Mathematical modeling

Математическое моделирование

UDC 519.857
https://doi.org/10.32362/2500-316X-2023-11-2-72-83

RESEARCH ARTICLE

Optimization of spline parameters in approximation of multivalued functions

Dmitry A. Karpov,
Valery I. Struchenkov @

MIREA – Russian Technological University, Moscow, 119454 Russia
@ Corresponding author, e-mail: str1942@mail.ru

Abstract
Objectives. Methods for spline approximation of a sequence of points in a plane are increasingly used in various disciplines. A spline is defined as a single-valued function consisting of a known number of repeating elements, of which the most widely used are polynomials. When designing the routes of linear structures, it is necessary to consider a problem with an unknown number of elements. An algorithm implemented for solving this problem when designing a longitudinal profile was published earlier. Here, since the spline elements comprise circular arcs conjugated by line segments, the spline is a single-valued function. However, when designing a route plan, the spline is generally a multivalued function. Therefore, the previously developed algorithm is unsuitable for solving this problem, even if the same spline elements are used. The aim of this work is to generalize the obtained results to the case of approximation of multivalued functions while considering various features involved in designing the routes of linear structures. The first stage of this work consisted in determining the number of elements of the approximating spline using dynamic programming. In the present paper, the next stage of solving this problem is carried out.
Methods. The spline parameters were optimized using a new mathematical model in the form of a modified Lagrange function and a special nonlinear programming algorithm. In this case, it is possible to analytically calculate the derivatives of the objective function with respect to the spline parameters in the absence of its analytical expression.
Results. A mathematical model and algorithm were developed to optimize the parameters of a spline as a multivalued function consisting of circular arcs conjugated by line segments. The initial approximation is the spline obtained at the first stage.
Conclusions. The previously proposed two-stage spline approximation scheme for an unknown number of spline elements is also suitable for approximating multivalued functions given by a sequence of points in a plane, in particular, for designing a plan of routes for linear structures.

Keywords: route, plan and longitudinal profile, spline, nonlinear programming, objective function, constraints

• **Submitted:** 02.03.2022 • **Revised:** 01.06.2022 • **Accepted:** 26.01.2023

For citation: Karpov D.A., Struchenkov V.I. Optimization of spline parameters in approximation of multivalued functions. *Russ. Technol. J.* 2023;11(2):72–83. https://doi.org/10.32362/2500-316X-2023-11-2-72-83

Financial disclosure: The authors have no a financial or property interest in any material or method mentioned.

The authors declare no conflicts of interest.

НАУЧНАЯ СТАТЬЯ

Оптимизация параметров сплайна при аппроксимации многозначных функций

Д.А. Карпов,
В.И. Струченков @

МИРЭА – Российский технологический университет, Москва, 119454 Россия
@ Автор для переписки, e-mail: str1942@mail.ru

Резюме
Цели. Методы сплайн-аппроксимации последовательности точек на плоскости получают все более широкое применение в различных областях. Сплайн рассматривается как однозначная функция с известным числом повторяющихся элементов. Наиболее широкое применение получили полиномиальные сплайны. Применительно к проектированию трасс линейных сооружений приходится рассматривать задачу с неизвестным числом элементов. Алгоритм решения задачи применительно к проектированию продольного профиля реализован и опубликован ранее. В этой задаче элементами сплайна являются дуги окружностей, сопрягаемые отрезками прямых, и сплайн представляет собой однозначную функцию. Однако при проектировании плана трассы в общем случае сплайн является многозначной функцией. Поэтому разработанный ранее алгоритм не пригоден для решения этой задачи, даже в случае использования тех же элементов сплайна. Цель настоящей статьи – обобщение полученных результатов на случай аппроксимации многозначных функций с учетом особенностей проектирования трасс линейных сооружений. На первом этапе работы было определено число элементов аппроксимирующего сплайна с помощью динамического программирования. В статье рассматривается следующий этап решения задачи.
Методы. Для оптимизации параметров сплайна используется новая математическая модель в виде модифицированной функции Лагранжа и специальный алгоритм нелинейного программирования. При этом удается вычислять аналитически производные целевой функции по параметрам сплайна при отсутствии ее аналитического выражения через эти параметры.
Результаты. Разработаны математическая модель и алгоритм оптимизации параметров сплайна (как многозначной функции), состоящего из дуг окружностей, сопрягаемых отрезками прямых. Начальным приближением является сплайн, полученный на первом этапе.
Выводы. Двухэтапная схема сплайн-аппроксимации при неизвестном числе элементов сплайна, предложенная ранее, пригодна и для аппроксимации многозначных функций, заданных последовательностью точек на плоскости, в частности для проектирования плана трасс линейных сооружений.

Ключевые слова: трасса, план и продольный профиль, сплайн, нелинейное программирование, целевая функция, ограничения

• **Поступила:** 02.03.2022 • **Доработана:** 01.06.2022 • **Принята к опубликованию:** 26.01.2023

Для цитирования: Карпов Д.А., Струченков В.И. Оптимизация параметров сплайна при аппроксимации многозначных функций. *Russ. Technol. J.* 2023;11(2):72–83. https://doi.org/10.32362/2500-316X-2023-11-2-72-83

Прозрачность финансовой деятельности: Авторы не имеют финансовой заинтересованности в представленных материалах или методах.

Авторы заявляют об отсутствии конфликта интересов.

INTRODUCTION

The previously proposed [1] method for approximating multivalued functions defined discretely by a special type of spline uses a two-stage scheme for solving the problem. At the first stage, the number of elements of the spline along with a calculation of the approximate spline parameter values is determined using dynamic programming method. At the second stage, the parameters are optimized by nonlinear programming using the spline obtained at the first stage as an initial approximation. The first stage was performed in our previous work [2]. The present article, which is a continuation of that work, considers the second stage.

A spline represents a chain of repeating "circular arc + + line segment" elements. At this stage, the starting point, both the direction of the tangent at this point, as well as the lengths of all the arcs and their conjugating line segments, are known. Despite the fact that the desired spline is a multivalued function, this allows continuous optimization methods to be applied—in particular, methods of nonlinear programming of the gradient type.

Optimization of the parameters of the spline obtained at the first stage is necessary not only due to the insufficient accuracy of the solution of the problem at the first stage, which is due to the discreteness of the search, but also because of the impossibility of strictly imposing the constraints on fixed points at the first stage, i.e., the starting points, which are not displaced during the approximation.

As is common practice in dynamic programming, accuracy can be improved by repeating the calculations at smaller search steps. In this problem, this is particularly important because, at a known number of elements, the amount of computation is sharply reduced, which enables one to solve the problem at reduced discretes with an increase in their number in a reasonable time on public computers.

The problem is considered as applied to designing a plan for the routes of linear structures. For some of them, e.g., for a trench for laying pipelines of various purposes, the spline of the considered type is final. When designing horizontal road alignment, straight lines and circles should be conjugated by clothoids to ensure continuity not only of the tangent, but also of the curvature. It was shown [2] that, if the clothoids are short, their addition to the resulting spline with circles leads to insignificant displacements. However, for the general case, it is necessary to implement a step-by-step spline approximation scheme with repeating elements "straight line + clothoid + circle + clothoid." The solution of this problem will be the subject of further research.

