Mathematical modeling

Математическое моделирование

UDC 519.857
https://doi.org/10.32362/2500-316X-2021-9-5-45-56

RESEARCH ARTICLE

Two-stage spline-approximation in linear structure routing

Dmitry A. Karpov,
Valery I. Struchenkov @

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

Abstract. In the article, computer design of routes of linear structures is considered as a spline approximation problem. A fundamental feature of the corresponding design tasks is that the plan and longitudinal profile of the route consist of elements of a given type. Depending on the type of linear structure, line segments, arcs of circles, parabolas of the second degree, clothoids, etc. are used. In any case, the design result is a curve consisting of the required sequence of elements of a given type. At the points of conjugation, the elements have a common tangent, and in the most difficult case, a common curvature. Such curves are usually called splines. In contrast to other applications of splines in the design of routes of linear structures, it is necessary to take into account numerous restrictions on the parameters of spline elements arising from the need to comply with technical standards in order to ensure the normal operation of the future structure. Technical constraints are formalized as a system of inequalities. The main distinguishing feature of the considered design problems is that the number of elements of the required spline is usually unknown and must be determined in the process of solving the problem. This circumstance fundamentally complicates the problem and does not allow using mathematical models and nonlinear programming algorithms to solve it, since the dimension of the problem is unknown. The article proposes a two-stage scheme for spline approximation of a plane curve. The curve is given by a sequence of points, and the number of spline elements is unknown. At the first stage, the number of spline elements and an approximate solution to the approximation problem are determined. The method of dynamic programming with minimization of the sum of squares of deviations at the initial points is used. At the second stage, the parameters of the spline element are optimized. The algorithms of nonlinear programming are used. They were developed taking into account the peculiarities of the system of constraints. Moreover, at each iteration of the optimization process for the corresponding set of active constraints, a basis is constructed in the null space of the constraint matrix and in the subspace – its complement. This makes it possible to find the direction of descent and solve the problem of excluding constraints from the active set without solving systems of linear equations. As an objective function, along with the traditionally used sum of squares of the deviations of the initial points from the spline, the article proposes other functions taking into account the specificity of a particular project task.

Keywords: route, horizontal and vertical alignment, spline, dymanic programming, objective function, restrictions

• **Submitted:** 25.01.2021 • **Revised:** 14.03.2021 • **Accepted:** 12.07.2021

For citation: Karpov D.A., Struchenkov V.I. Two-stage spline-approximation in linear structure routing. *Russ. Technol. J.* 2021;9(5):45–56. https://doi.org/10.32362/2500-316X-2021-9-5-45-56

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

Двухэтапная сплайн-аппроксимация в компьютерном проектировании трасс линейных сооружений

Д.А. Карпов,
В.И. Струченков @

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

Резюме. В статье компьютерное проектирование трасс линейных сооружений рассматривается как задача сплайн-аппроксимации. Принципиальной особенностью соответствующих проектных задач является то, что план и продольный профиль трассы состоят из элементов заданного вида. В зависимости от типа линейного сооружения используются отрезки прямых, дуги окружностей, парабол второй степени, клотоид и др. В любом случае результатом проектирования является кривая, состоящая из нужной последовательности элементов заданного вида. В точках сопряжения элементы, как правило, имеют общую касательную, а в наиболее сложном случае – и общую кривизну. Подобные кривые принято называть сплайнами. В отличие от других применений сплайнов в проектировании трасс линейных сооружений приходится учитывать многочисленные ограничения на параметры элементов сплайна, возникающие из необходимости соблюдения технических нормативов с целью обеспечения нормальной эксплуатации будущего сооружения. Технические ограничения формализуются в виде системы неравенств. Главная отличительная особенность рассматриваемых проектных задач состоит в том, что число элементов искомого сплайна неизвестно и должно быть определено в процессе решения задачи. Это обстоятельство принципиально усложняет задачу и не позволяет применить для ее решения математические модели и алгоритмы нелинейного программирования, так как неизвестна размерность задачи. В статье предлагается двухэтапная схема сплайн-аппроксимации плоской кривой, заданной последовательностью точек, при неизвестном числе элементов сплайна и наличии ограничений на параметры его элементов. На первом этапе определяется число элементов сплайна и приближенное решение задачи аппроксимации. Используется метод динамического программирования. На втором этапе выполняется оптимизация параметров элементов сплайна. Используются алгоритмы нелинейного программирования, разработанные с учетом особенностей системы ограничений. При этом на каждой итерации процесса оптимизации для соответствующего набора активных ограничений строится базис в нуль-пространстве матрицы ограничений. Это позволяет найти направление спуска и решить вопрос об исключении ограничений из активного набора без решения систем линейных уравнений вообще, а в наиболее сложных случаях – решая линейные системы малой размерности. В качестве целевой функции наряду с традиционно используемой суммой квадратов отклонений аппроксимируемых точек от сплайна в статье предлагаются другие функции с учетом специфики конкретной проектной задачи.

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

• **Поступила:** 25.01.2021 • **Доработана:** 14.03.2021 • **Принята к опубликованию:** 12.07.2021

Для цитирования: 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

Прозрачность финансовой деятельности: Никто из авторов не имеет финансовой заинтересованности в представленных материалах или методах.

Авторы заявляют об отсутствии конфликта интересов.

INTRODUCTION

A linear structure is a structure the ground position of which is determined by the axis of the structure, which is called the route. Among linear structures are roads and railways, pipelines of various purposes, channels, water conduits, etc. A route is a three-dimensional curve, which is conventionally represented by two plane curves: the plan and the longitudinal profile.

The plan of a route is its projection on the *XOY* plane, and the longitudinal profile is the graph of the function $Z(s)$, where s is the length of the curve in plan as calculated from a given initial point. The longitudinal profile is a developed view of the vertical surface passing through the route.

Design of the longitudinal profile of a structure of any type can be considered as the construction of a spline consisting of elements of a given shape. This spline should have the minimum (in a given meaning) deviation from the initial broken line, which is the ground profile in the case of design of new structures and is the profile of the existing structure in the case of design a reconstruction.

