AN ALGORITHM FOR SOLVING INVERSE PROBLEMS CHARACTERIZED BY MULTI-EXTREMAL FUNCTIONS

*Ekaterina Gribanova**

Tomsk state university of control systems and radioelectronics, Tomsk
Russia

* Corresponding Author, e-mail: ekaterina.b.gribanova@tusur.ru

Abstract: This article addresses the solution of a single-point inverse problem for multi-extremal functions in the context of target indicator formation tasks. A description of the proposed algorithm is provided, which was developed based on an inverse approach and coordinate descent with stochastic argument selection. The algorithm was tested using well-known benchmark functions, including comparisons with methods implemented in existing Python libraries. The application of the algorithm for solving the practical problem of pricing under oscillatory demand is examined. The experimental results demonstrate the feasibility of applying the algorithm to solve problems involving multi-extremal functions.

Key words: multi-extremal function, global optimization, cyclical demand, stochastic algorithm, inverse problem

1. INTRODUCTION

Typically, inverse problems involve reconstructing the input parameters or system characteristics that yield particular observed or target outputs, whereas direct problems focus on computing the resulting effects from specified initial conditions. These characteristics underscore the distinction between the two approaches: in the case of a direct problem, there occurs an unambiguous progression from cause to effect, whereas an inverse problem often requires the specification of additional conditions for the reconstruction of the original parameters.

The diversity of inverse problem formulations gives rise to their conditional classification: retrospective, coefficient, boundary, etc. [1]. This study focuses on single-point inverse problems in which both the analytical form of the function and target output are specified. The goal is to find the input parameters that produce the given target value.

One of the primary challenges in solving inverse problems is the fact that different initial conditions can lead to identical outcomes. This results in non-uniqueness in the determination of solutions, which, in turn, necessitates the application of additional methods to resolve this ambiguity. Classical methods for solving inverse problems, including the quasi-solution method, discrepancy method, and regularization, involve the use of additional conditions to obtain a unique solution.

The incorporation of expert information also contributes to obtaining a unique solution to the inverse problem. This is particularly relevant for domains characterized by high levels of uncertainty, limited information channels in computer systems, and rapidly changing environmental conditions. These factors create significant difficulties in process forecasting and determine the necessity of utilizing expert knowledge and judgments based on intuition and practical experience. To solve economic problems, an apparatus of single-point reverse calculations was developed, which involves determining the necessary changes in arguments to achieve a specified function value. This process is based on initial argument values and uses expert information represented in the form of coefficients of relative importance of arguments and directions of their changes [2–4].

In real-world problems, the function may exhibit complex nonlinear behavior and possess multiple local extrema, which necessitates the application of global optimization methods. The most widely used global optimization methods are metaheuristic approaches, encompassing swarm intelligence, evolutionary techniques, and algorithms that simulate physical processes and human behavioral models [5–7]. Despite the advantages that heuristic methods offer, they are associated with notable drawbacks, including high computational and time costs, as well as the tendency to become trapped in local minima. This underscores the importance of exploring novel optimization methodologies. The present research focuses on examining the potential application of an inverse approach to address inverse single-point problems characterized by multi-extremal objective functions. The novelty of the research lies in the development of an algorithm for solving the inverse problem for a multiextremal function, which is based on an inverse approach: stochastic selection of the argument for modification and solving a single-variable equation.

2. PROBLEM STATEMENT

The primary objective of this research is to develop an algorithm for solving an inverse problem characterized by a multi-extremal function.

The input data for the investigated inverse problem include:

- initial argument values (x);
- relative importance coefficients of the arguments (β);
- target function value (y^*).

It is necessary to determine the changes in arguments (Δx) such that the function value f equals y^*:

$$f(x+\Delta x) = y^* . \tag{1}$$

The assumptions underlying this investigation are as follows: the objective function exhibits continuity throughout its domain, a global minimum exists for the optimization problem (1), and the function possesses differentiability properties.

Figure 1 illustrates the problem under consideration for the single-argument case. Point A represents the initial value of argument x; the task is to determine the required change in x to achieve the target function value y^*, avoiding the local minimum at point C.

To obtain a unique solution to problem (1), the following approaches can be used:

• L2 regularization: the solution to the problem is determined in such a way as to minimize the sum of squares of argument changes.

• L1 regularization: the solution to the problem aims to minimize the sum of absolute values of argument changes, which allows for zeroing out some arguments.