As shown previously [2], this approach differs significantly from the method of selecting elements in interactive mode as accepted in design practice, from various semiautomatic methods for searching for curve boundaries based on curvature graphs and angular diagrams, as well as from a novel heuristic method for searching curve boundaries [3] with subsequent application of genetic algorithms [4–12]. In contrast, the use of adequate mathematical models and mathematically correct algorithms seems to be more promising.

1. PROBLEM STATEMENT AND ITS FORMALIZATION

The problem is to find a spline of a given type that satisfies all the constraints and best approximates a given sequence of points in a plane [2, Fig. 1].

The starting point A and the direction of the tangent to the desired spline at this point are set and remain unchanged during the search for the spline. The quality of the approximation is estimated by the sum of the squared deviations h_i of the given points of the spline.

It is necessary to find

$$\min F(\mathbf{h}) = \sum_1^n h_j^2. \tag{1}$$

Here, $\mathbf{h}(h_1, h_2, \ldots, h_n)$ is the vector of unknowns, while n is their number. Instead of a simple sum, a weighted sum of squares can be given.

Deviations h_j are calculated differently in design practice in different countries and in corresponding studies carried out by various researchers. Typically, the deviation of a point from a spline is calculated along the normal to the spline [2]. In Russia, however, it is customary to calculate the deviation along the normal to the original polyline [2], i.e., toward the center of the circle connecting three adjacent points. If three points lie on the same straight line, then the deviation is calculated along the normal to this straight line.

Since the noted difference in calculation methods does not affect the search for the number of spline elements, the simplest method was adopted at the first stage, i.e., calculation along the normal to the desired spline. At the second stage, when optimizing the spline parameters, we use precomputed normals to the original polyline, i.e., fixed directions that do not need to be recalculated in an iterative process. These are the same normals that contain the points that determine the "states of the system" in dynamic programming [2, Fig. 2].

The starting point of the first curve may not coincide with the starting point A; therefore, the length L_1 of the initial line is considered unknown, and, unlike the first stage, the spline elements are considered in the order "straight line + circle". If the number of such repeating elements is m, then the system of constraints has the form

$$L_j^{\text{line}} \geq L_{\min}^{\text{line}}, \tag{2}$$

$$L_j^{\text{curve}} \geq L_{\min}^{\text{curve}}, \tag{3}$$

$$R_{\min} \leq \left|R_j\right| \leq R_{\max},\ j = \overline{1,m}. \tag{4}$$

Here, as at the first stage [2], L_j^{line} and L_j^{curve} are the lengths of the straight line and the curve in the jth element, respectively, while R_j are the radii of the circles, whose signs are known. This makes it possible to avoid taking the absolute value in constraint (4) and obtain a linear constraint on each R_j in the form of a two-sided inequality:

$$R_{\min} \leq R_j \leq R_{\max},\ \text{if } R_j > 0, \tag{5}$$

$$-R_{\max} \leq R_j \leq -R_{\min},\ \text{if } R_j < 0. \tag{6}$$

The end point of the spline is fixed, but its length is unlimited. If the final direction is also fixed, then constraints are imposed not only on h_n, but also on h_{n-1}. In addition, constraints can also be imposed on the displacements of individual points in the form of both inequalities, including double ones,

$$h_{\min} \leq h_m \leq h_{\max}, \tag{7}$$

and equalities,

$$h_m = h_0. \tag{8}$$

These are the same fixed points whose presence cannot be taken into account in dynamic programming.

As a result, we obtain a nonlinear programming problem with the objective function $F(\mathbf{h})$ under constraints (2), (3), and (5)–(8), some of which may be absent.

2. FEATURES OF THE PROBLEM

Constraints (2), (3), (5), and (6) are not expressed in terms of unknown displacements h_i, but if all the lengths and radii are known, then all the h_j can be calculated. Further, all the lengths and radii are considered as the main variables, and all the h_i are regarded as intermediate variables, which depend on the main ones. Analytical expressions of these dependences are unknown and will not be determined. There is also no analytical expression of the objective function $F(\mathbf{h})$ in terms of the main variables. As a result, we obtain a nonlinear programming problem under a simple system of constraints (2), (3) and (5), (6) on the main variables, under several constraints (7), (8) on intermediate variables, and with the objective function expressed in terms of intermediate variables.

Nonlinear programming algorithms, with all their diversity[1] [13–26], reduce to an iterative process with the following steps:

1) construct an admissible initial approximation;
2) determine the direction of descent from the next iteration point, in particular, from the starting point;
3) check the conditions for terminating the account. If they are not met, then go to the next item, otherwise, end the calculations;
4) find the step in the found direction from the condition that the constraints are satisfied and the minimum point in the direction is reached;
5) go to a new point, and then go to step 2.

In order to solve our problem, we need to repeatedly calculate intermediate variables (normal displacements) as the main variables are changed. To do this, the intersection points of two straight lines and a straight line with a circle have to be found (Fig. 1).

The shifts of the initial points to the design position are considered positive if they are directed along with the outward normal.

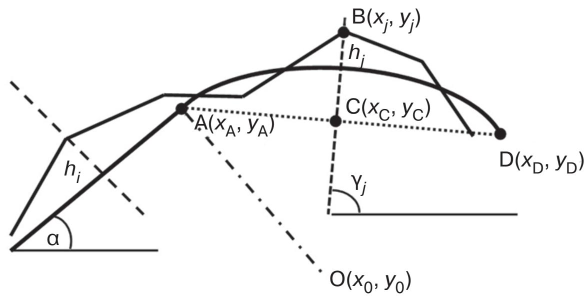

Fig. 1. Calculation of normal displacements

Let x_A and y_A be the coordinates of the beginning of the arc of the circle (point A in Fig. 1) and α is the angle of the tangent at this point with the OX axis. Then the coordinates of the center of the circle are written as $x_0 = x_A - R\sin\alpha$ and $y_0 = y_A + R\cos\alpha$. Here and henceforth, the radius is positive when moving along the curve counterclockwise. The point C of intersection with the normal can be both outside and inside the arc of the circle. Without loss of generality, for the point of intersection of the normal with the circle, one can write $x_C = x_j + h_j\cos\gamma_j$ and $y_C = y_j + h_j\sin\gamma_j$. Here and henceforth, γ_j is the angle of the jth normal with the OX axis.

From the condition that the point C belongs to the circle, one can obtain a quadratic equation for h_j, the solution of which gives a formula for h_j:

$$h_j = (x_A - R\sin\alpha - x_j)\cos\gamma_j + (y_A + R\cos\alpha - y_j)\sin\gamma_j \pm \\ \pm\sqrt{(R^2 - [(x_A - R\sin\alpha - x_j)\sin\gamma_j - (y_A + R\cos\alpha - y_j)\cos\gamma_j]^2)}. \tag{9}$$

[1] Panteleev A.V., Letova T.A. *Optimization methods*: A handbook. Moscow: Logos; 2011. 424 p. ISBN 978-5-98704-540-4 (in Russ.).

