models are built on the basis of statistical data. Moreover, both the volume of demand and the probability of purchase can be considered as the predicted value. Among the most famous forecasting models, regression models [1–7], as well as their combinations with other methods [8], can be noted. At the same time, forecasting models can take into account the interchangeability of products [9], competition in the market [10], and the limited shelf space of the store [2].

When optimizing prices, depending on the statement of the problem, it becomes necessary to solve the problem of linear, nonlinear, integer, linear-fractional programming. Moreover, the use of classical methods for solving problems can be difficult due to their complexity and complexity in computer implementation, as well as the high cost of computing resources, especially when considering tasks of large dimension. Therefore, the development of optimization algorithms that are more efficient and simple in computer implementation (for example, that do not require performing multiple iterations, determining additional variables that increase the dimension of the problem, and forming modified functions) is an urgent task.

2. Literature analysis and problem statement

The solutions to the problems of price optimization are devoted to [6, 9, 11–15]. They present optimization models taking into account the specifics of the object of study. So, in [11], the problem of maximizing revenue by considering the set of prices for competing goods is considered, provided that the sum of prices of the group of goods is equal to a given value. In article [12], income from the sale of "fast fashion" class clothes consists of revenue until the last week of sales and revenue in the last week of sales when the collection is liquidated at the lowest price. In [6], when maximizing profit, producer prices are taken into account. The authors also consider the solution to the problem of price optimization by modeling the user's choice of goods taking into account their interchangeability [9], setting individual prices in each sales channel (website, mobile application, social networks) [13]. In [14], cost accounting for the purchase and storage of goods is performed, in [15] two types of customers are investigated: loyal and disloyal, and the shelf space in the store is considered limited.

From the studies presented in [7–9, 12–15] it follows that the problem of price optimization is often presented as a nonlinear programming problem. Let's consider the optimization problem with one constraint in the form of equality. A linear dependence of demand on price is assumed, linear regression parameters for determining the forecast value of weekly demand are determined on the basis of available statistical data on the values of prices and demand for previous periods. The classic method for estimating regression parameters is the least squares method.

Let's define the following notation:

– p_j – the desired price of the product in the j-th period (j=1..n, n – the number of periods);

– p_i – the desired price for the product of the i-th type (i=1..m, m – the number of types of products);

– q_i – the current price of the i-th product;

– a_i and b_i – the linear regression parameters used to determine the demand v_i for an i-type product:

$$v_i = a_i + b_i \cdot p_i.$$

In this case, negative elasticity of demand is assumed, i. e. its decrease with rising prices, therefore, the following restrictions are imposed on the sign of the parameters: $a_i \geq 0$ and $b_i \leq 0$.

If a product of one type is considered, then the parameters are indicated without indices: a and b.

– c_j – the planned output of the product in the j-th period;

– P_1, P_2 – the value of the revenue to be received;

– r – the resource costs for the manufacture of a unit of product;

– R – the value of the available material stock of the enterprise;

– h_i – the volume of a unit of a product of the i-th type;

S – the delivery volume.

It is possible to determine the following options for the problems of price optimization p (revenue is used as a financial indicator of the company's activity):

Option 1: Minimizing the deviation of projected demand in the j-th period from the planned production volume with a restriction on the volume of the used resource: the volume of the used resource for the production of the product is equal to the available stock [16]:

$$f(p) = \sum_{j=1}^{n} \left(a + b \cdot p_j - c_j\right)^2 \to \min,$$

$$h(p) = \sum_{j=1}^{n} r v_j = \sum_{j=1}^{n} r\left(a + b \cdot p_j\right) \leq R, \quad p_j \geq 0. \tag{1}$$

Option 2: Maximizing the total revenue (minimization of the value obtained by multiplying the revenue by –1) obtained by summing the proceeds from the sale of the i-th type of product, if there is a limit on the delivery volume of all types of products:

$$f(p) = -\sum_{i=1}^{m} \left(a_i + b_i \cdot p_i\right) p_i \to \min,$$

$$h(p) = \sum_{i=1}^{m} h_i v_i = \sum_{i=1}^{m} h_i \left(a_i + b_i p_i\right) \leq S, \quad p_i \geq 0. \tag{2}$$

The solution to this problem while maximizing profits is considered in [19].