• Relative importance coefficients: the optimization problem is formulated so that argument changes correspond to predefined coefficients. This approach is applied when the solution must be found in accordance with a specific expert-defined achievement method

In this case, the general constrained optimization problem will have the form [4]:

$$\begin{aligned} g(x+\Delta x) &\to \min, \\ f(x+\Delta x) &= y^*, \end{aligned} \tag{2}$$

where g is a function that represents either the minimization of the sum of squares of arguments changes, or the sum of absolute values of arguments changes, or correspondence to relative importance coefficients

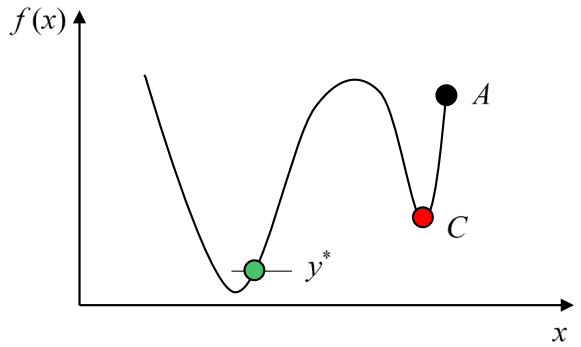

Figure 1. The problem of determining the argument magnitude to achieve the target function value

3. ALGORITHM FOR SOLVING INVERSE PROBLEMS WITH MULTI-EXTREMAL FUNCTIONS

The direct approach is a common strategy in the field of global optimization and is actively employed in gradient-based, metaheuristic, and other optimization methods. Simultaneously, the application of the inverse approach remains underexplored and is not as prominently represented in the scientific literature.

The developed algorithm is based on the idea of coordinate descent, according to which only one argument is selected for modification at each step. Each function argument has an indicator u that characterizes the possibility of using the argument in calculations. This indicator can take two values: one, which indicates the possibility of using the argument, and zero, which indicates the impossibility of using the argument.

The stochastic method presented in sources [4] served as the foundation for the algorithm development. The implemented modification includes dynamic adjustment of the step size, which contributed to improving both the accuracy and speed of problem solving. The proposed algorithm comprises the following steps.

Algorithm Inverse

Input: *initial values of arguments* x, *function* $f(x)$, *target function value* y^*, *number of arguments* n, *step size for function modification* Δy, *ratio of step size reduction* r, *modifying the step reduction factor* q, *upper limit of the step reduction factor* r_{max}, *indicator of the possibility of use in calculations* $u = 1$, *maximum iterations* N

Output: x, $f(x)$

$y = f(x)$, ***if*** $y < y^*$ ***then*** $t = 1$ ***else*** $t = -1$

for $v = 1, N$ ***do***

  $y_{last} = y$, $y = y + t\frac{\Delta y}{r}$

  $if\ (y > y^*$ ***and*** $t = 1)$ ***or*** $(y < y^*$ ***and*** $t = -1)$ ***then*** $y = y^*$

  for $m = 1, n$ ***do***

    if $u_m = 1$ ***then*** $\beta_m^* = \frac{\beta_m}{\sum_{j=1}^{n} u_j \cdot \beta_j}$ ***else*** $\beta_m^* = 0$

  end

  Randomly selecting argument k *according to probability* β^*

  if $u_i = 0\ (\forall i = 1, n)$ ***then***

    $r = r \cdot q$

    if $r > r_{max}$ ***then exit***

    else $u_i = 1\ (\forall i = 1, n)$, $y = y_{last}$, ***continuing to the next iteration in the loop***

  end

  Computing x_k^* *by resolving the equation* $f(x_k^*) = y$

  if $(f(x_k^*) < y_{last}$ ***and*** $t = 1)$***or***$(f(x_k^*) > y_{last}$ ***and*** $t = -1)$ ***then***

  $x_k = x_k^*$, $y = f(x_k^*)$, $u_i = 1\ (\forall i = 1, n)$ ***else*** $y = y_{last}, u_k = 0$

end

Figure 2 presents an illustration of the algorithm operation, where each move to a new point is accompanied by the following main actions: random selection of an argument from x_1 and x_2; modification of the selected argument such that the function value decreases by Δy.