To select a point on the circle that is closest to the analyzed point, a minus sign is placed in front of the root if the expression in front of the root in formula (9) is positive, and *vice versa*. The deviations h_j are calculated sequentially from the starting point to the end point. In this case, for angles of rotation of the arcs of circles that are greater than π, there are features of determining whether or not the point of intersection of the circle and the normal (point C in Fig. 1) falls inside the arc. For example, in the general case, it cannot be assumed that, if chord AC < AD (Fig. 1), then point C will fall inside the arc AD (Fig. 2).

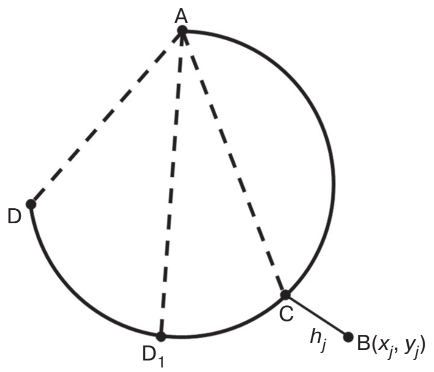

Fig. 2. Determination of whether or not the normal and the arc intersect

In the example in Fig. 2, AC > AD, but AC < AD_1, and the rule of determining the position of a point on an arc by comparing the lengths of the chords does not work.

The position of the point C with respect to the chord AD (Fig. 1) is determined by the sign of *d*.

$$d = (y_C - y_A)(x_D - x_A) - (x_C - x_A)(y_D - y_A).$$

If $d > 0$, then the point C is to the left of the direction of AD, while if $d < 0$, it is to the right.

This can be easily verified by passing to a coordinate system centered at the point A and directing the *OX* axis along AD. Hence, we obtain the rule: if $Rd > 0$, then the point C is outside the arc AD; otherwise, it is inside.

Another feature of the problem being solved is that the admissible domain is unlimited due to the one-sidedness of because inequalities (2) and (3). However, this circumstance is not significant in this case because the search for a step in the direction at each iteration can be limited to a maximum increase of 1 m in the radii and in the lengths of line segments and circular arcs.

A more important complicating feature is the already noted absence of an analytical expression for the objective function in terms of the main variables. On the other hand, a significant simplification consists in an extremely simple form of constraints on the main variables, owing to which the advisability of changing the set of active constraints on the main variables is easy to check by considering the sign of the corresponding components of the gradient at each iteration when searching for the direction of descent.

3. CALCULATION OF THE DERIVATIVES OF THE OBJECTIVE FUNCTION WITH RESPECT TO THE MAIN VARIABLES

Here, we are talking about an attempt to analytically calculate the gradient of the objective function with respect to the main variables without having its analytical expression in terms of these variables by recalculating the derivatives. It turned out that, in the context of our problem, such a recalculation is quite possible.

Let us assume that the set of the main variables forms vector **x**. Then the formula for recalculating the derivatives has the form

$$\frac{\partial F}{\partial x_i} = \sum_1^n \frac{\partial F}{\partial h_j} \cdot \frac{\partial h_j}{\partial x_i},\ i = \overline{1, n}, \tag{10}$$

where $\frac{\partial F}{\partial h_j} = 2h_j$ on the strength of expression (1).

This comes down to a calculation of the derivatives of the displacements with respect to the normals in terms of the main variables. Let us show how this can be done in our case. To do this while omitting the subscripts and keeping the notation *R* for the radius of an arbitrary circular arc, we denote the length of an arbitrary line segment by *l* and the length of the circular arc by *L*.

Let us start with the length of the line segment and assign it increment δ*l* without changing all other lengths and radii. Obviously, the desired $\frac{\partial h_j}{\partial l} = 0$ for all normals that are closer to the start of the spline than the end of the line segment being varied.

If we find the displacement δh_j of the point of intersection of the *j*th normal with the spline along this normal caused by change δ*l* at unchanged values of all the other variables, then, by passing to the limit in $\delta hj/\delta l$ as $\delta l \to 0$, we obtain the desired derivative without having an analytical expression for the function $h_j(l)$.

An increase in the length of the line segment by δ*l* at unchanged values of all the other variables results in a shift of the entire remaining part of the spline in the direction of this straight line by δ*l*. This is the simplest variation of the spline. If the point of intersection of the spline and the *j*th normal lies on the straight line (Fig. 3), then

$$\frac{\partial h_j}{\partial l} = \frac{\sin(\alpha - \beta)}{\sin(\gamma_j - \beta)}, \tag{11}$$

where β is the angle of this line (spline element AB in Fig. 3) with the *OX* axis; α is the angle of the line being varied with the *OX* axis (establishing the direction of the displacement); γ_j is the angle of the normal (C_0C_1 in Fig. 3) with the *OX* axis.

In Fig. 3, point C is the initial position of the point of intersection of the normal and the spline, which corresponds to the intermediate variable h_j. When shifting in the direction determined by the angle α by δl, AB transforms into A_1B_1, point C transforms into C_2, while C_1 becomes the point of intersection of the normal with the spline. The displacement h_j gets an increment of $h_j = CC_1$.

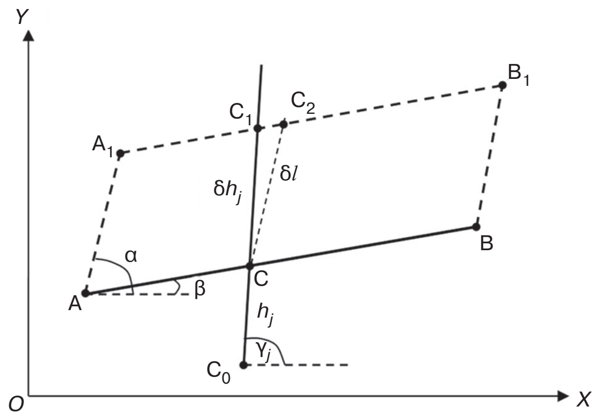

Fig. 3. Calculation of partial derivatives with changing length of the line segment

When applied to the triangle C_1CC_2, formula (11) follows from the law of sines. In this formula, $\sin(\gamma_j - \beta) \neq 0$ because the normal to the initial route at the point C and the normal to the spline, i.e., to line AB are close to each other; i.e., $\gamma_j - \beta \approx \pi/2$. At $\alpha = \beta$, the direction of the displacement coincides with the direction of the straight line; therefore, $\delta h_j = 0$. Formula (11) is also valid at $\alpha > \pi$.

If the point of intersection of the spline and the normal lies not in a straight line, but a circular arc, then β is the angle between the *OX* axis and the tangent to the circular arc at the point of its intersection with the normal, and formula (11) remains unchanged.

Let us consider the effect of the increment of the length of the circular arc on δL at unchanged values of all the other variables. In this case, the spline element from the beginning to L inclusive is not changed; in the remaining part, there is a shift by δL in the direction making angle α with the *OX* axis and a rotation by angle δα of the entire next section of the route plan. Here, α is the angle of the tangent at the end of the circular arc with the *OX* axis, while δα is its increment when the arc length is changed by δL. The center of rotation is at the end of the arc being varied.

The effect of the shift is taken into account by formula (11).

Let us now consider the rotation of the element of the route plan by the angle δα under the action of the elongation δL. Since $\delta\alpha = \delta L/R$, it is sufficient to calculate $\dfrac{\partial h_j}{\partial \alpha}$.