Option 3: Minimizing the deviation of the forecasted demand in the j-th period from the planned production volume of the product if there is a limit on the value of total revenue for n periods, the value of which should be equal to the established:

$$f(p) = \sum_{j=1}^{n} \left(a + b \cdot p_j - c_j\right)^2 \to \min,$$

$$h(p) = \sum_{j=1}^{n} p_j v_j = \sum_{j=1}^{n} p_j \left(a + b \cdot p_j\right) \geq P_1, \quad p_j \geq 0. \tag{3}$$

Option 4: Minimizing the deviation of the projected demand for the i-th type product from the planned production volume of a product of this type if there is a limit on the value of total revenue for all types of products, the value of which should be equal to the established:

$$f(p) = \sum_{i=1}^{m} \left(a_i + b_i \cdot p_i - c_i\right)^2 \to \min,$$

$$h(p) = \sum_{i=1}^{m} p_i v_i = \sum_{i=1}^{m} p_i \left(a_i + b_i \cdot p_i\right) \geq P_1, \quad p_i \geq 0. \tag{4}$$

Option 5: Minimizing the deviation of the desired price from the current (find the value closest to the desired) if there is a limit on the total revenue for all types of products, the value of which should be equal to the given value:

$$f(p) = \sum_{i=1}^{m} \left(p_i - q_i\right)^2 \to \min,$$

$$h(p) = \sum_{i=1}^{m} p_i v_i = \sum_{i=1}^{m} p_i \left(a_i + b_i \cdot p_i\right) \geq P_2, \quad p_i \geq 0. \tag{5}$$

The solution of the presented problems can be performed using classical nonlinear programming methods: fines and Lagrange multipliers, which are based on reducing the conditional optimization problem to the unconditional optimization problem.

In the penalty method, a function is formed that includes the objective function and the penalty function of the restriction and the penalty parameter. In the case of a restriction in the form of inequality, a logarithmic penalty, a penalty of the inverse function type, a penalty of the type of a cutoff square are used. Let's consider the use of a logarithmic penalty. The process of solving the problem involves iterative reduction of the penalty parameter W and unconditional optimization of the penalty function. The algorithm stops when the change in the values of the arguments and the function is less than the specified accuracy. So, for the third option, the penalty function V will have the form:

$$V(p,W)=\sum_{j=1}^{n}\left(a+b\cdot p_j-c_j\right)^2-$$
$$-W\cdot\ln\left(\sum_{j=1}^{n}p_j\left(a+b\cdot p_j\right)-P_1\right).$$

Let's take the following values of the initial data n=3, a=148.2, b=–1.15, c_1=10, c_2=5, c_3=11, P_1=3,400 rubles. Table 1 shows the results obtained using the penalty method (the initial values of the arguments are 300).

Table 1

Results obtained using the penalty method

Penalty parameter W	Function arguments			Target function $f(p)$
	p_1	p_2	p_3	
100	116.4	120.46	115.59	58.84
10	118.70	122.93	117.85	8.988
1	119.36	123.65	118.51	2.71
0.1	119.5	123.79	118.64	1.9
0.01	119.51	123.81	118.65	1.813
0.001	119.51	123.81	118.65	1.804

In the Lagrange multiplier method, a function L is formed that includes an unknown parameter – the Lagrange multiplier: the sum of the objective function and the constraint multiplied by the Lagrange multiplier λ are determined [16–18]. So, for a variant of problem (1) (r=30, R=600, the Lagrange function will have the form:

$$L(p,\lambda)=\sum_{j=1}^{n}\left(a+b\cdot p_j-c_j\right)^2+$$
$$+\lambda\left(\sum_{j=1}^{n}r\left(a+b\cdot p_j\right)-R\right).$$

To solve the problem, it is necessary to calculate the partial derivatives of the function with respect to the variables p, equate them to zero and solve the system of equations:

$$\begin{cases}2.645p_1-34.5\lambda-317.86=0;\\ 2.645p_2-34.5\lambda-329.36=0;\\ 2.645p_3-34.5\lambda-315.56=0;\\ \lambda\begin{pmatrix}(148.2-1.15\cdot p_1)\cdot 30+(148.2-1.15\cdot p_2)\cdot 30+\\ +(148.2-1.15\cdot p_3)\cdot 30-600\end{pmatrix}=0.\end{cases}$$

The considered methods are complex and time-consuming in computer implementation and require high computational resources when considering large-dimensional problems. Thus, the penalty method requires the multiple solution of the unconditional optimization problem with various values of the penalty parameter; the Lagrange multiplier method involves the formation of an expression to determine the Lagrange multiplier. In addition, with unconditional optimization, a modified function is considered, including the objective function and the restriction, for which the use of local search methods can be difficult.

An option to overcome such difficulties is use of stochastic methods based on direct random search [20], application of evolutionary mechanisms [21], etc. In this case, the formation of an optimized modified function does not occur. Such methods make it possible to obtain a certain solution in a user-specified time, which makes it possible to use them in problems of large dimension, when the use of classical methods can lead to an unacceptably large solution time. However, the resulting solution will be suboptimal and change in different launches of software implementation. In addition, the implementation of the algorithms themselves can be difficult due to the many rules for adjusting the solutions obtained at each iteration.