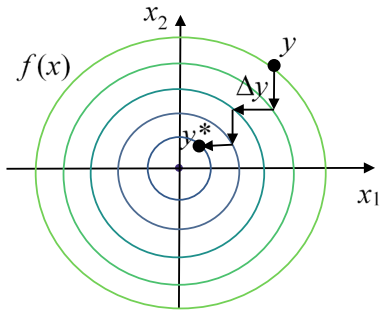

Figure 2. Illustration of the algorithm operation

The algorithm is metaheuristic and provides iterative solution improvement through the following rules: at each step, the y value approaches the target with a certain step size; a search is performed for the argument that ensures the target function value in the iteration.

4. COMPUTATIONAL EXPERIMENTS

4.1. Experiments using standard functions

For conducting computational experiments, six standard functions commonly used for testing optimization methods were selected (Table 1). Except for the Rosenbrock function, all considered functions are multimodal; Figure 3 shows the plot of the Schwefel function as an example. The implementation of the developed algorithm was performed in Python programming language. To compare the obtained results with those of alternative methods, an evaluation of solving the constrained optimization problem (2) was conducted when minimizing the sum of squares of argument changes. For this purpose, three well-known algorithms implemented in the SciPy Optimize library that allow solving problems with equality constraints were selected:

- Differential Evolution (DE): a metaheuristic differential evolution algorithm.
- Sequential Least Squares Programming (SLSQP): a sequential quadratic programming algorithm with constraints.
- Trust-region method for constrained optimization (trust-constr): an algorithm for iterative formation of approximations within a dynamically adjustable trust region.

DE belongs to the class of global optimization methods, whereas SLSQP and trust-constr are categorized as local search techniques. For the experimental investigations, the default parameter settings were employed for these established methods. Regarding the Inverse algorithm, the following parameter configuration was adopted: $\Delta y = 1000$, $r_{max} = 10^{18}$, with the maximum number of iterations (maxiter) and tolerance (tol) for the Newton method used in root-finding procedures set to 100 and 10^{-5}, respectively. For the Rosenbrock function with 5 and 10 variables, the parameters were set as follows: $\Delta y = 100$, maxiter = 150, tol = 10^{-6}. Starting from 50 variables, the parameters were modified: $\Delta y = 10^4$, maxiter = 100, tol = 10^{-3}. This parameter adjustment is attributed to the fact that optimization of this type of function represents a computationally expensive process. The Rosenbrock function is characterized by a narrow parabolic flat valley; consequently, improvement in the function value can only be achieved through small incremental steps.

For root finding, both the classical Newton's method and more contemporary approaches described in references [8–9] were examined, with the primary advantage of the latter being the elimination of the need for derivative calculations. Since Newton's method demonstrated the best performance in most of the conducted experiments, it was selected to obtain the presented results.

In the first part of algorithm testing, an evaluation of their ability to find the target value near the global minimum was conducted. It should be noted that the importance coefficients for all arguments were equal in this case, which allowed focusing on the overall algorithm efficiency without considering the influence of argument weights.

The global minimum for all functions presented in Table 1 is located at point 0. The target value was set to 0.1, and the coordinates of the initial point were randomly generated within the interval specified in Table 1.

Based on the obtained results, Inverse demonstrates the most stable results on most functions and dimensions (Table 2). DE performs well on low dimensions, but its effectiveness decreases with increasing number of arguments. SLSQP is effective for certain functions, but shows heterogeneous results; in particular, it performs worse in determining the target value with fewer than 10 parameters, when the function landscape is very noisy with a large number of local minima. trust-constr is the least effective method for most functions, especially at high dimensions. Comparing the problem-solving times of the Inverse approach and DE, it is evident that the Inverse method outperforms DE in most experiments (Table 3).

Table 1. Standard functions

Function ID	Function name	Search interval
f1	Rastrigin	(-100, 100)
f2	Rosenbrok	(-3, 3)
f3	Ackley	(-33, 33)
f4	Griewank	(-600, 600)
f5	Schwefel	(-500, 500)
f6	Alpine 1	(-10, 10)