To calculate $\dfrac{\partial h_j}{\partial \alpha}$, it is necessary to calculate the radius of rotation S from the coordinates of the end of the arc (center of rotation: point A in Fig. 4) and the point of intersection of the spline element (the line segment or the tangent to the circle) with the normal (point C in Fig. 4).

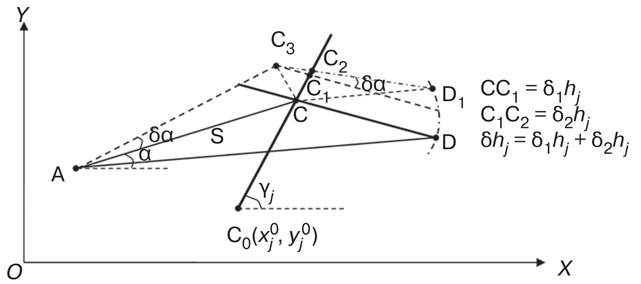

Fig. 4. Calculation of partial derivatives by rotation

The position of the straight line CD after rotation can be obtained in two ways. First, by rotating AC by the angle δα, point C_3 is found. Then, by rotating AD by the angle δα, point D_1 is determined. The intersection of the line C_3D_1 with the normal gives the sought-for $\delta h_j = CC_2$. However, CC_2 cannot be analytically expressed in terms of the known angles and coordinates.

One can make a parallel shift of CD in the direction of CC_3 (to obtain point C_1 at the intersection with the normal) followed by a rotation with the center at C_3 by the angle δα. Thereby, the same points D_1 and then C_2 are obtained.

Let us represent increment δh_j as the sum $\delta h_j = \delta_1 h_j + \delta_2 h_j$. The increment $\delta_1 h_j = CC_1$ arises by rotating point C about point A (which transforms to point C_3) followed by a parallel displacement in the direction of CC_3. Since we want to calculate partial derivatives, the lengths of the chord, arc and tangent are of the same order at small angles of rotation; therefore, we take $CC_3 = S\delta\alpha$. Here, CC_3, CC_1, and C_1C_3 are of the order δα, while the increment $\delta_2 h_j = C_1C_2$, which is caused by the rotation about the point C_3 at a first-order radius by the angle δα, has a higher order of smallness than $\delta_1 h_j$. Therefore, the point C_2 is not needed at all.

The increment $\delta_1 h_j$ is calculated from expression (11) as above by taking into account the shift by $S\delta\alpha$ in the direction along the normal to AC, which makes the angle $\alpha + \pi/2$ with the *OX* axis.

According to formula (11),

$$\delta_1 h_j = \frac{S\delta\alpha\sin(\pi/2+\alpha-\beta)}{\sin(\gamma_j-\beta)},$$

where β is the angle of the displaced straight line CD with the *OX* axis. Hence, $\frac{\delta_1 h_j}{\delta\alpha} = \frac{S\cos(\alpha-\beta)}{\sin(\gamma_j-\beta)}$.

Let x_A and y_A be the coordinates of the center of rotation, while x_C and y_C are the points of intersection with the normal. Then the derivative can be expressed as

$$\frac{\delta_1 h_j}{\delta\alpha} = \frac{(x_C-x_A)\cos\beta+(y_C-y_A)\sin\beta}{\sin(\gamma_j-\beta)}. \quad (12)$$

Here, as above, β is the angle between the *OX* axis and the tangent to the spline at the point of intersection with the *j*th normal, and γ_j is the angle of this normal with the *OX* axis.

The expression $S\cos(\alpha-\beta)$ is replaced by $(x_C-x_A)\cos\beta+(y_C-y_A)\sin\beta$.

Taking into account tangential shift (11) and rotation (12), which reduces to a shift by $S\delta\alpha$ for the sought-for derivative of the displacement h_j along the length of the circular arc L_i^{curve} (here, $\delta\alpha = \delta L_i^{\text{curve}}/R_i$, where R_i is the radius of the circular curve being varied), we obtain the formula

$$\frac{\partial h_j}{\partial L_i^{\text{curve}}} = \frac{\sin(\alpha-\beta)+\dfrac{(x_C-x_A)\cos\beta+(y_C-y_A)\sin\beta}{R_i}}{\sin(\gamma_j-\beta)}. \quad (13)$$

Formulas (11)–(13) can also be applied if the normal intersects the circular arc rather than the straight line. In this case, β is the angle between the *OX* axis and the tangent to the circle at the point of intersection.

Let us turn to the calculation of the partial derivatives of the intermediate variables with respect to the radii.

In Fig. 5, AC is the initial position of the arc, while AC_1 is the position of this arc at a changed value of the radius and constant values of the starting point $A(x_A, y_A)$, the angle α of the tangent with the *OX* axis, and the length *L* of the entire arc AC. Instead of point B in the normal, we obtain point B_1. Although the displacement along the normal is $\partial h_j = BB_1$, the new position of the point B is not B_1 (Figs. 3 and 5) because the point B leaves the normal.

Knowing the coordinates of the points $A(x_A, y_A)$ and $B(x_B, y_B)$, the angles of the tangent at these points with the *OX* axis (α and β, respectively), the length *L* of the arc AB, and the angle γ of the normal with the *OX* axis, one can calculate the derivative of the displacement along the normal $\partial h/\partial R$ (the subscripts of the normal and the curve are omitted because the point B is an arbitrary point of an arbitrary arc):

$$x_B - x_A = R(\sin\beta - \sin\alpha).$$

Here, x_A is constant, and x_B and β depend on *R*. Hence, it follows that

$$\frac{\partial x_B}{\partial R} = \sin\beta - \sin\alpha + R\cos\beta\cdot\frac{\partial\beta}{\partial R}.$$

Hereinafter, $\beta-\alpha = \frac{L}{R}$. *L* and α are fixed, while $\frac{\partial\beta}{\partial R} = -\frac{L}{R^2}$.

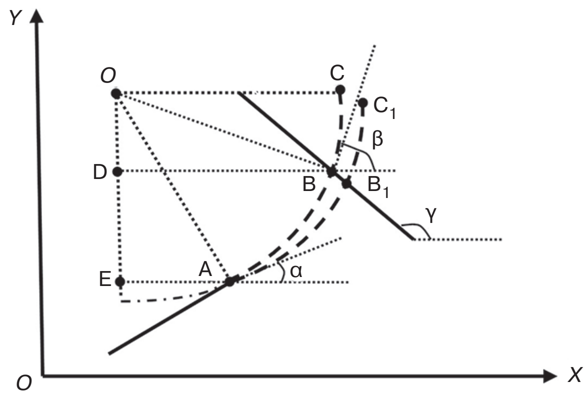

Fig. 5. Calculation of the derivatives of the displacements inside the arc with changing radius

Finally, we obtain

$$\frac{\partial x_B}{\partial R} = \sin\beta - \sin\alpha - (\beta-\alpha)\cos\beta. \quad (14)$$

Similarly, we obtain

$$\frac{\partial y_B}{\partial R} = \cos\alpha - \cos\beta - (\beta-\alpha)\sin\beta. \quad (15)$$

The increment δR of the radius gives the linear part of the increment of the coordinates of the point B:

$$\delta x_B = \frac{\partial x_B}{\partial R}\delta R \text{ and } \delta y_B = \frac{\partial y_B}{\partial R}\delta R. \quad (16)$$

The shift δh_j along the normal is obtained as a result of the shift of the tangent at point B along the *OX* axis by δx_B and then along the *OY* axis by δy_B. In the former case, the direction of the shift in formula (11) is given by the angle α = 0, and in the latter case, α = π/2.