The simplest spline of the first order is the grade line of the longitudinal profile of a railway. In this case, the problem is to convert the initial broken line (ground profile) to another broken line that satisfies a variety of constraints: on the slopes of elements and the differences of the slopes of neighboring elements, on the minimum length of elements, and on the height at some points and in some zones [1, 2]. Because the design slopes are small, the length of an element and the difference of the abscissas of its ends virtually coincide; therefore, the difference of the slopes of neighboring elements is equated with the angle of rotation, and the slope is identified with the angle between the element and the abscissa axis.

Meanwhile, the number of elements of the sought spline is unknown. This fact and also numerous constraints distinguish significantly the considered design problem of spline approximation from problems solved in spline theory and its applications [3–5], where the number of spline knots and their abscissas are considered to be given, and constraints are typically absent.

In a simplified formulation, the problem of seeking the optimal spline as a broken line at an unknown number of elements under constraints was solved in the last century as applied to the design of the longitudinal profile of new railways [6, 7].

The problem was solved in two stages. At the first stage, the initial ground profile was converted to a broken line comprising short elements under all the constraints, except the constraint on the length of an element. The developers of the first designing algorithms called such a profile the chain [6].

At the second stage, the chain was converted to the grade line under all the constraints, including those on the length of elements.

In a realistic formulation as applied to design under rugged terrain and complex geology conditions, the problem was solved on a BESM-4 computer by nonlinear programming. The corresponding program gained a wide practical use despite a long computational time because of the extremely low computational speed of this and subsequent computer models (Minsk 32, ES 1020, and others) of the last century [1].

In CAD systems, highly popular in Russia, which were developed by international companies [8–10], and their Russian [11] and Belarusian [12] analogs, the computer is used to solve auxiliary problems, rather than to elaborate the optimal design solutions. In these systems, a spline approximation problem is solved "through the eyes"; i.e., the designer should specify some information that completely determines a sought line. At best, he or she considers several of the theoretically infinitely many possible solutions.

At the present time, the mathematical model, algorithm, and previously developed designing programs should be improved because of changes in the technical specifications for design of high-speed railways. The first order spline should be replaced by a spline comprising line segments and circular arcs, the number of which remains unknown.

A similar spline is also used in designing big-inch pipelines.

In designing the longitudinal profile of roads, a problem arises to seek a parabolic spline of the second order [13] with the above specific features. This problem was solved by nonlinear programming [13].

A spline with circular arcs is used as an alternative to a spline with parabolas in designing the longitudinal profile of roads and also the plane of the routes of various linear structures [14].

The study aimed to analyze the above design problems from a single theoretical standpoint as problems of spline approximation and to present the key stages and specific features of their solution algorithms.

1. FORMULATION OF A PROBLEM OF CIRCULAR ARC SPLINE APPROXIMATION AND ITS FORMALIZATION

Let us consider a problem of designing a longitudinal profile using straight-line elements conjugated to circular arcs.

In the case of redesigning, the initial profile is the profile of the existing structure. If a new structure is designed, the initial profile is the ground profile. The chain longitudinal profile (Fig. 1, dashed line), which can be constructed using the existing designing programs [14], is used to find the number of spline elements. The lengths of elements of the chain need not be equal, but the abscissas of its knots and the abscissas of the knots of the initial broken line coincide.

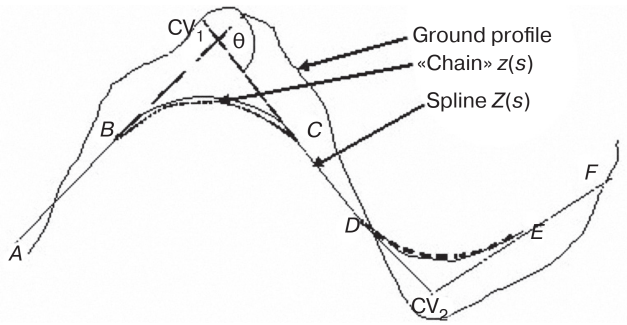

Fig. 1. Spline with circular arcs

Thus, we have the broken line $z(s)$, which should be converted with minimum deviations to the spline $Z(s)$ consisting of line segments conjugated to circular arcs (Fig. 1).

There are the following constraints:

(1) on the slopes I_j of the straight-line spline elements: $-I_{\min} \le I_j \le I_{\max}$, $j = 1, 2, \ldots, N-1$, where N is the number of spline knots (vertices of the sought broken line, hereinafter referred to as control vertices (CVs)). Actually, this is a constraint on the first derivative of the function $Z(s)$;

(2) on the radii (curvature) of the convex and concave inscribed curves: $1/R_{\text{convex}} \le 1/R_j \le 1/R_{\text{concave}}$, $j = 1, 2, \ldots, N$, $R_{\text{convex}} < 0$, and $R_{\text{concave}} > 0$;

(3) on the lengths of the circular arcs (BC, DE in Fig. 1): $L_{\text{arc},j} \ge L_{\text{arc,min}}$;

(4) on the lengths of the straight-line inserts between the curves (CD in Fig. 1): $L_{\text{ins},j} \ge L_{\text{ins,min}}$.

Additional constraints can be imposed on the ordinates of some points (height constraints at points of intersection of water conduits, other communications, and so on).

Objective function

Equal deviations in different directions from the initial line can often be nonequivalent. Therefore, the conventional minimization of the sum of the squared deviations at given points (also with different weights) is inappropriate.

In designing new roads, the total cut-and-fill quantity can be taken to be the objective function at this stage. The construction costs can be taken to be the objective function if the cut and fill does not give rise to a relationship between elements, which arises if the earth removed from cuts is used to construct fills and requires one to consider the grade line as a whole [14] as in nonlinear programming. At the stage of the conversion of the initial broken line (the chain or the existing profile) to a spline of a necessary type, the ordinate deviations (working marks) are small (about 0.5 m [14]), which allows one to use simplified efficiency criteria because the purpose of this stage is to determine the number of elements and their approximate positions, i.e., construct the initial approximation for nonlinear programming.