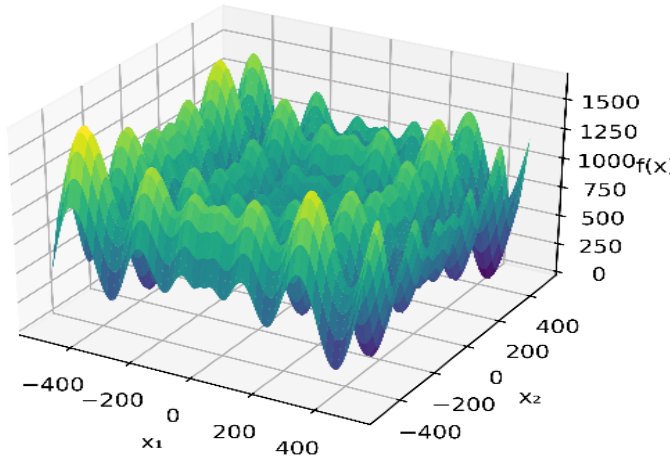

Figure 3. Schwefel function

Table 2. Mean function value

Dimension	Method	Function ID					
		f1	f2	f3	f4	f5	f6
5	Inverse	**0.1**	0.1	0.1	0.1	118.93	0.1
	DE	0.28	0.87	0.1	0.1	0.1	0.1
	SLSQP	9.2	0.1	2.3	0.44	1025.75	0.1
	trust-constr	481.5	0.1	16.55	2.36	995.68	0.58
10	Inverse	**0.1**	0.1	0.1	0.1	321.16	0.1
	DE	0.46	0.1	0.1	0.1	11.93	0.1
	SLSQP	0.58	0.1	0.1	0.11	2695.	0.1
	trust-constr	583.39	0.12	12.63	0.16	2945.77	0.36
50	Inverse	0.1	**0.48**	0.1	0.1	**928.86**	0.1
	DE	77.9	29.36	0.1	0.1	13169.46	0.1
	SLSQP	0.1	47.97	0.1	0.1	12107.9	0.1
	trust-constr	3225.97	37.28	8.5	0.1	16425.93	1.95
100	Inverse	0.1	**77.83**	0.1	0.1	**2025.59**	0.1
	DE	206.2	120.91	0.1	0.1	36720.14	0.4
	SLSQP	0.1	97.15	0.1	0.1	29242.67	0.1
	trust-constr	5870.33	88.68	6.57	0.1	31540.56	2.3
200	Inverse	0.1	389	0.1	0.1	**3837.38**	0.1
	DE	1787.4	284.55	0.95	0.1	72348.86	0.85
	SLSQP	0.1	195.99	0.1	0.1	61817.03	0.1
	trust-constr	6816.39	185.29	7.42	0.1	73703.75	11.28

Table 3. Mean problem solving time

Dimension	Method	Function ID					
		f1	f2	f3	f4	f5	f6
5	Inverse	0.1	11.2	1	0.4	0.6	0.19
	DE	14.3	12.15	12.8	12.87	10.45	11
	SLSQP	0.1	0.04	0.04	0.07	0.12	0.03
	trust-constr	1.9	0.65	2.05	0.65	2.22	0.99
10	Inverse	0.05	36	2.26	3.4	2	0.5
	DE	28	26.57	23.94	25.12	21.2	21.17
	SLSQP	0.05	0.13	0.02	0.09	0.3	0.07
	trust-constr	2.08	2.77	1.82	0.69	1.96	1.91
50	Inverse	20.7	157.6	22.6	32.2	25.6	0.5
	DE	145.2	281.72	161.04	153.1	149.01	138.14
	SLSQP	0.12	1.03	0.04	0.12	0.67	0.09
	trust-constr	34.92	45.67	15.69	6.84	15.98	23.77
100	Inverse	105	199.2	67	120.6	61.6	13.8
	DE	359.68	782.07	357.79	347.9	296.38	307.39
	SLSQP	0.21	3.02	0.13	0.4	1.61	0.12
	trust-constr	65.85	110.12	40.43	9.26	40.37	46.93
200	Inverse	661	1658	206	336.8	197.2	40.6
	DE	1008.9	2421.38	945.64	936.7	820.51	823.16
	SLSQP	0.8	13	0.76	1.48	8.4	0.53
	trust-constr	156.86	241.13	150.46	33.31	69.76	137.13

An evaluation of the algorithms was also conducted in terms of the correspondence between the obtained solutions and the established relative importance coefficients. For this purpose, a three-argument Rastrigin function was considered, with importance coefficients of the first, second, and third arguments set to 0.5, 0.3, and 0.2, respectively, and initial argument values equal to 50. The target function value y^* was set equal to 1000. For the application of SLSQP, DE, and trust-constr algorithms, problem (2) was considered, where the objective function includes weighting coefficients β [4]:

$$g(\Delta x) = \frac{1}{\beta_1}\Delta x_1^{\,2} + \frac{1}{\beta_2}\Delta x_2^{\,2} + \frac{1}{\beta_3}\Delta x_3^{\,2} \qquad (3)$$