As a result, for points inside the curve, the linear part of the normal displacement is represented by the formula

$$\delta h_j = \frac{\delta y_B \cos\beta - \delta x_B \sin\beta}{\sin(\gamma-\beta)}.$$

Using expressions (14)–(16), the derivative is expressed as

$$\frac{\delta h_j}{\delta R} = \frac{(\cos\alpha - \cos\beta - (\beta-\alpha)\sin\beta)\cos\beta}{\sin(\gamma-\beta)} - \frac{(\sin\beta - \sin\alpha - (\beta-\alpha)\cos\beta)\sin\beta}{\sin(\gamma-\beta)} = \frac{\cos(\beta-\alpha)-1}{\sin(\gamma-\beta)}. \quad (17)$$

Application of formulas (14) and (15) to the end point of the curve C gives the linear part of the increments of its coordinates:

$$\delta x_C = \left[\sin\beta - \sin\alpha - \frac{L}{R}\cos\beta\right]\delta R,$$

$$\delta y_C = \left[\cos\alpha - \cos\beta - \frac{L}{R}\sin\beta\right]\delta R.$$

Here and henceforth, α and β are the angles between the *OX* axis and the tangents at the initial and final points of the arc, respectively.

All the subsequent points of the spline are allocated the same increments (shift in the same direction). Therefore, for the linear part of the displacement along the normal for the point of intersection with the spline, we obtain

$$\delta_1 h_j = \frac{\delta y_C \cos\beta_t - \delta x_C \sin\beta_t}{\sin(\gamma_j - \beta_t)}\delta R. \quad (18)$$

Here and henceforth, β_t is the angle of the straight line or the tangent to the circle with the *OX* axis at the point of intersection of the *j*th normal, and γ_j is the angle of the normal with the *OX* axis.

Formula (18) after simplifications takes the form

$$\delta_1 h_j = \frac{\cos(\beta_t-\alpha) - \cos(\beta_t-\beta) + (\beta-\alpha)\sin(\beta_t-\beta)}{\sin(\gamma_j-\beta_t)}\delta R. \quad (19)$$

This is only a consequence of the shift with changing radius. It is also necessary to take into account the rotation of the tangent at the end of the arc (point C in Fig. 5) with changing radius; as a result, the rotation of all the next points of the spline centered at the end of the arc (point C in Fig. 5) by the angle $\delta\varphi = -\frac{L}{R^2}\delta R$, where L is the length of the arc AC.

As the rotation is taken into account with changing length of the curve (12), so the linear part $\delta_2 h_j$ of the displacement is taken into account by the shift by $S\delta\varphi$ along the normal to the straight line (or the tangent to the circle) at the point D of intersection with the normal. Here, S = CD is the radius of rotation.

According to (12), $\frac{\delta_2 h_j}{\delta\varphi} = \frac{S\cos(\varphi-\beta_t)}{\sin(\gamma_j-\beta_t)}$. Here, φ, β_t, and γ_j are the angles between the *OX* axis and the straight line CD, the straight line (or the tangent), and the intersected normal, respectively.

As a result, we obtain

$$\frac{\delta_2 h_j}{\delta\varphi} = \frac{(x_D - x_C)\cos\beta_t + (y_D - y_C)\sin\beta_t}{\sin(\gamma_j-\beta_t)}.$$

and, then,

$$\delta_2 h_j = -\frac{(x_D - x_C)\cos\beta_t + (y_D - y_C)\sin\beta_t}{\sin(\gamma_j-\beta_t)R^2}L\delta R. \quad (20)$$

Using expressions (19) and (20), the derivative of the displacement along the normal with respect to the radius is written as

$$\frac{\partial h_j}{\partial R} = \frac{\cos(\beta_t-\alpha) - \cos(\beta_t-\beta) + (\beta-\alpha)\sin(\beta_t-\beta)}{\sin(\gamma_j-\beta_t)} - \frac{(x_D - x_C)\cos\beta_t + (y_D - y_C)\sin\beta_t}{\sin(\gamma_j-\beta_t)R^2}L. \quad (21)$$

Formulas (13), (17), and (21) can be applied to any normal as well as to all spline elements preceding it. This means that it is possible to calculate the gradient of objective function (1) without having its analytical expression in terms of the main variables.

4. CONSTRUCTION AND USE OF THE MODIFIED LAGRANGE FUNCTION

Let us consider the problem of imposing constraints (7) and (8) on the intermediate variables.

Even though we do not have the expressions for the are nonlinear constraints (7) and (8) on the main variables, the penalty method can be used if the derivatives of the intermediate variables can be calculated with respect to the main variables [19, 24]. In this case, one can add a term, called a penalty function, to the objective function, with this penalty function determining a penalty for violation of constraints. In other words, instead of the original objective function, a modified Lagrange function is constructed, which takes into account both equality and nonequality constraints.

There are several versions of this method, which differ in the form of the penalty function and in methods for changing its parameters [18, 19, 23, 24].

When solving practical problems, good results were obtained using Powell's method, in which the minimum of the original objective function $F(\mathbf{x})$ under constraints $c_j(\mathbf{x}) \le 0,\ j = \overline{1,m}$, is searched for using the function

$$\Phi(\mathbf{x},\boldsymbol{\sigma},\boldsymbol{\theta}) = F(\mathbf{x}) + 1/2\sum\nolimits_1^m \sigma_j (c_j(\mathbf{x}) + \theta_j)_+^2.$$

Here, $\mathbf{x}$ is the vector of unknowns, and $F(\mathbf{x})$ is the original objective function. In the context of the present problem, the components of the vector $\mathbf{x}$ are the lengths of the spline elements, while the radii (main variables) and $c_j(\mathbf{x})$ are implicit functions of $\mathbf{x}$; however, since their dependence on the intermediate variables $\mathbf{h}$ is explicit ((7) and (8)), we can calculate their partial derivatives. Constraints (7) and (8) can always be represented as one-sided inequalities.

The setting by the user vectors σ and θ, which have m components each (according to the number of constraints), represent a set of parameters of the penalty function, with two parameters per constraint. The plus sign means that the sum includes only the terms for which $c_j(\mathbf{x}) + \theta_j > 0$. Here, $\theta_j > 0$ is the overconstraint in the jth constraint; i.e., a penalty is imposed on not only an actual violation at $c_j(\mathbf{x}) > 0$, but also at $c_j(\mathbf{x}) > -\theta_j$.

If there are equalities in the system of constraints, then the terms corresponding to them are always present in the sum. If $\theta_j = 0$ and $\sigma_j = k_n$ (k_n are set by the user), $j = \overline{1,m}$, the penalty function is simpler; however, its second derivatives with respect to x_i are discontinuous at the boundary of the admissible domain. These discontinuities increase with greater k_n, which have to be increased in each new iterative minimization cycle in order to reduce the residuals of the constraints. It is a different matter when σ_j are constant and only θ_j are varied. In this case, the surfaces of the discontinuities of the second derivatives are far from the minimum points determined when solving problems in each optimization cycle [19].