The option of obtaining a simpler numerical solution to the quadratic programming problem is considered in [22] and is based on the formation of the system in accordance with the Kuhn-Tucker conditions. Thus, the quadratic problem is represented as a linear programming problem. However, the approach is suitable only for the case when the restriction has a linear form, and its application is associated with the formation of simplex tables and the use of the simplex method.

The identified shortcomings of existing methods indicate the feasibility of conducting a study on the development of an effective algorithm for solving the presented optimization problems, devoid of the listed disadvantages associated with the formation of a modified function, requirements for the type of restriction. For its development, the use of the inverse computing apparatus is considered.

By solving problems with the help of inverse calculations [23] let's mean finding the increments of the arguments of the function based on the following information: the initial values of the arguments and the function, the new value of the function, the coefficients of the relative importance of the arguments, the direction of change of the arguments. If it is necessary to determine the new value of the function so that the sum of the squares of the increments of the arguments is minimal, then in this case there is no need to use expert information. The solution to this problem is considered in [24], where expressions for the additive, multiplicative, and multiple dependencies between arguments are determined, including those obtained using geometric constructions.

3. The aim and objectives of the study

The aim of the study is development of a method for solving optimization problems of pricing, which differs from the existing ones using a two-stage approach, including unconditional optimization of the objective function and adjustment of the obtained values of the arguments using inverse calculations.

To achieve the aim, the following objectives are set:

– to build mathematical models to solve the problem of price optimization;

– to develop a method for solving the problem of price optimization based on inverse calculations;

– to perform a comparison of the solutions obtained as a result of the implementation of the algorithm with the solutions of problems in the MathCad mathematical package.

4. An algorithm for solving the problem of price optimization based on inverse calculations

The optimization problems considered (1)–(5) relate to quadratic programming problems and differ in the form of the objective function and constraint.

By type of objective function, problems can be divided into two groups:

1 Changing the function argument by β value (relative to the minimum point) will lead to the same change in the objective function as when changing another argument by β value. This means the fulfillment of the following ratio:

$$f\left(p_1^*+\beta, p_2^*, \ldots, p_n^*\right)=f\left(p_1^*, p_2^*+\beta, \ldots, p_n^*\right)= \\ =f\left(p_1^*, p_2^*, \ldots, p_n^*+\beta\right), \qquad (6)$$

where $\left(p_1^*, p_2^*, \ldots, p_n^*\right)$ – the minimum point; β – a certain number.

In this case, the second partial derivatives of the function will be constant and equal to each other.

This kind of function is found in options 1, 3, 5.

2. The condition for the objective function (6) is not satisfied. Partial derivatives of functions are linear one-dimensional functions (the second partial derivatives are constant).

This kind of function is found in options 2, 4.

Fig. 1, *a*, *b* show level lines for the first and second cases, respectively.

By type of constraint, problems are also divided into two groups:

1. The restriction has the form of linear equality.

This type of restriction is found in options 1, 2.

2. The restriction is non-linear. Partial derivatives of the restriction function are one-dimensional linear functions (the second partial derivatives are constant).

This type of restriction is found in options 3, 4, 5.

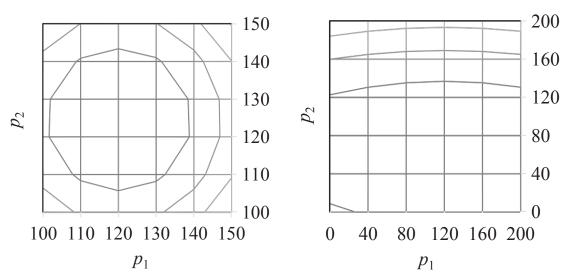

Fig. 1. Outline graph:
a – function $f(p)=\left(a+b\cdot p_1-10\right)^2+\left(a+b\cdot p_2-5\right)^2$;
b – function $f(p)=\left(a_1+b_1\cdot p_1-10\right)^2+\left(a_2+b_2\cdot p_2-5\right)^2$

Let's consider the application of this approach to solve the presented problems. The solution of the problem will include two main stages: the solution of the unconditional optimization problem and the subsequent correction of the obtained solution p^* by Δp value taking into account the limitations. The Δp value is determined by the difference $\Delta p=p-p^*$, where p is the value of the argument, which is the solution to the quadratic programming problem. In this case, it is necessary to take into account the influence of individual arguments on the change in the objective function.

Let's consider the option when the objective function satisfies condition (6). In the case of a nonlinear constraint, the expression of the argument can't be performed, and, consequently, the use of the method described in [24] can't be performed; therefore, the use of the gradient method for solving the problem of adjusting the values of arguments p^* is proposed.