In redesigning of the longitudinal profile of a road, at this stage, it is expedient to use modeling functions, which take into account specific features of a problem.

For example, in designing the longitudinal profile during redesigning railways by straight-line elements without taking into account circular curves, which were inscribed into the found line, smooth modeling function $F(h)$ (a spline of the second order with the defect 1) was successfully used (Fig. 2). Here, h is the working mark, i.e., the difference of the ordinates of the sought and initial splines: $h(s) = Z(s) - z(s)$. The h_0 and Δ values and the parameters of the elements of $F(h)$ were found from the existing and designed depths of ballast (H_{ex} and H_{des}, respectively), and the rail and tie heights.

$\Delta = \max(0, H_{\text{ex}} - H_{\text{des}})$, and the portions of the graph of $F(h)$ represent (*1*) filling up of ballast, (*2*) cutting of ballast, and (3) cutting of roadbed.

At $\Delta = 0$, portion *2* of the graph of $F(h)$ is absent. If the existing and designed heights of rails and toes are equal, we have $h_0 = 0$.

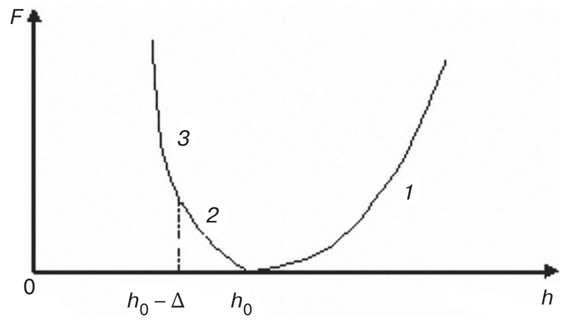

Fig. 2. Modeling function

The $F(h_i)$ values were calculated at the knots of the initial spline, and the objective function had the form

$$\min \Phi(\mathbf{h}) = \sum_{1}^{k} v_i F(h_i), \qquad (1)$$

where the coefficients v_i are equal to the half-sums of the lengths of its adjacent elements. Similar modeling functions were used in redesigning the longitudinal profile of roads using parabolic splines [13].

If the objective function is the cut-and-fill quantity, then $F(h_i)$—the cross-sectional area at the ith point—remains piecewise quadratic and corresponds to the calculation of the volume as an integral using the trapezoidal rule.

2. SPLINE APPROXIMATION BY DYNAMIC PROGRAMMING

Dynamic programming under a number of conditions [15–20] makes it possible to create algorithms of several-step construction of the optimal route of motion of a certain system from a given initial state to a final state by solving same-type problems at each of the steps, which are simpler than the initial problem [21]. Variants of reaching one and the same state by various ways are considered to be comparable, and in each state, only the best (according to a chosen criterion) variant remains.

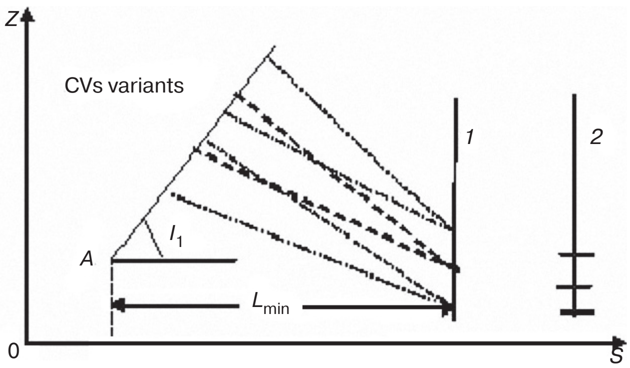

Fig. 3. First step of algorithm

The key concept of dynamic programming is the state of a system, which, for our problem, we define as a pair of two items: one is a point in a straight-line element of a spline, such that beginning with this point a circular arc can be constructed or a straight line can be continued, and the other is the angle between this element and the *OS* (abscissa) axis. The initial state (point *A* and slope I_1 in Fig. 3) is considered to be given.

With respect to the knots of the initial spline (the broken line $z(s)$), a variation grid at a given step is constructed (points in verticals *1* and *2* in Fig. 3). At each of these points, proceeding from the slope of the initial spline, angles with the *OS* axis (search sector) are assigned. The step of assignment of points and angles in the verticals, and the numbers of steps in the vertical and angular directions at each of the points are the initial parameters of an algorithm of seeking the design spline and are specified before calculation. If necessary, the calculation can initially be performed at large increments and then, using the obtained solution, at smaller increments. This is a common trick to reduce the computational time in dynamic programming, which was successfully used in parabolic spline approximation [13, 14].

The first vertical was chosen to be the one spaced apart from the initial point *A* (in abscissas) at distance L_{min}, which is given under constraints 3 and 4 on the lengths of curves and straight-line inserts. In addition, distance L_{max} is given as the sum of the maximum length of the curve and the length of the straight-line insert. Dynamic programming is performed using the angles of rotation (Fig. 1) and the coordinates of their vertices.

In seeking the first curve of the design spline, the left-hand side of the angle is given (the point *A* and the angle I_1). The points in each vertical within the range from L_{min} to L_{max} together with the directions assigned at each of the points constitute the possible variants of the right-hand side of the first angle of rotation (Fig. 3) and determine the corresponding variants of the first CV. For each variant, using the minimum radius of a convex or concave curve (depending on the sign of the angle of rotation), the possibility of satisfying all the constraints is analyzed, and only the variants for which all the constraints are met are retained. Further, it is considered whether or not the radius of the inscribed arc can be increased without violation of constraints using the difference of the slopes of the adjacent elements of the initial spline that are within the angle under consideration. The radius is chosen such that the value of the objective function for the corresponding CV is minimum.