The initial values of the parameters $\theta_j > 0$ and $\sigma_j > 0$ should be selected based on the meaning and importance of the corresponding constraints and the residuals of the constraints at the initial approximation point. In our case, the solution was started with $\theta_j = 0.1$ and $\sigma_j = 1$ for all j.

Then, after solving the minimum problem $\Phi(\mathbf{x}, \boldsymbol{\sigma}, \boldsymbol{\theta})$ under simple constraints (2), (3), (5), and (6) on the main variables, constraints (7) and (8) were checked. If there were violations, then the parameters $\boldsymbol{\sigma}$ and $\boldsymbol{\theta}$ were changed according to the following rule: if there was an overconstraint in the jth constraint, i.e., if $c_j(\mathbf{x}) > -\theta_j$, then the new value $\theta^1{}_j = 0$; otherwise, $\theta^1{}_j = \theta_j + c_j(\mathbf{x})$. Such a substitution was carried out in all the constraints. To recalculate σ_j, another rule was applied: if, as a result of solving the problem, the residual of the jth constraint decreased rapidly, then σ_j was not changed; however, if the residual decreased slowly, then σ_j was increased. The following constants were used: if the residual decreased by a factor of less than 4, then the corresponding σ_j was multiplied by 10, and θ_j was divided by 2.

After the parameters were recalculated, the process was repeated; i.e., the next outer iteration was done. The calculation was terminated in the following cases:

1. A solution with acceptable residuals was obtained. In this case, one could make one more outer iteration for control to make sure that the solution remains virtually unchanged.
2. After a specified limit of outer iterations was exhausted, no solution was obtained. At the same time, there was every reason to doubt the compatibility of the system of constraints—and, consequently, the existence of a solution to the original problem. Such situations arose when specifying fixed points through which it was impossible to pass at the given minimum length values of elements and radii.

5. MAIN PROVISIONS OF THE METHOD FOR SOLVING THE PROBLEM

The initial approximation of the sought-for spline, which was obtained using dynamic programming, is used to calculate the parameters of the spline optimization problem. To do this, the following steps are performed:

1. Outer normals to the original polyline are constructed successively at the given points, and their angles with the OX axis are memorized.
2. The points of intersection of the normals with the spline are determined and memorized. At large angles of rotation, one normal can intersect the spline at two points. In this case, the point closest to the given point is selected.
3. The values of all intermediate variables are calculated.
4. At each point of intersection of the normals with the spline element (straight line or circle), the angle with the OX axis of the straight line or tangent to the circle is calculated and memorized.

The results of the calculations are sufficient to calculate the gradient of the modified Lagrange function.

Due to the simple form of the constraints on the main variables, various gradient methods can be used, including the simple coordinate-wise descent method [18, 19]. For example, when using the gradient projection method, after setting the gradient components corresponding to the variables that take limit values (the so-called active set) to zero, the standard algorithm [19] is applied. It has been experimentally established that this method does not guarantee obtaining exact solutions if penalty functions are used.

The application of optimally efficient second-order methods [19] requires the inversion of the matrix of second derivatives (Hessian matrix), which, in our case, cannot be calculated. Therefore, we used the variable metric method, namely, the so-called Davidon–Fletcher–Powell (DFP) optimization. In this method, during the course of the descent, increasingly accurate approximations of the matrices $\mathbf{H}_i$ to the inverse Hessian matrix $\mathbf{G}^{-1}$ are carried out using the gradients of the objective function at already passed points of iteration [23].

Thus [23, 24], if $\mathbf{x}_i$ be the iteration point, $\mathbf{g}_i$ be the gradient, $\mathbf{p}_i$ be the direction of descent at the ith iteration, $\mathbf{z}_i = \mathbf{x}_{i+1} - \mathbf{x}_i$, and $\mathbf{y}_i = \mathbf{g}_{i+1} - \mathbf{g}_i$, then $\mathbf{H}_0 = \mathbf{E}$ and $\mathbf{p}_i = -\mathbf{H}_i\mathbf{g}_i$. Under no constraints, we have

$$\mathbf{H}_{i+1} = \mathbf{H}_i + \mathbf{z}_i\mathbf{z}_i^{\mathrm{T}} \big/ (\mathbf{z}_i, \mathbf{y}_i) - \mathbf{H}_i\mathbf{y}_i\mathbf{y}_i^{\mathrm{T}}\mathbf{H}_i \big/ (\mathbf{H}_i\mathbf{y}_i, \mathbf{y}_i^{\mathrm{T}}). \quad (22)$$

This formula is applicable to unconstrained problems. However, in our case, constraints (2), (3), (5), and (6) on the main variables remain. If the initial approximation contains the limiting lengths or radii, then the identity matrix $\mathbf{E}$ should not be used to begin with. Instead, a projection matrix should be used, which in our cause is simply constructed as follows: in $\mathbf{E}$, 1 is replaced by 0 in the rows the numbers of which coincide with the numbers of variables that have taken limit values (active set).

When changing the set of active constraints, the matrix $\mathbf{H}_i$ should be modified [24] before calculating the direction of descent. This happens both when a constraint is included in the active set and when a constraint is excluded from the active set. The simple form of the constraints allowed the corresponding formulas [24] to be significantly simplified using the noted simple method of constructing a projection matrix.

Since DFP optimization works well (in the sense of approaching the inverse Hessian matrix) for points close to the extremum [19, 23, 24], a combination of methods was used: the simple gradient projection method (ensures a descent into the "ravine") with subsequent DFP optimization.

6. MAIN RESULTS AND OBJECTIVES OF FURTHER RESEARCH

The main result of this work is a solution to the problem of optimizing a sequence of points in a plane by a spline that is not a single-valued function. This solution is obtained not by heuristic techniques, but by mathematically correct methods (dynamic and nonlinear programming). However, due to the variety of nonlinear programming methods used, we cannot claim that the method used in the calculations is the most efficient. Of particular interest is the use of ravine algorithms [23]. It was stated [23] that the ravine conjugate gradient method, in which $\mathbf{p}_0 = -\mathbf{g}_0$, $\mathbf{p}_{i+1} = -\mathbf{g}_{i+1} + b_i\mathbf{p}_i$, and $b_i = (\mathbf{g}_{i+1} - \mathbf{g}_i, \mathbf{g}_i)/|\mathbf{p}_i, \mathbf{g}_i|$, has advantages for inaccurate one-dimensional minimization and for ravine bends.

In the context of our problem, instead of the gradient, its projection should be used; here, when the active set is changed, one should start with updating, i.e., from the step along the antigradient projection.

There are more complex algorithms than DFP, e.g., the Broyden–Fletcher–Goldfarb–Shanno method [23]. In this method, the term $\mathbf{v}_i\mathbf{v}_i^{\mathrm{T}}$ is added to formula (22) for calculating the matrix $\mathbf{H}$, where the vector $\mathbf{v}_i = (\mathbf{y}_i, \mathbf{H}_i)^{1/2}[\mathbf{z}_i/(\mathbf{z}_i, \mathbf{y}_i) - \mathbf{H}_i\mathbf{y}_i/(\mathbf{y}_i, \mathbf{H}_i\mathbf{y}_i)]$.