The gradient is a partial derivative vector that shows the direction of the greatest increase in the function. Accordingly, the anti-gradient shows the direction of the greatest decrease in the function. The essence of the proposed method is that the ratio of the values of the increments of the arguments corresponds to the ratio of the elements of the gradient vector, i. e., the change of the arguments occurs in the direction of the greatest increase/decrease of the restriction function. Since when moving in the direction of the gradient/anti-gradient, the largest increase/decrease of the function is observed, this indicates that it is possible to achieve its predetermined value with smaller changes in the arguments. In turn, a smaller change in the arguments will lead to a smaller deviation of the value of the objective function from the value obtained by solving the problem of unconditional optimization. So, for example, for function (6) in the case of two arguments and their positive increments, the following relation holds:

$$f\left(p_1^*+\Delta p_1^{(1)}, p_2^*+\Delta p_2^{(1)}\right)<f\left(p_1^*+\Delta p_1^{(2)}, p_2^*+\Delta p_2^{(2)}\right)$$

at

$$\Delta p_1^{(1)}+\Delta p_2^{(1)}<\Delta p_1^{(2)}+\Delta p_2^{(2)}.$$

Fig. 2 shows an example of solving a problem for a function with two arguments. The starting point A is obtained by solving the unconstrained optimization problem, B is the point obtained by moving the function $h(x)$ in the direction of the anti-gradient to the intersection with the curve of a given level $\left(x_1=\sqrt{\frac{3-0.5x_2^2}{2}}\right)$. Elements of the anti-gradient vector of the function $h(x)$ at point A are equal to $(-4; -2)^{\mathrm{T}}$. The use of the anti-gradient vector is due to the fact that the value of the constraint at point A is 4, which exceeds the specified value – 3, therefore, the value of the function must be reduced.

However, the direction of the gradient can change when moving to a given value of the constraint function, therefore, the use of the values of the vector elements calculated at the starting point can lead to a solution that differs significantly from the optimal one. In this case, the movement to the set limit value can be performed step by step.

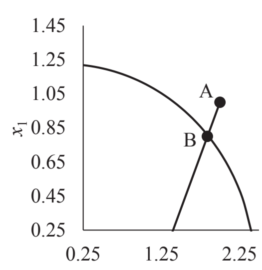

Fig. 2. Gradient method for solving the problem
$f(x)=\left(x_1-1\right)^2+\left(x_2-2\right)^2$, $h(x)=2x_1^2+0.5x_2^2=3$

Thus, the solution of the problem when using the gradient vector includes the following steps:

Step 1. Solving the problem of unconditional optimization of the objective function $f(p)$. The resulting solution includes a set of prices p^*. Substitution of the obtained p^*

values into the constraint and verification of the condition: if the inequality is fulfilled, then the algorithm completes, otherwise the transition to the next step.

Step 2. Substitution of the obtained p^* values into the constraint $U^*=h(p^*)$. Checking the direction of changing the arguments: if $U^*>U$, then the value of the constraint function must be reduced (elements of the anti-gradient vector are used) and $t=-1$, otherwise, increase (elements of the gradient vector are used) and $t=1$.

Step 3. Determination of the step of changing the constraint υ based on a given number of iterations ρ:

$$\upsilon = \text{integer}\left[\frac{(U-U)^*}{\rho}\right],$$

current iteration number $\alpha=1$.

Step 4. Change of the value of the resulting indicator by the value of the specified step:

$$U^* = U^* + \upsilon.$$

Step 5. Determination of the necessary increments of the arguments Δp_i to achieve the specified value of the constraint U^* by solving the system of equations:

$$\begin{cases} \dfrac{\Delta p_\eta}{\Delta p_i} = \dfrac{t\cdot\dfrac{\partial h(p^*)}{\partial p_\eta}}{t\cdot\dfrac{\partial h(p^*)}{\partial p_i}}, i=1..n, i\neq\eta; \\ h(p^*+\Delta p) = U^*. \end{cases}$$

As a result of solving the system, let's obtain the values of the increments of the arguments Δp_i. In the case of a linear constraint, the resulting relation is equivalent to the system presented in [24], where the increments of the arguments are determined based on the criterion of minimizing their sum of squares:

$$\begin{cases} \dfrac{\Delta p_\eta}{\Delta p_i} = \dfrac{k_\eta}{k_i}, i=1..l, i\neq\eta; \\ h(p^*+\Delta p) = U. \end{cases}$$

where k_i – the coefficients of p_i in the linear constraint equation.

Step 6. Changing the values of the function arguments:

$$p_i^* = p_i^* + \Delta p_i.$$

Step 7. Checking the completion of the algorithm: if $\alpha=\rho$, then the operation of the algorithm ends, otherwise $\alpha=\alpha+1$, go to step 4.