All considered algorithms successfully achieved the target value. The average Euclidean norm distance between the obtained argument change values and the established importance coefficient magnitudes over 10 random realizations is presented in Figure 4. The argument ratios most consistent with the relative importance coefficients were obtained using the Inverse algorithm (Δy=100). This is explained by the fact that the probability of selecting an argument for adjustment is proportional to the importance coefficient, which ensures the specified proportion of argument changes.

It should be noted that using Newton's method for finding a root requires specifying only an initial point, not an interval. However, if besides importance coefficients it is also necessary to specify the direction of argument changes [2], then in this case interval methods such as the bisection method are more suitable. The formation of the interval for finding the root is carried out as follows: if the argument change should be positive, then the interval will be [x, b], otherwise – [a, x] (where a and b are the lower and upper bounds of the search interval, x is the current value of the argument). Thus, for solving the target value formation problem for the Rastrigin function, the following parameters were established: initial argument values equal to 2, target value of 1000, importance coefficients for the first, second, and third arguments set to 0.5, 0.3, and 0.2, respectively. The solution was required to be found with positive argument variation. The obtained argument values were 21.18, 17.3, and 15.15. For the DE, SLSQP, and trust-constr methods, argument bounds from 2 to 100 were established for problem solving to ensure their positive variation. The trust-constr method did not reach the target value ($f(x)$ = 96.94). The mean values of the Euclidean distance metric between the obtained argument variations and the relative importance coefficients are presented in Figure 4.

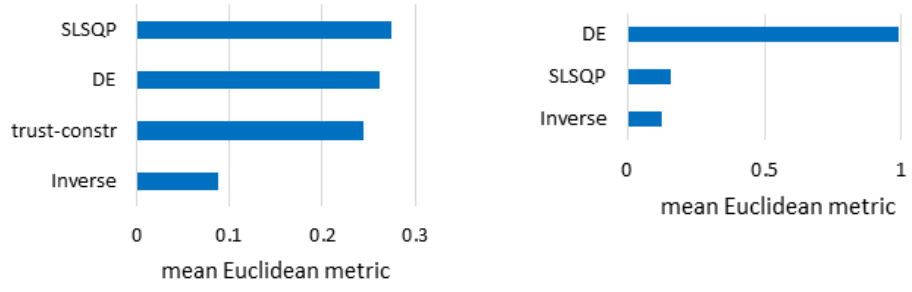

Figure 4. Euclidean distance metric of argument variations from the established importance coefficients: arbitrary direction of argument changes (left); positive argument changes (right)

4.2. Experiments on solving the pricing problem under oscillatory demand

Oscillatory demand [10] represents an economic phenomenon in which the volume of demand for a good or service is subject to periodic fluctuations (oscillations) over time. These fluctuations may be caused by various factors: seasonality, economic cycles, changes in consumer preferences, competitors' marketing campaigns, and other market forces.

Under conditions of oscillatory demand, traditional pricing models prove to be ineffective, as they fail to account for the cyclical nature of demand variation. This creates the necessity for developing a model for optimal price determination that is capable of adapting to demand fluctuations.

In the problem under consideration, seasonal demand is examined, whose maximum potential value is also subject to fluctuations within the working week (Figure 5):

$$d(t) = 150 - \frac{(-25 + 0.5t)^2 + 50\cos\left(\frac{2\pi}{7} \cdot t\right)}{5}$$

where t is time over the considered time horizon T (T=100);

$d(t)$ is potential demand dependent on time.

The demand realization coefficient depends on the established price x and is determined by the following relationship (Figure 5):

$$p(x) = e^{-0.1x},$$

where $p(x)$ is the demand realization coefficient.

Thus, actual demand will constitute the product of maximum demand and the demand realization coefficient: $d(t) \cdot p(x)$.