The efficiency of using more complex methods can only be determined experimentally.

It should be noted that gradient methods give a local minimum of the objective function. Therefore, to obtain an initial approximation in our problem, it is especially important to use dynamic programming (possibly with repetition of calculations with decreasing search increments), since dynamic programming gives a global minimum, excluding the discreteness effect.

To combat local minima, descents from different points were used. Following completion of the optimization process, the obtained solution is checked. The process starts anew with the obtained solution used as the initial approximation. All the coefficients of the modified Lagrange function take initial values. At the first iterations, the sum of their squares takes a smaller value due to the violation of constraints on the intermediate variables. However, the coefficients of the modified Lagrange function then change, resulting in the disappearance of the constraint violations. As a result, in experimental calculations, virtually the same solution was obtained. While this is not a guarantee of reaching the global minimum, it offers a real opportunity to recede from a local minimum point. To select the most efficient method of nonlinear programming for solving the problem under consideration, additional experimental studies are needed.

The main objective of further research is to generalize the results obtained for a spline with circles to the more complex problem of approximation by a spline with clothoids. First of all, it is necessary to obtain formulas for calculating derivatives in the absence of an analytical expression of the objective function in terms of the parameters of the clothoid in addition to the formulas for lines and arcs of circles that were presented in this article.

Authors' contribution. All authors equally contributed to the research work.

REFERENCES

1. Karpov D.A., Struchenkov V.I. Two-stage spline-approximation in linear structure routing. *Russ. Technol. J.* 2021;9(5):45−56 (in Russ.). https://doi.org/10.32362/2500-316X-2021-9-5-45-56
2. Karpov D.A., Struchenkov V.I. Spline approximation of multivalued function in leaner structures routing. *Russ. Technol. J.* 2022;10(4):65–74 (in Russ.). https://doi.org/10.32362/2500-316X-2022-10-4-65-74
3. Li W., Pu H., Schonfeld P., et al. A method for automatically recreating the horizontal alignment geometry of existing railways. *Comput.-Aided Civ. Inf.* 2019;34(1):71–94. https://doi.org/10.1111/mice.12392
4. Jha M.K., McCall C., Schonfeld P. Using GIS, genetic algorithms, and visualization in highway development. *Comput.-Aided Civ. Inf.* 2001;16(6):399–414. https://doi.org/10.1111/0885-9507.00242
5. Jha M.K., Schonfeld P. A highway alignment optimization model using geographic information systems. *Transportation Research Part A: Policy and Practice*. 2004;38(6):455–481. https://doi.org/10.1016/j.tra.2004.04.001
6. Jong J.C., Jha M.K., Schonfeld P. Preliminary highway design with genetic algorithms and geographic information systems. *Comput.-Aided Civ. Inf.* 2000;15(4):261–271. https://doi.org/10.1111/0885-9507.00190
7. Kang M.W., Schonfeld P., Yang N. Prescreening and repairing in a genetic algorithm for highway alignment optimization. *Comput.-Aided Civ. Inf.* 2009;24(2):109–119. https://doi.org/10.1111/j.1467-8667.2008.00574.x
8. Pushak Y., Hare W., Lucet Y. Multiple-path selection for new highway alignments using discrete algorithms. *Eur. J. Oper. Res.* 2016;248(2):415–27. https://doi.org/10.1016/j.ejor.2015.07.039
9. Sarma K.C., Adeli H. Bilevel parallel genetic algorithms for optimization of large steel structures. *Comput.-Aided Civ. Inf.* 2001;16(5):295–304. https://doi.org/10.1111/0885-9507.00234
10. Shafahi Y., Bagherian M. A customized particle swarm method to solve highway alignment optimization problem. *Comput.-Aided Civ. Inf.* 2013;28(1):52–67. https://doi.org/10.1111/j.1467-8667.2012.00769.x
11. Bosurgi G., D’Andrea A. A polynomial parametric curve (PPC-curve) for the design of horizontal geometry of highways. *Comput.-Aided Civ. Inf.* 2012;27(4):303–312. https://doi.org/10.1111/j.1467-8667.2011.00750.x
12. Cerf R. The quasispecies regime for the simple genetic algorithm with roulette wheel selection. *Adv. Appl. Probability*. 2017;49(3):903–926. https://doi.org/10.1017/apr.2017.26
13. Borodakii Yu.V., Zagrebaev A.M., Kritsyna N.A., Kulyabichev Yu.P., Shumilov Yu.Yu. *Nelineinoe programmirovanie v sovremennykh zadachakh optimizatsii* (*Nonlinear Programming in Modern Optimization Problem*). Moscow: NIYAU MEPhI; 2008. 244 p. (in Russ.).
14. Bazaraa M., Sherali Y., Shetty C. *Nonlinear programming. Theory and algorithms*. 3rd ed. Hoboken, NJ: Wiley; 2006. 872 p. ISBN 978-0-471-48600-8

СПИСОК ЛИТЕРАТУРЫ

1. Карпов Д.А., Струченков В.И. Двухэтапная сплайн-аппроксимация в компьютерном проектировании трасс линейных сооружений. *Russ. Technol. J.* 2021;9(5):45–56. https://doi.org/10.32362/2500-316X-2021-9-5-45-56
2. Карпов Д.А., Струченков В.И. Сплайн-аппроксимация многозначных функций в проектировании трасс линейных сооружений. *Russ. Technol. J.* 2022;10(4):65–74. https://doi.org/10.32362/2500-316X-2022-10-4-65-74
3. Li W., Pu H., Schonfeld P., et al. A method for automatically recreating the horizontal alignment geometry of existing railways. *Comput.-Aided Civ. Inf.* 2019;34(1):71–94. https://doi.org/10.1111/mice.12392
4. Jha M.K., McCall C., Schonfeld P. Using GIS, genetic algorithms, and visualization in highway development. *Comput.-Aided Civ. Inf.* 2001;16(6):399–414. https://doi.org/10.1111/0885-9507.00242
5. Jha M.K., Schonfeld P. A highway alignment optimization model using geographic information systems. *Transportation Research Part A: Policy and Practice*. 2004;38(6):455–481. https://doi.org/10.1016/j.tra.2004.04.001
6. Jong J.C., Jha M.K., Schonfeld P. Preliminary highway design with genetic algorithms and geographic information systems. *Comput.-Aided Civ. Inf.* 2000;15(4):261–271. https://doi.org/10.1111/0885-9507.00190
7. Kang M.W., Schonfeld P., Yang N. Prescreening and repairing in a genetic algorithm for highway alignment optimization. *Comput.-Aided Civ. Inf.* 2009;24(2):109–119. https://doi.org/10.1111/j.1467-8667.2008.00574.x
8. Pushak Y., Hare W., Lucet Y. Multiple-path selection for new highway alignments using discrete algorithms. *Eur. J. Oper. Res.* 2016;248(2):415–27. https://doi.org/10.1016/j.ejor.2015.07.039
9. Sarma K.C., Adeli H. Bilevel parallel genetic algorithms for optimization of large steel structures. *Comput.-Aided Civ. Inf.* 2001;16(5):295–304. https://doi.org/10.1111/0885-9507.00234
10. Shafahi Y., Bagherian M. A customized particle swarm method to solve highway alignment optimization problem. *Comput.-Aided Civ. Inf.* 2013;28(1):52–67. https://doi.org/10.1111/j.1467-8667.2012.00769.x
11. Bosurgi G., D’Andrea A. A polynomial parametric curve (PPC-curve) for the design of horizontal geometry of highways. *Comput.-Aided Civ. Inf.* 2012;27(4):303–312. https://doi.org/10.1111/j.1467-8667.2011.00750.x
12. Cerf R. The quasispecies regime for the simple genetic algorithm with roulette wheel selection. *Adv. Appl. Probability*. 2017;49(3):903–926. https://doi.org/10.1017/apr.2017.26
13. Бородакий Ю.В., Загребаев А.М., Крицына Н.А., Кулябичев Ю.П., Шумилов Ю.Ю. *Нелинейное программирование в современных задачах оптимизации*. М.: НИЯУ МИФИ; 2011. 244 с. ISBN 987-5-7262-1451-1
14. Bazaraa M., Sherali Y., Shetty C. *Nonlinear programming. Theory and algorithms*. 3rd ed. Hoboken, NJ: Wiley; 2006. 872 p. ISBN 978-0-471-48600-8
15. Betts J.T. *Practical methods for optimal control using nonlinear programming*. Ser. Advances in Design and Control. Philadelphia: SIAM; 2001. 190 p.