Finally, let's consider the case when condition (6) for the objective function is not satisfied. This means that changing arguments has a different effect on changing the objective function. The case when the partial derivatives of the objective function are linear one-dimensional functions is considered. To take into account the influence of the arguments on the change in the objective function relative to the minimum point, let's use the values of the second partial derivatives:

$$\begin{cases} \dfrac{\Delta p_\eta}{\Delta p_i}\dfrac{\dfrac{\partial^2 f(p)}{\partial p_\eta^2}}{\dfrac{\partial^2 f(p)}{\partial p_i^2}} = \dfrac{t\cdot\dfrac{\partial h(p^*)}{\partial p_\eta}}{t\cdot\dfrac{\partial h(p^*)}{dp_i}}, i=1..l, i\neq\eta; \\ h(p^*+\Delta p) = U. \end{cases}$$

The solution to the problem can also be carried out iteratively in accordance with the above algorithm.

For the presented problems (1)–(5), the obtained systems for determining the increments of the arguments will accordingly look as follows.

The determination of the increments of the arguments for solving problem (1) is performed using the system of equations:

$$\begin{cases} \dfrac{\Delta p_\eta}{\Delta p_j} = \dfrac{b\cdot r}{b\cdot r} = 1, j=1..n, \eta\neq j; \\ h(p^*+\Delta p) = \sum_{j=1}^{n} r\left(a + b\cdot\left(p_j^*+\Delta p_j\right)\right) = R. \end{cases}$$

Having completed the solution of the system, let's obtain the following expression for calculating the growth of function arguments:

$$\Delta p_j = \frac{\dfrac{\dfrac{R}{r} - n\cdot a}{b} - \sum_{i=1}^{n} p_i^*}{n}, j=1...n.$$

The system of equations for solving problem (2) has the form:

$$\begin{cases} \dfrac{\Delta p_\eta}{\Delta p_i} = \dfrac{h_\eta}{h_i}; i=1..m, i\neq\eta \\ h(p^*+\Delta p) = \sum_{i=1}^{m} h_i\left(a_i + b_i\left(p_i^*+\Delta p_i\right)\right) = S. \end{cases}$$

Solving the system, let's obtain expressions for calculating the arguments:

$$\Delta p_\eta = \frac{S - \sum_{j=1}^{n}\left(h_j\left(a_j + b_j p_j^*\right)\right)}{\sum_{j=1}^{n}\left(\dfrac{h_j^2\cdot b_j}{h_\eta}\right)}, \quad \Delta p_i = \frac{h_i}{h_\eta}\cdot\Delta p_\eta,\ i=1..m,\ i\neq\eta.$$

The calculation of the increments of the arguments for solving problem (3) is as follows:

$$\begin{cases} \dfrac{\Delta p_\eta}{\Delta p_j} = \dfrac{\left(a+2bp_\eta^*\right)}{\left(a+2bp_j^*\right)}, j=1..n, j\neq\eta; \\ h(p^*+\Delta p) = \sum_{j=1}^{n}\left(p_j+\Delta p_j\right)\left(a + b\cdot\left(p_j+\Delta p_j\right)\right) = P_1. \end{cases}$$

In this and subsequent versions, the restriction is non-linear, therefore, the determination of the increment of the basic argument can be performed using standard methods for solving the quadratic equation. So, for the current version, the equation will look like:

$$\Delta p_\eta^2\left(b\sum_{i=1}^{n}\left(\frac{a+2bp_i^*}{a+2bp_\eta^*}\right)^2\right)+$$

$$+\Delta p_\eta\left(a\sum_{i=1}^{n}\left(\frac{a+2bp_i^*}{a+2bp_\eta^*}\right)+2b\sum_{i=1}^{n}\left(p_2^*\frac{a+2bp_i^*}{a+2bp_\eta^*}\right)\right)+$$

$$+a\sum_{i=1}^{n}p_i^*+b\sum_{i=1}^{n}\left(p_i^*\right)^2-P_1=0.$$

The system of equations for solving problem (4) has the form:

$$\begin{cases}\dfrac{\Delta p_\eta \dfrac{b_\eta^2}{b_i^2}}{\Delta p_i}=\dfrac{\left(a_\eta+2b_\eta p_\eta^*\right)}{\left(a_i+2b_i p_i^*\right)}, i=1..m, i\neq\eta;\\ h\left(p^*+\Delta p\right)=\sum_{i=1}^{m}\left(p_i^*+\Delta p_i\right)\left(a_i+b_i\cdot\left(p_i^*+\Delta p_i\right)\right)=P_1.\end{cases}$$