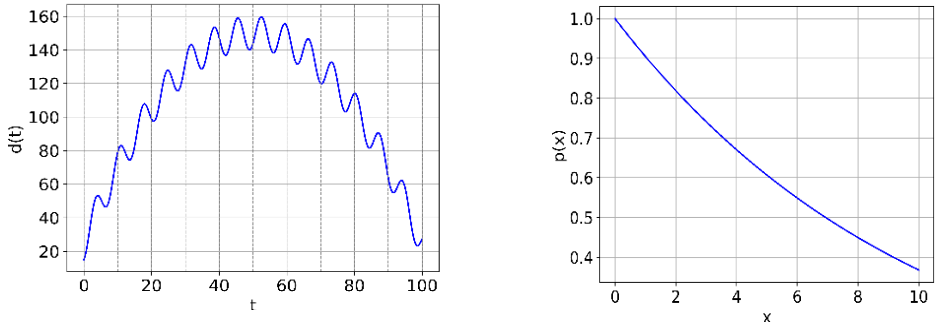

Figure 5. Maximum demand (left) and dependence of demand realization coefficient on price (right)

Within the framework of this study, two problem formulations are considered. The first problem formulation consists in determining the price and time point for achieving target profit. In this problem, it is necessary to determine the price x and time point t on the interval [0, T] such that profit reaches the target value y^*:

$$g(x,t) = x^2 + t^2 \to \min,$$
$$f(x,t) = d(t) \cdot p(x) \cdot (x - h(t)) = y^*,$$
$$x_{\min} \le x \le x_{\max},$$
$$0 \le t \le T,$$

where $h(t)$ is the cost function per unit of product, which varies over time (Figure 6):

$$h(t) = 8 - \frac{1}{500}(t-40)^2.$$

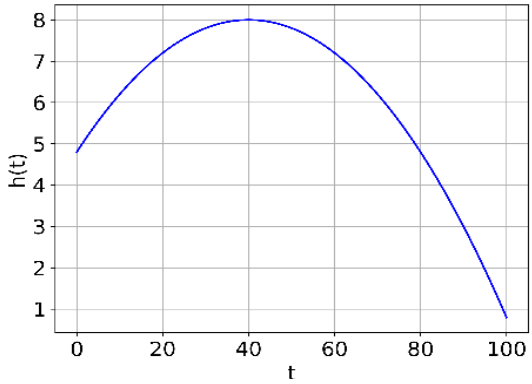

Figure 6. Cost dependence on time

By setting the target value equal to some large unattainable number, the profit maximization problem can be solved. Thus, Table 4 presents the results obtained with the following parameters: y^*=10,000, T=100, $5 \le x \le 100$, $0 \le t \le 100$. The maximum profit was determined using the Inverse and DE methods.

Table 4. Results of solving the price determination problem for profit maximization

Method	x	t	$f(x)$	$g(x)$	Time, sec.	Memory, Mib
Inverse, Δy=100000, $r_{max}=10^{20}$	16.6	66.49	278.47	4696.09	0.24	<0.01
DE	16,6	66.49	278.47	4696.09	20	0.58
SLSQP	11.46	3.78	102.24	145.57	0.17	0.09
trust-constr	15	4	113.98	253.26	2.9	0.02

The second problem formulation consists of determining the price with constant production volume. This problem considers pricing to achieve target demand d^* under conditions of fixed production volume over a specified period (Figure 5). Thus, for each time segment, it is necessary to determine the price so that demand reaches the target value d^*.

To determine demand over a time interval, both numerical integration methods (trapezoidal rule, rectangular rule) and analytical solutions can be used:

$$I(a,b)=\int_a^b 150-\frac{\left(-25+0.5t\right)^2+50\cos\left(\frac{2\pi}{7}\cdot t\right)}{5}dt=25b-25a+11.14\cdot\sin\left(0.898a\right)-$$

$$11.14\cdot\sin\left(0.898b\right)-2.5a^2+0.0167a^3+2.5b^2-0.0167b^3$$

The optimization problem when the number of segments equals 10 has the following form:

$$g(x)=\sum_{j=0}^{9}x_j^2\rightarrow\min,$$

$$f(x)=\sum_{j=0}^{9}\left(p\left(x_j\right)\cdot I(j\cdot 10,j\cdot 10+10)-d^*\right)^2=0,$$

$$x\geq x_{\min}.$$

Table 5 presents the results of solving the problem with d^*=300, $x_{\min}$=2. Since in this case the function is not multi-extremal, all methods provided a solution to the problem with high accuracy. The obtained variable values for all methods: x_1=4.67, x_2=10.95, x_3=13.56, x_4=15.44, x_5=15.85, x_6=16.07, x_7=15.25, x_8=13.68, x_9=10.95, x_{10}=4.36. Thus, as potential demand increases, price grows, and on intervals with minimum demand, price is at its minimum level.