15. Betts J.T. *Practical methods for optimal control using nonlinear programming*. Ser. Advances in Design and Control. Philadelphia: SIAM; 2001. 190 p.
16. Lee J., Leyffer S. *Mixed integer nonlinear programming*. NY: Springer; 2011. 707 p. https://doi.org/10.1007/978-1-4614-1927-3
17. Sun W., Yuan Y.-X. *Optimization theory and methods. Nonlinear programming*. NY: Springer; 2006. 688 p. https://doi.org/10.1007/b106451
18. Struchenkov V.I. *Metody optimizatsii trass v SAPR lineinykh sooruzhenii* (*Methods for route optimization in CAD of linear structures*). Moscow: SOLON-Press; 2014. 271 p. (in Russ.). ISBN 978-5-91359-139-5
19. Gill F., Myurrei U., Rait M. *Prakticheskaya optimizatsiya* (*Practical Optimization*): transl. from Engl. Moscow: Mir; 1985. 509 p. (in Russ.).
[Gill Ph.E., Murray W., Wright M.H. *Practical Optimization*. London: Academic Press; 1981. 402 p.]
20. Audet C., Hare W. *Derivative-free and blackbox optimization.* Springer Series in Operations Research and Financial Engineering. Springer International Publishing; 2017. 302 p. https://doi.org/10.1007/978-3-319-68913-5
21. Kokhenderfer M.D., Uiler T.A. *Algoritmy optimizatsii* (*Algorithms for Optimization*). Moscow: Vil'yams; 2020. 528 p. (in Russ.).
[Kochenderfer M.D., Wheeler T.A. *Algorithms for Optimization*. London; The MIT Press; 2019. 520 p.]
22. Chernorutskii I.G. *Metody optimizatsii. Komp'yuternye tekhnologii* (*Methods of optimization*. *Computer technologies*). St. Petersburg: BHV-Petersburg; 2011. 329 p. (in Russ.).
23. Larichev O.I., Gorvits G.G. *Metody poiska lokal'nykh ekstremumov ovrazhnykh funktsii* (*Methods for Finding Local Extrema of Ravine Functions*). Moscow: Nauka; 1990. 96 p. (in Russ.).
24. Gill F., Myurrei U. *Chislennye metody uslovnoi optimizatsii* (*Numerical methods of conditional optimizatio*n): transl. from Engl. Moscow: Mir; 1977. 296 p. (in Russ.).
[Gill Ph.E., Murray W. *Numerical methods for constrained optimization*. London: Academic Press; 1974. 283 p.]

16. Lee J., Leyffer S. *Mixed integer nonlinear programming*. NY: Springer; 2011. 707 p. https://doi.org/10.1007/978-1-4614-1927-3
17. Sun W., Yuan Y.-X. *Optimization theory and methods. Nonlinear programming*. NY: Springer; 2006. 688 p. https://doi.org/10.1007/b106451
18. Струченков В.И. *Методы оптимизации трасс в САПР линейных сооружений*. М.: СОЛОН-Пресс; 2015. 271 с. ISBN 978-5-91359-139-5
19. Гилл Ф., Мюррей У., Райт М. *Практическая оптимизация*: пер. с англ. М.: Мир; 1985. 509 с.
20. Audet C., Hare W. *Derivative-free and blackbox optimization.* Springer Series in Operations Research and Financial Engineering. Springer; 2017. 302 p. https://doi.org/10.1007/978-3-319-68913-5
21. Кохендерфер М.Д., Уилер Т.А. *Алгоритмы оптимизации*. М.: Вильямс; 2020. 528 с.
22. Черноруцкий И.Г. *Методы оптимизации. Компьютерные технологии*. СПб.: БХВ-Петербург; 2011. 329 с.
23. Ларичев О.И., Горвиц Г.Г. *Методы поиска локальных экстремумов овражных функций*. М.: Наука; 1990. 96 с.
24. Гилл Ф., Мюррей У. *Численные методы условной оптимизации*: пер. с англ. М.: Мир; 1977. 296 с.

About the authors

Dmitry A. Karpov, Cand. Sci. (Eng.), Head of the General Informatics Department, Institute of Artificial Intelligence, MIREA – Russian Technological University (78, Vernadskogo pr., Moscow, 119454 Russia). E-mail: karpov@mirea.ru. RSCI SPIN-code 2619-7100, https://orcid.org/0000-0003-3734-7182

Valery I. Struchenkov, Dr. Sci. (Eng.), Professor, General Informatics Department, Institute of Artificial Intelligence, MIREA – Russian Technological University (78, Vernadskogo pr., Moscow, 119454 Russia). E-mail: str1942@mail.ru. RSCI SPIN-code 4581-4698, https://orcid.org/0000-0002-9801-745

Об авторах

Карпов Дмитрий Анатольевич, к.т.н., заведующий кафедрой общей информатики Института искусственного интеллекта, ФГБОУ ВО «МИРЭА – Российский технологический университет» (119454, Россия, Москва, пр-т Вернадского, д. 78). E-mail: karpov@mirea.ru. SPIN-код РИНЦ 2619-7100, https://orcid.org/0000-0003-3734-7182

Струченков Валерий Иванович, д.т.н., профессор, профессор кафедры общей информатики Института искусственного интеллекта, ФГБОУ ВО «МИРЭА – Российский технологический университет» (119454, Россия, Москва, пр-т Вернадского, д. 78). E-mail: str1942@mail.ru. SPIN-код РИНЦ 4581-4698, https://orcid.org/0000-0002-9801-7454

Translated from Russian into English by Vladislav V. Glyanchenko
Edited for English language and spelling by Thomas A. Beavitt