Finally, the determination of the increments of the function arguments for solving problem (5) is carried out by solving the system of equations:

$$\begin{cases}\dfrac{\Delta p_\eta}{\Delta p_i}=\dfrac{\left(a_\eta+2b_\eta q_\eta\right)}{\left(a_i+2b_i q_i\right)}, i=1..m, i\neq\eta;\\ h\left(p^*+\Delta p\right)=\sum_{i=1}^{m}\left(p_i^*+\Delta p_i\right)\left(a_i+b_i\cdot\left(p_i^*+\Delta p_i\right)\right)=P_2.\end{cases}$$

It can be concluded that in the case of a linear constraint, analytical formulas for calculating increments can be obtained. In the nonlinear form of the constraint (problems (3)–(5)), to determine the increments of the arguments, it becomes necessary to solve the quadratic equation. In this case, classical methods for finding roots (Newton's methods, dichotomies, the use of discriminant, etc.) can be used.

5. Results of solving optimization problems

To solve the inverse problem, the data presented in Table 2 are used.

Table 2

Problem input data

Indicator	Product number i/Period number j		
	1	2	3
Linear regression parameter a	148.2	152.1	130.5
Linear regression parameter b	–1.15	–1.21	–1.1
Resource costs per unit of output r, g	30	–	–
Planned volume of production, c, kg	10	5	11
Volume of a product unit, h, m^3	0.2	0.4	0.5
The current price of products, q, rub.	80	75	83

Limit values: R=600 g, S=60 m^3, P_1=3,400 rub., P_2= =12,700 rub.

Table 3 shows the solution to optimization problems (1)–(5) (the number of iterations ρ is 1). The last column presents the difference of the obtained solution with the solution of the problem using a mathematical package:

$$\varepsilon=f(x)-f^*(x),$$

where $f(x)$ – the value of the objective function obtained by solving the problem using inverse calculations; $f^*(x)$ – the value of the objective function obtained by solving the problem using the built-in MathCad "Minimize" function.

Table 3

Solution to the optimization problems

Option	The value of the objective function, $f(x)$	Product price			Difference ε
		p_1	p_2	p_3	
1	12	121.91	126.26	121.04	$-7\cdot10^{-12}$
2	–12800	71.38	76.74	76.68	$-5\cdot10^{-6}$
3	1,803	119.51	123.81	118.65	$-1\cdot10^{-5}$
4	4,434	119.1	120.49	107.64	$6\cdot10^{-6}$
5	30,245	77.18	72.68	78.89	$2\cdot10^{-3}$

From the Table 3 it is possible to see that the solution to the third problem is also consistent with the solutions obtained using the penalty method (Table 1) and the Lagrange multipliers. The solution for the fifth option using the inverse calculation method has the least accuracy. Fig. 3 shows a graph of the change in the objective function of this problem depending on the number of iterations ρ. It is possible to see that with an increase in the number of iterations, the value of the objective function decreases.

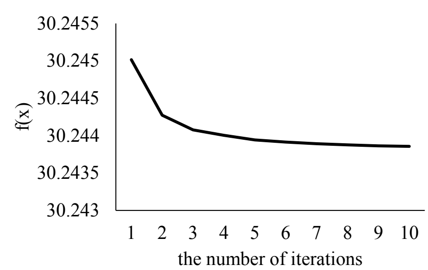

Fig. 3. The dependence of the objective function on the number of iterations

As an example, consider option (3). Substituting the initial numerical values, let's obtain the following problem:

$$f(p)=\left(148.2-1.15\cdot p_1-10\right)^2+\left(148.2-1.15\cdot p_2-5\right)^2+$$
$$+\left(148.2-1.15\cdot p_3-11\right)^2\rightarrow\min,$$

$$p_1\left(148.2-1.15\cdot p_1\right)+p_2\left(148.2-1.15\cdot p_2\right)+$$
$$+p_3\left(148.2-1.15\cdot p_3\right)=3{,}400.$$

The solution to the problem of unconditional optimization:

$$p_1^*=120.17\text{ rubles},\ p_2^*=124.52\text{ rubles},\ p_3^*=119.3\text{ rubles}.$$

Substituting the values in the constraint, let's obain:

$$h(p)=(148.2-1.15\cdot 120.17)+(148.2-1.15\cdot 124.52)+$$
$$+(148.2-1.15\cdot 119.3)=3{,}137.$$

Since the values need to be increased, gradient values are used. Thus, the system of equations has the form (with a step ρ equal to 1):

$$\begin{cases} \dfrac{\Delta p_1}{\Delta p_2}=\dfrac{-128.2}{-138.2}; \\ \dfrac{\Delta p_1}{\Delta p_3}=\dfrac{-128.2}{-126.2}; \\ (120.174+\Delta p_1)(148.2-1.15\cdot(120.174+\Delta p_1))+ \\ +(124.52+\Delta p_2)\times(148.2-1.15\cdot(124.52+\Delta p_2))+ \\ +(119.3+\Delta p_3)(148.2-1.15\cdot(119.3+\Delta p_3))=3{,}400. \end{cases}$$

The solution to the system of equations will be the values of the increments of the arguments: $\Delta p_1=-0.66$, $\Delta p_2=-0.71$, $\Delta p_3=-0.65$. Then the following values of the arguments will be the solution of the problem: $p_i=p_i^*+\Delta p_i$: $p_1=119.51$, $p_2=123.81$, $p_3=118.65$.