Table 5. Results of solving the problem with constant production volume

Method	$f(x)$	$g(x)$	Time, sec.	Memory, Mib
Inverse, Δy=100000, $r_{\max}=10^{17}$	$1.26\cdot10^{-12}$	1632.15	13	0.05
DE	$5.46\cdot10^{-11}$	1632.15	30	0.21
SLSQP	$4.45\cdot10^{-6}$	1632.14	0.13	<0.01
trust-constr	$5.3\cdot10^{-11}$	1632.15	1.42	0.02

5. CONCLUSION

This work addresses the development of an algorithm for solving inverse problems with multi-extremal functions. Within the framework of this research, an algorithm has been developed that employs random argument selection taking into account relative importance coefficients. The algorithm testing was conducted using standard test functions, such as Rosenbrock, Rastrigin, Ackley functions, and others. The algorithms were evaluated according to two criteria: the ability to achieve target values and compliance with established relative importance coefficients. Functions included in Python libraries were used for comparison of results. The application of the algorithm for solving pricing problems in two formulations was also considered. The results of the conducted experiments confirm the feasibility of applying the proposed algorithm for solving the investigated problems. Future research directions will focus on the application of the developed algorithms for solving other machine learning problems, such as data clustering.

ACKNOWLEDGEMENT

This study was funded by the Russian Science Foundation, grant number 25-21-00123 «Application of methods for solving inverse ill-posed problems in machine learning»

REFERENCES

[1] Kabanikhin, S. Inverse problems and artificial intelligence. *Russian Journal of Cybernetics*, vol.2, no.3, 2021, pp. 33-43. doi: https://doi.org/10.51790/2712-9942-2021-2-3-5

[2] Odintsov, B.E. *Inverse calculations in the formation of economic decisions*. Finance and Statistics, Moscow, 2024, (193 p.)

[3] Gribanova, E., Savitsky, A. Algorithm for Estimating the Time of Posting Messages on Vkontakte Online Social Network. *International Journal on Information Technologies and Security*, vol.12, no.1, 2020, pp. 3-14.

[4] Gribanova, E. Elaboration of an Algorithm for Solving Hierarchical Inverse Problems in Applied Economics. *Mathematics*, vol.10, no.15, 2022, 2779. doi: https://doi.org/10.3390/math10152779

[5] Nugraheni, C. E., Abednego, L., Widyarini, M. Whale optimization algorithm for production scheduling problems in small bakery enterprises. *International Journal on Information Technologies and Security*, vol.17, no.1, 2025, pp. 57-68. doi: https://doi.org/10.59035/BSWR7756

[6] Yiying, Z., Zhigang, J. Group teaching optimization algorithm: A novel metaheuristic method for solving global optimization problems. *Expert Systems with Applications*, vol.148, 2020, 113246. doi: https://doi.org/10.1016/j.eswa.2020.11324

[7] Guilmeau, T., Chouzenoux, E., Elvira, V. Simulated Annealing: a Review and a New Scheme. *2021 IEEE Statistical Signal Processing Workshop (SSP)*, Rio de Janeiro, Brazil, July 2021, pp. 101-105. DOI: 10.1109/SSP49050.2021.9513782.

[8] Eskandari, H. Hybrid Steffensen's Method for Solving Nonlinear Equation. *Applied Mathematics*, vol.13, No.9, 2022. doi: https://doi.org/10.4236/am.2022.139046

[9] Neta, B. A New Derivative-Free Method to Solve Nonlinear Equations. *Mathematics*, vol.9, no.6, 2021, pp. 583. doi: https://doi.org/10.3390/math9060583

[10] Klug, F. Modelling oscillations in the supply chain: the case of a just-in-sequence supply process from the automotive industry. *Journal of Business Economics*, vol.92, 2022, pp.85–113. doi: https://doi.org/10.1007/s11573-021-01049-6

Information about the authors:

Gribanova Ekaterina – Head of image processing and artificial intelligence laboratory Tomsk state university of control systems and radioelectronics. Areas of Scientific Research: economic modeling, optimization methods, inverse problems, information technology.

Manuscript received on 12 Jine 2025