At the first step of comparing paths and rejecting variants, one and the same state is not reached. Each of the allowed states of the first step together with the corresponding values of the objective function (the cost to reach the initial state), the CV coordinates, the radius, and the angle of rotation are stored in memory.

General step of algorithm

The knots of the initial spline are considered such that the abscissas (and the corresponding verticals) of which are within the range from $S_A + 2L_{min}$ to $S - L_{min}$, where *S* is the abscissa of the end of the profile. For each vertical, a sequential analysis is made of all the preceding knots that are no less than L_{min} and no more than L_{max} apart from the vertical and of the straight lines passing through them. For each intersection, the same operations as at the first step are sequentially performed. The difference is that, in the considered state (the right-hand side of the angle, line *BC* in Fig. 4), there may be many intersections with the sides of the preceding angles that originate from one or different CVs (the points A_i and A_{i+1} in Fig. 4). As at the first step, only the joints that satisfy the constraints are considered and compared. As a result, each state in each vertical (the point *C* and the angle) is reached by either one variant, or none. For each of such variants, additionally stored in memory are the point and the direction (the point A_i and the angle A_iB with the abscissa axis in Fig. 4) corresponding to the left-hand side of the angle; i.e., for each new state, the relationship with the best of the preceding states is memorized.

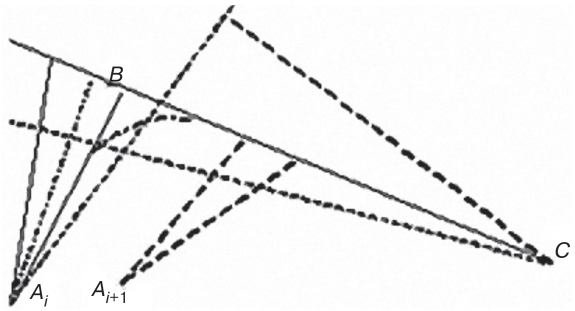

Fig. 4. Comparison and rejection of variants

Last step of algorithm

At the last step, the right-hand side of the angle of rotation is known. These are the final point and the angle determining the final direction. The same operations as at the general step are carried out, and a comparison of the allowable joints determines the minimum value of the objective function. The optimal spline and its parameters are restored by a turn over memorized ties, which is typical for dynamic programming [21, 22].

Of course, one can also consider several final directions and points and perform the same operations for each of them with subsequent selection of the best variant.

The use of the algorithm encountered difficulties in handling long line segments. This gave rise to small angles of rotation. Depending on a specific problem, either such angles are not allowed at all, or curves are not inscribed in them (e.g., in designing low-type roads). In the former case, instead of two intersecting straight lines, one straight line can be formed (using the terminal points) in the course of the exhaustion of variants. But because the algorithm is intended only for the construction of the initial assumption, such transformations were made only for the obtained spline to avoid excessive complications. This is also justified by the fact that not nearly all such intersections at small angles are contained in the final solution.

3. OPTIMIZATION OF SPLINE PARAMETERS

The design line is completely determined by the coordinates of the vertices of the angles of rotation and the radii of the inscribed circles, which are found by dynamic programming (Fig. 5).

To start with, we consider the CV abscissas constant, i.e., analyze the possibility of optimization of the position of the spline by moving CVs along fixed verticals. Because the slopes are small (no more than several tens of permille), the lengths of the sides of each angle are considered to be equal to the difference of the CV abscissas, which are invariable. Since the initial and final points and the directions at them are given, the ordinates of the first and last CVs cannot change. Therefore, the variables are only Z_j, $j = 1, 2, \ldots, n$, the ordinates of the CVs being varied (their number is $n = N - 2$), and the radii R_j of the inscribed curves. The given boundary conditions are taken into account by the calculation of the limiting values of the slopes, I_1 and I_n, and then the ordinates, Z_1 and Z_n [14].

To obtain a nonlinear programming problem with objective function $\Phi(\mathbf{h})$ (1), one should express in terms of these variables the working marks at the knots of the initial broken line, i.e., the difference of the ordinates of the design spline and the initial broken line ($B'B''$ in Fig. 5), and all the constraints.

In designing the longitudinal profile of new roads, the objective function corresponds to the minimum cost of construction of subgrade and artificial structures. The corresponding models are the same as in the case of using parabolic splines in codesigning the longitudinal and transverse profiles with taking into account the earth mass distribution [14].

If there are such expressions, the calculation of the gradient of objective function (1) reduces to the simple recalculation of derivatives [14] because the ordinates of the points (D and B in Fig. 5) in straight-line elements depend linearly on the CV ordinates. Because the slopes are small, the angle of rotation is considered to be equal to the difference of the adjacent slopes (ΔI_j in Fig. 5).

This enables one to express, with sufficient accuracy, the deviations of the points of the curve from the corresponding points of the straight lines (CC'' and BB'' in Fig. 5), i.e., the corrections to the working marks calculated from the sides of the angle or rotation ("of a boom").

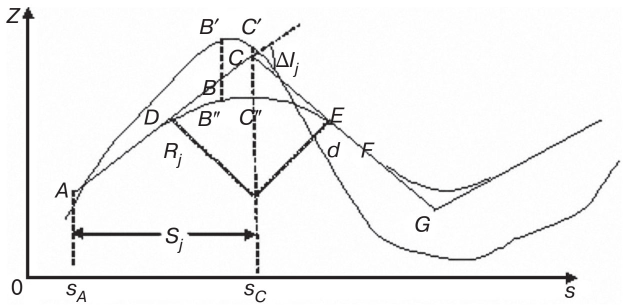

Fig. 5. To the recalculation of derivatives in the presence of circular arcs

In Fig. 5, $CC'' = \delta_j = R_j\Delta I_j^2/8$; $BB'' = \delta_B = \delta_j - t_B\Delta I_j/2 + t_B^2/(2R_j)$, where $t_B = |s_C - s_B|$ is the difference of the abscissas of the CV and the point in the curve; $\Delta I_j = I_{j+1} - I_j$, where $I_j = (Z_j - Z_{j-1})/S_j$, Z_j are unknown design marks of the vertices of the angles of rotation, and S_j are the differences of the abscissas, which differ insignificantly from the lengths of the sides of the angles; and $s_C - s_A = S_j \approx AC$.