6. Discussion of the results of the development of an algorithm for solving the problem of price optimization

An algorithm based on the inverse calculations of the solution to the problem of price optimization is proposed, five options of quadratic programming problems with one constraint are considered, which differ in the form of the objective function and constraints. The results of the application of the algorithm are consistent with the solutions obtained using the mathematical package (Table 3). Moreover, the greatest degree of correspondence is achieved with linear restriction and equality of all second-order partial derivatives of the objective function (in this case, the inverse problem is solved in one iteration). With nonlinear constraint, the smallest degree of correspondence is achieved (option 5), in this case an iterative multi-step procedure was applied, which ensured that a lower value of the objective function is obtained compared to a single-step one.

The following advantages of the proposed algorithm can be noted. Unlike the penalty method, the presented algorithm does not require multiple solutions to the unconditional optimization problem (the unconditional optimization task objective problem is solved once, after which the inverse problem is solved). In addition, the penalty method uses a modified function that combines the objective function and restriction, and finding its minimum is a more difficult task compared to determining the minimum of the objective function. The advantage of the proposed algorithm compared to the method of Lagrange multipliers is that there is no need to compose a relation to determine the Lagrange multiplier.

The disadvantage of the algorithm is its limited application, in particular, the number of constraints is equal to unity, and the objective function and constraint must satisfy the following requirements. Partial derivatives of the first order of the objective function are linear one-dimensional functions, the restriction has a linear form or partial derivatives of the first order of the restriction function are linear one-dimensional functions.

The directions of further research are related to the modification of the developed algorithm for solving optimization problems in the presence of several limitations and its application in other subject areas (for example, in inventory management).

7. Conclusions

1. An algorithm is proposed for solving the problem of price optimization, which is a quadratic programming problem with one restriction. Application of the algorithm allows one to obtain results that are consistent with the results of using classical nonlinear optimization methods. Confirmation of this is given in the results of the numerical solution of five problems of price optimization. A feature of the proposed approach is the absence of the need to form a modified function and repeatedly solve the problem of unconditional optimization, which simplifies the procedure for its computer implementation and accelerates the time of solving the problem. This is possible due to the use of the inverse computation apparatus in the proposed algorithm, which allows one to go from the values of the arguments obtained as a result of unconditional optimization of the objective function to the values of the arguments satisfying the constraint of the problem.

2. The presented algorithm and optimization models can be implemented in decision support systems, providing the organization's specialists with the opportunity to form a set of prices that maximizes revenue or ensures its predetermined value. An option may also be considered to ensure maximum compliance of demand with the planned value of the volume of production. The above algorithm can be applied to create optimization systems for other objects, provided that the objective function and the restrictions to the requirements are satisfied. The two-stage approach used, including a one-time solution to the problem of unconditional optimization, will ensure the speed of such software systems.

References

1. Kunz, T. P., Crone, S. F., Meissner, J. (2016). The effect of data preprocessing on a retail price optimization system. Decision Support Systems, 84, 16–27. doi: https://doi.org/10.1016/j.dss.2016.01.003
2. Reisi, M., Gabriel, S. A., Fahimnia, B. (2019). Supply chain competition on shelf space and pricing for soft drinks: A bilevel optimization approach. International Journal of Production Economics, 211, 237–250. doi: https://doi.org/10.1016/j.ijpe.2018.12.018
3. Theysohn, S., Klein, K., Völckner, F., Spann, M. (2013). Dual effect-based market segmentation and price optimization. Journal of Business Research, 66 (4), 480–488. doi: https://doi.org/10.1016/j.jbusres.2011.11.007
4. Krasheninnikova, E., García, J., Maestre, R., Fernández, F. (2019). Reinforcement learning for pricing strategy optimization in the insurance industry. Engineering Applications of Artificial Intelligence, 80, 8–19. doi: https://doi.org/10.1016/j.engappai.2019.01.010

5. Gupta, V. K., Ting, Q. U., Tiwari, M. K. (2019). Multi-period price optimization problem for omnichannel retailers accounting for customer heterogeneity. International Journal of Production Economics, 212, 155–167. doi: https://doi.org/10.1016/j.ijpe.2019.02.016
6. Chen, R., Jiang, H. (2017). Capacitated assortment and price optimization for customers with disjoint consideration sets. Operations Research Letters, 45 (2), 170–174. doi: https://doi.org/10.1016/j.orl.2017.01.009
7. Chen, R., Jiang, H. (2019). Capacitated assortment and price optimization under the multilevel nested logit model. Operations Research Letters, 47 (1), 30–35. doi: https://doi.org/10.1016/j.orl.2018.11.006
8. Qu, T., Zhang, J. H., Chan, F. T. S., Srivastava, R. S., Tiwari, M. K., Park, W.-Y. (2017). Demand prediction and price optimization for semi-luxury supermarket segment. Computers & Industrial Engineering, 113, 91–102. doi: https://doi.org/10.1016/j.cie.2017.09.004
9. Gallego, G., Wang, R. (2014). Multiproduct Price Optimization and Competition Under the Nested Logit Model with Product-Differentiated Price Sensitivities. Operations Research, 62 (2), 450–461. doi: https://doi.org/10.1287/opre.2013.1249
10. Choi, T.-M., Ma, C., Shen, B., Sun, Q. (2019). Optimal pricing in mass customization supply chains with risk-averse agents and retail competition. Omega, 88, 150–161. doi: https://doi.org/10.1016/j.omega.2018.08.004
11. Ferreira, K. J., Lee, B. H. A., Simchi-Levi, D. (2016). Analytics for an Online Retailer: Demand Forecasting and Price Optimization. Manufacturing & Service Operations Management, 18 (1), 69–88. doi: https://doi.org/10.1287/msom.2015.0561
12. Caro, F., Gallien, J. (2012). Clearance Pricing Optimization for a Fast-Fashion Retailer. Operations Research, 60 (6), 1404–1422. doi: https://doi.org/10.1287/opre.1120.1102
13. Harsha, P., Subramanian, S., Ettl, M. (2019). A Practical Price Optimization Approach for Omnichannel Retailing. INFORMS Journal on Optimization, 1 (3), 241–264. doi: https://doi.org/10.1287/ijoo.2019.0018
14. Salvietti, L., Smith, N. R. (2008). A profit-maximizing economic lot scheduling problem with price optimization. European Journal of Operational Research, 184 (3), 900–914. doi: https://doi.org/10.1016/j.ejor.2006.11.031
15. Katsifou, A., Seifert, R. W., Tancrez, J.-S. (2014). Joint product assortment, inventory and price optimization to attract loyal and non-loyal customers. Omega, 46, 36–50. doi: https://doi.org/10.1016/j.omega.2014.02.002
16. Zaichenko, Yu. P. (2000). Doslidzhennia operatsiy. Kyiv: ZAT "VIPOL", 687.
17. Trunov, A. N. (2014). The increasing of degree of completeness for solution of financial and business problems in nonlinear programming with constraints inequalities. Journal of Computational Optimization in Economics and Finance, 243–253.
18. Trunov, A. N. (2015). Modernization of means for analyses and solution of nonlinear programming problems. Quantitative Methods in Economics, 16 (2), 133–141.
19. Gribanova, E. B. (2019). Reshenie zadachi optimizatsii tseny s pomoshch'yu obratnyh vychisleniy. Modelirovanie, optimizatsiya i informatsionnye tehnologii, 7 (3). doi: http://doi.org/10.26102/2310-6018/2019.26.3.014
20. Li, B., Nguyen, V. H., Ng, C. L., del Rio-Chanona, E. A., Vassiliadis, V. S., Arellano-Garcia, H. (2016). ICRS-Filter: A randomized direct search algorithm for constrained nonconvex optimization problems. Chemical Engineering Research and Design, 106, 178–190. doi: https://doi.org/10.1016/j.cherd.2015.12.001
21. El-Shorbagy, M. A., Mousa, A. A., Nasr, S. M. (2016). A chaos-based evolutionary algorithm for general nonlinear programming problems. Chaos, Solitons & Fractals, 85, 8–21. doi: https://doi.org/10.1016/j.chaos.2016.01.007
22. Mitsel', A. A., Hvashchevskiy, A. N. (1999). Noviy algoritm resheniya zadachi kvadratichnogo programmirovaniya. Avtometriya, 3, 93–98.
23. Odintsov, B. E. (2004). Obratnye vychisleniya v formirovanii ekonomicheskih resheniy. Moscow: Finansy i statistika, 256.
24. Gribanova, E. B. (2018). Methods for solving inverse problems of economic analysis by minimizing argument increments. Proceedings of Tomsk State University of Control Systems and Radioelectronics, 21 (2), 95–99. doi: https://doi.org/10.21293/1818-0442-2018-21-2-95-99