Instead of the constraints on the difference of the slopes, there are constraints on the minimum lengths of the curves, $R_j\Delta I_j \ge L_{j,\min}$, and on the minimum length of the straight-line insert, i.e., the sum $CE + FG$ (Fig. 5), should meet the condition

$$R_j\Delta I_j/2 + R_{j+1}\Delta I_{j+1}/2 + L_{\text{ins, min}} \le S_{j+1}, j = 1, 2, ..., n. \quad (2)$$

Here, $L_{\text{ins,min}}$ is a given minimum length of the straight-line inset, and n is the number of CVs.

At small ΔI_j, to change the length of the straight-line insert by 10 m, it is required to change the radius by 1000 m and more, which can hardly be done by optimizing a spline constructed by dynamic programming. Therefore, condition (2) can be simplified by eliminating the relationship between the variables for the adjacent CVs using the spline obtained at the first stage as the initial approximation.

This can be done by making the following operations:

1. Calculate all the $T_j = R_j\Delta I_j/2$ (in design practice, they are called tangents).
2. Calculate all the straight-line inserts $d_j = S_j - (T_{j-1} + T_j)$, $j = 2, \ldots, n$, and $c_j = d_j - L_{\text{ins,min}}$ ("store").
3. If $d_j = L_{\text{ins,min}}$, then T_{j-1} and T_j are fixed as the maximum values of $R_{j-1}\Delta I_{j-1}/2$ and $R_j\Delta I_j/2$. The fixed values are not further changed.
4. Sequentially consider the straight-line inserts in ascending order, beginning with the smallest d_k. The values $T_{k-1} + T_k$ can be increased by $c_k = d_k - L_{\text{ins,min}}$ without risking a violation of the constraint on the straight-line insert at the neighboring elements. If the maximum values of $R_{k-1}\Delta I_{k-1}/2$ and $R_k\Delta I_k/2$ are not yet fixed, then $T_{k-1} + c_k/2$ and $T_k + c_k/2$, respectively, are taken as their maximum values. The values c_{k-1} and c_{k+1} are decreased by $c_k/2$. If the value T_{k-1} is fixed, then $\max(R_k\Delta I_k/2) = T_k + c_k$ and c_{k+1} are decreased by c_k. If the value T_k is fixed, then $\max(R_{k-1}\Delta I_{k-1}/2) = T_{k-1} + c_k$.
5. Let us proceed to step 3 and continue the process until there are unfixed maximum values of $R_j\Delta I_j/2$. If necessary, the positions of the initial and final points of the profile are taken into account, and the maximum values of $R_1\Delta I_1/2$ and $R_n\Delta I_n/2$ are corrected (decreased).

Bearing in mind that $R_j\Delta I_j$ is the length of the jth curve and $L_{\text{cur,min}}$ is its minimum value, and denoting the calculated maximum values of $R_j\Delta I_j$ as $L_{j,\max}$, we obtain the system of two-sided inequalities

$$L_{\text{cur,min}} \le R_j\Delta I_j \le L_{j\ \max}, j = 1, 2, ..., n.$$

Let us convert this system of nonlinear inequalities to a linear system by change of variables from radii to curvatures $\sigma_j = 1/R_j$. The constraint on $L_{j,\max}$ is $\Delta I_j \le L_{j,\max}\sigma_j$ at $R_j > 0$ and $L_{j,\max}\sigma_j \le \Delta I_j$ at $R_j < 0$. The constraint on $L_{\text{cur,min}}$ is $L_{\text{cur,min}}\sigma_j \le \Delta I_j$ at $R_j > 0$ and $\Delta I_j \le L_{\text{cur,min}}\sigma_j$ at $R_j < 0$.

The signs of R_j are known; hence, we have the linear system of the form

$$\alpha_j\sigma_j \le \Delta I_j \le \sigma_j\beta_j, j = 1, 2, ..., n. \quad (3)$$

At $R_j > 0$, $\beta_j = L_{j,\max}$ and $\alpha_j = L_{\text{cur,min}}$. At $R_j < 0$, conversely, $\beta_j = L_{\text{cur,min}}$ and $\alpha_j = L_{j,\max}$.

The algorithm of solving the nonlinear programming problem of finding $\min\Phi(\mathbf{x})$, where $\mathbf{x}$ is the vector of unknowns and $\Phi(\mathbf{x})$ is the objective function, under linear constraints $\mathbf{Ax} \le \mathbf{b}$ consists of the following steps:

1. Construction of an allowable initial approximation.
2. Calculation of antigradient **f**.
3. Construction of active constraint matrix $\mathbf{A}_k$ and descent direction **p**.
4. Check of conditions of termination of calculation.

If the length of the descent vector exceeds given ε, then go to step 5, else check the possibility of eliminating constraints from the active set. If there are no such constraints, then the process is over, else exclude one of the constraints and go to step 3.

5. Search for a step in the direction of the descent as the minimum of the steps to the boundary and to the minimum point. In this case, a one-dimensional minimum search problem is solved.
6. Transition to a new point. Further, if the antigradient at the new point has already been calculated in the search for a step, then go to step 3, else go to step 2.

In the general case, the algorithm ensures a hit of the vicinity of the local minimum point. Therefore, it is important to obtain a good initial approximation by dynamic programming.

There are two key steps: the construction of the descent direction and the elimination of constraints from the active set [22–24]. The problem can be solved using standard algorithms, which require solving systems of linear equations (matrix inversion) at each iteration. For example, the projection of the gradient at the kth iteration can be calculated from the Rosen formula:

$$\mathbf{p} = (\mathbf{E} - \mathbf{A_k}(\mathbf{A_k}\mathbf{A_k^T})^{-1}\mathbf{A_k})\mathbf{f}.$$

To solve the question of the elimination of constraints from the active set, the vector $\mathbf{u} = (\mathbf{A_k}\mathbf{A_k^T})^{-1}\mathbf{A_k}\mathbf{f}$, should be calculated, for which the matrix $\mathbf{A_k}\mathbf{A_k^T}$. should be inverted.

Instead of this, let us consider the possibility of constructing the descent direction using the simple structure of the system of constraints [25]. For this purpose, it is necessary to be capable of constructing a basis in the null space of the matrix $\mathbf{A}_k$ for any active set, which was implemented in the program of spline optimization as a broken line without inscribed curves [14].

For example, if the basis matrix **C** has already been constructed, then the descent vector has the form $\mathbf{p} = \mathbf{C}\mathbf{C}^{\mathbf{T}}\mathbf{f}$, where **f** is the antigradient.

Constraints (3) contain additional variables σ_j, but the previously constructed basis vectors [25] can also be converted for this system.

If, in our problem, a certain variable z_j is contained in none of the active constraints, then $p_j = f_j$. The presence of such free points enables one to divide the profile into legs of independent construction of basis vectors and the corresponding components of the descent vector. For example, for system (3) of active constraints on the straight-line insert in the range of CVs from the $(m + 1)$th to the $(m + r - 1)$th (Fig. 6), the variables are z_{m-1}, z_m, …, z_{m+r-1}, z_{m+r} and σ_m, σ_{m+1}, …, σ_{m+r-1}; and the free variables are z_{m-2} and z_{m+r+1}.

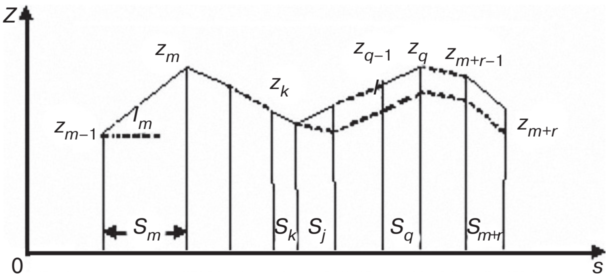

Fig. 6. Example of construction of basis vectors

The active constraints are the following:

$$\begin{aligned}
&-\Delta I_m + \alpha_m \sigma_m \le 0,\\
&-\Delta I_{m+1} - \beta_{m+1}\sigma_{m+1} \le 0,\\
&\ldots\\
&-\Delta I_{m+r-1} - \alpha_{m+r-1}\sigma_{m+r-1} \le 0.
\end{aligned} \tag{4}$$

This system in the variable ordinates has the form:

$$\begin{aligned}
&-1/S_m\, z_{m-1} + (1/S_m + 1/S_{m+1})\, z_m -\\
&-1/S_{m+1}\, z_{m+1} + \alpha_m \sigma_m \le 0,\\
&-1/S_{m+1}\, z_m + (1/S_{m+1} + 1/S_{m+2})\, z_{m+1} -\\
&-1/S_{m+2}\, z_{m+2} - \beta_{m+1}\sigma_{m+1} \le 0,\\
&\ldots\\
&-1/S_{m+r-1}\, z_{m+r-2} + (1/S_{m+r-1} + 1/S_{m+r})\times\\
&\times z_{m+r-1} - 1/S_{m+r}\, z_{m+r} - \alpha_{m+r-1}\sigma_{m+r-1} \le 0.
\end{aligned} \tag{5}$$

The sought basis vectors should convert the inequalities of this system to equalities and be linearly independent. For example, the vector $\mathbf{c_1} = (1\ 1\ \ldots\ 1\ 1\ 0\ 0\ 0)^{\mathrm{T}}$ ($r + 2$ units and r zeros) shifts all the CVs along the ordinate axis without changing slopes and radii. Obviously, the difference of the adjacent slopes and the curvature also remain unchanged.

If all slopes are increased equally, e.g., by 1 (i.e., if a rotation with the center at the $(m - 1)$th CV is made) without changing radii, then the constraints of system (4) and its corresponding system (5) remain active. Therefore, the vector $\mathbf{c_2} = (0\ \ S_m\ \ S_m + S_{m+1}\ \ S_m + S_{m+1} + S_{m+2}\ \ldots\ S_m + S_{m+1} + S_{m+2} + S_{m+r}\ \ 0\ \ldots\ 0)^{\mathrm{T}}$ can also be included in the sought basis. Another r basis vectors are obtained by making rotations about the mth, $(m + 1)$th, …, and $(m + r - 1)$th CVs chosen sequentially as the centers of rotation, changing the right-hand slopes by 1, and compensating the change in the difference of the slopes at the center of rotation by changing the corresponding curvature: (6).

The linear independence of the obtained vectors follows from their construction method.

If, in such a leg, a certain curve σ_j takes the limiting value, then the corresponding component of the descent vector is zero, σ_j is excluded from the variables taken into account in constructing the basis, and the vector corresponding to the change in this variable is not included in the basis.

If the limiting value is taken by the slope of a certain element I_k, then the vector $\mathbf{c}_1$ remains in the basis, but the vectors corresponding to the rotations about the centers at CV_j ($j = m - 1, m, \ldots, k - 1$) change this slope and are not included in the basis.

New basis vectors are constructed by searching through the CVs, beginning with the $(k - 1)$th to the mth, if $k > m$. The center of rotation is taken to be the $(k - 1)$th CV, but the left-hand part is rotated, so that all the left-hand slopes gain equal increments. The difference of the slopes changes only at the $(k - 1)$th CV by 1, which is compensated by changing the $(k - 1)$th curvature. The basis vector is obtained:

$$\mathbf{c} = (s_{k-1} + s_{k-2} + \ldots + s_m\quad s_{k-1} + s_{k-2} + \ldots + s_{m+1}\ \ldots\ s_{k-1} + s_{k-2}\quad s_{k-1}\quad 0 \ldots 0\ \ 0 \ldots 1/\delta \ldots 0 \ldots 0)^{\mathrm{T}},$$

where $\delta = \alpha_{k-1}$ or $\delta = \beta_{k-1}$, depending on the sign of σ_{k-1}.

$$\begin{aligned}
&\mathbf{c_3} = (0\ 0\ S_{m+1}\ \ S_{m+1} + S_{m+2}\ \ldots S_{m+1} + S_{m+2} + \ldots + S_{m+r}\qquad 1/\alpha_m\ \ 0\qquad \ldots\ 0)^{\mathrm{T}},\\
&\mathbf{c_4} = (0\ 0\ 0\qquad S_{m+2}\qquad \ldots S_{m+2} + S_{m+3}\ \ \ldots\ \ S_{m+2} + S_{m+3} + \ldots + S_{m+r}\ \ 0\qquad 1/\beta_{m+1} \ldots 0)^{\mathrm{T}},\\
&\ldots\\
&\mathbf{c_{r+2}} = (0\ 0\ 0\quad 0\quad \ldots\quad 0\quad \ldots\qquad S_{m+r}\quad 0\ \ldots\ 0\ 1/\alpha_{m+r-1})^{\mathrm{T}}.
\end{aligned} \tag{6}$$

If the limiting slope is not the last one, then vertices k, $k + 1$, …, $m + r - 1$ are sequentially taken to be the centers of rotation (Fig. 6), the right-hand part is rotated, and the corresponding basis vector is constructed with the compensation of the change in the difference of the slopes at the center of rotation.

If the limiting slope is the slope of the initial element, then only the rotation of the right-hand part and the motion only to the right are considered. Similarly, if the limiting slope is the slope of the last element, then only left-hand part of the profile is rotated, and the CVs are tested only to the left.

If the limiting value is also taken by the qth slope ($q > k + 1$), then the basis vectors are constructed to the left of the ($k - 1$)th CV and to the right of the qth CV, as for the only limiting slope. If $q = k + 1$, then it is sufficient, else the basis vectors should also be constructed for $k < j < q$. For this purpose, sequentially, beginning with $j = k + 1$ and to $j = q - 1$, all the components of the basis vector are $\mathbf{c}_i = 0$ at $i < j$ and $\mathbf{c}_i = 1$ at $i \geq j$. In this case, only the slope I_j changes, and the constraints are violated at the ($j - 1$)th and jth CVs. They are compensated by changing σ_{j-1} and σ_j in view of the fact that the increments $\Delta I_j = 1/S_j$ and $\Delta I_{j+1} = -1/S_j$. The next basis vector is constructed.

$$\mathbf{c} = (0 \ldots 0 \underset{j}{1} \ldots \underset{m+r}{1} \; 0 \ldots 0 \underset{m+r+j-1}{-1/(\delta_{j-1} S_j)} \; 1/(\delta_j S_j) \; 0 \ldots 0)^{\mathrm{T}}.$$

If, at some j of $k < j < q$, the curvature σ_j is limiting, then the number of basis vectors is decreased by 1, and in constructing each of them, ΔI_j is retained, and the violation of the difference of the slopes at other CVs is compensated using the curvature at the CVs with the nonlimiting curvature values.

If the leg under active constraints of type (3) contains more than two limiting slopes in, the basis vectors are constructed similarly.

If two legs of the considered form share one common CV to which an inactive constraint of type (3) corresponds, then, for these legs, the basis vectors are constructed as for an integral whole. But for the basis vector obtained by the rotation with the center at this CV, the curvature is not required to be changed. If two legs have no common CVs, then they are considered separately.

To satisfy the conditions for the fixed initial and final points and directions, these conditions are converted to constraints of the form $z_{1,\,\min} \leq z_1 \leq z_{1,\,\max}$ and $z_{n,\,\min} \leq z_n \leq z_{n,\,\max}$ [25].

If some of them becomes active, then the shift vector is not included in the basis. If an active constrain of type (3) is imposed at CV_1 or CV_n, then the vector of rotation with the center at these points is constructed with the compensation of the difference of the slopes by changing the corresponding curvature.

The height constraints at points in the inscribed curves are nonlinear and have to be taken into account by penalty functions [25].

To solve the question of the possibility of eliminating a constraint from the active set, it is necessary to construct vector **g** that violates this and only this constraint. If $(\mathbf{f}, \mathbf{g}) < 0$, the constraint is excluded. For active constraints of type (3), this is a basis vector, but without compensation at the center of rotation. And if the corresponding curvature is limiting, then it is necessary to construct vector **g** as a basis vector with compensation. If it does not violate the curvature constraint and $(\mathbf{f}, \mathbf{g}) > 0$, then the curvature constraint can be excluded. For the active constraint on the slope I_k (Fig. 6), such a vector is obtained by allowing the rotation of the right-hand part of the leg with the center at the ($k - 1$)th CV, which was not used in the construction of the basis, with the compensation of the change in ΔI_{k-1}.

If the active set does not contain constraints of type (3), then the question of the possibility of eliminating such constraints from the active set is solved quite simply [25].

CONCLUSIONS

The proposed method to construct the basis enables one to solve the problem of optimization of parameters of a spline with circular arcs and at variable CV abscissas obtained at the first stage. This question, as well as the optimization of parameters of a spline that is not a one-to-one function, which often takes place in designing the plan of road routes, requires a separate consideration.

As was determined as far back as the 1970s−1980s [1, 26], using adequate mathematical models and correct optimization algorithms yielded a significant economic effect. The Profil, Profil-r, Profil-2a, and Profil-2r systems, which were used at that time on slow (by modern standards) computers for designing the longitudinal profile of roads and railways [1, 26], are currently not used, first of all, because of the absence of entities interested in reducing the cost estimate of construction and reconstruction by improving project quality. These systems were replaced by foreign-made CAD systems, which accelerated the processes of preparation and release of numerous drawings and other project documents. However, these systems do not include designing programs. In an expert designer's apt words, they are "convenient drawing tools with no signs of optimization." On the other hand, both updated old systems of longitudinal profile design, and new programs of route plan design solve complex problems of optimization and visualization of computer design solutions, but they cannot completely replace the used foreign-made interactive-design CAD systems.

The point is that designing systems are developed "by inertia," as a personal initiative, without funding sources; therefore, they do not contain subsystems of preparation and release of drawings and various output documents.

The emerging trend toward artificial intelligence in other sciences and technologies gives promise that designing programs will also be in demand in routing of linear structures, which will significantly reduce the labor and money inputs in construction by using intelligent design systems.

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

REFERENCES

1. Struchenkov V.I. The use of mathematical optimization methods and a computer in the design of the longitudinal profile of railways; B.K. Malyavskii (Ed.). *Seriya: Trudy Vsesoyuznogo nauchno-issledovatel'skogo instituta transportnogo stroitel'stva = Series: Proceedings of the All-Union Scientific Research Institute of Transport Construction*. Iss. 101. Moscow: Transport; 1977. 169 p. (in Russ.).
2. Struchenkov V.I. Computer technologies in line structure routing. *Rossiiskii tekhnologicheskii zhurnal = Russian Technological Journal*. 2017;5(1):29−41 (in Russ.). https://doi.org/10.32362/2500-316X-2017-5-1-29-41
3. Ahlberg J.H., Nilson E.N., Walsh J.L. The theory of splines and their applications. Academic press; 1967. 296 p. [Alberg J., Nilson E., Walsh J. *Teoriya splainov i ee prilozheniya* (*The theory of splines and their applications*). Moscow: Mir; 1972. 312 p. (in Russ.).]
4. Khakimov B.V. *Modelirovanie korrelyatsionnykh zavisimostei splainami na primerakh v geologii i ekologii* (*Modeling of correlation dependences by splines on examples in geology and ecology*). St. Petersburg: Neva; 2003. 144 p. (in Russ). ISBN 5-7654-2951-3
5. Dierckx P. *Curve and Surface fitting with splines*. Oxford University Press; 1995. 285 p.
6. Mikhalevich V.S., Bykov V.I., Sibirko A.N. To the question of designing the optimal longitudinal profile of the road. *Transportnoe stroitel'stvo = Transport Construction*. 1975;6:39−40 (in Russ.).
7. Kosmin V.V., Struchenkov V.I., Fradkov E.B. Computer design of the longitudinal profile of the road. *Transportnoe stroitel'stvo = Transport Construction*. 1971;4:38−42 (in Russ.).
8. Bentley Rail Track. Available from URL: https://www.bentley.com/-/media/1EA2B937CB5B42BEA5EAE802620C0BA3.ashx
9. CARD/1. Available from URL: http://card-1.ru/
10. Autodesk. Available from URL: https://www.architect-design.ru/autodesk/autocad/
11. Topomatic Robur. Available from URL: http://www.topomatic.ru/
12. Credo-Dialog. Available from URL: https://credo-dialogue.ru/
13. Struchenkov V.I. The use of parabolic splinees in CAD of linear structures. *Rossiiskii tekhnologicheskii zhurnal = Russian Technological Journal*. 2018;6(1):40−51 (in Russ). https://doi.org/10.32362/2500-316X-2018-6-1-40-52
14. 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

СПИСОК ЛИТЕРАТУРЫ

1. Струченков В.И. Использование математических методов оптимизации и ЭВМ при проектировании продольного профиля железных дорог; под ред. Б.К. Малявского. Серия: *Труды Всесоюзного научно-исследовательского института транспортного строительства.* Вып. 101. М.: Транспорт; 1977. 169 с.
2. Струченков В.И. Компьютерные технологии в проектировании трасс линейных сооружений. *Российский технологический журнал*. 2017;5(1):29−41. https://doi.org/10.32362/2500-316X-2017-5-1-29-41
3. Альберг Дж., Нильсон Э., Уолш Дж. *Теория сплайнов и ее приложения*: пер. с англ. М.: Мир; 1972. 312 с.
4. Хакимов Б.В. *Моделирование корреляционных зависимостей сплайнами на примерах в геологии и экологии*. СПб.: Нева; 2003. 144 с. ISBN 5-7654-2951-3
5. Dierckx P. *Curve and Surface fitting with splines*. Oxford University Press; 1995. 285 p.
6. Михалевич В.С., Быков В.И., Сибирко А.Н. К вопросу проектирования оптимального продольного профиля дороги. *Транспортное строительство*. 1975;6:39−40.
7. Космин В.В., Струченков В.И., Фрадков Е.Б. Проектирование продольного профиля дороги на ЭВМ. *Транспортное строительство*. 1971;4:38−42.
8. Bentley Rail Track. URL: https://www.bentley.com/-/media/1EA2B937CB5B42BEA5EAE802620C0BA3.ashx
9. CARD/1. URL: http://card-1.ru/
10. Autodesk. URL: https://www.architect-design.ru/autodesk/autocad/
11. Topomatic Robur. URL: http://www.topomatic.ru/
12. Credo-Dialog. URL: https://credo-dialogue.ru/
13. Струченков В.И. Использование параболических сплайнов в САПР линейных сооружений. *Российский технологический журнал.* 2018;6(1):40−52. https://doi.org/10.32362/2500-316X-2018-6-1-40-52
14. Струченков В.И. *Методы оптимизации трасс в САПР линейных сооружений*. М.: Солон-Пресс; 2015. 272 с. ISBN 978-5-91359-139-5
15. Лежнев А.В. *Динамическое программирование в экономических задачах*. М.: Бином; 2016. 285 с. ISBN 5-94774-344-2
16. Cavagnari G., Marigonda A., Piccoli B. Generalized dynamic programming principle and sparse mean-field control problems. *Journal of Mathematical Analysis and Applications*. 2020;481(1):123437. https://doi.org/10.1016/j.jmaa.2019.123437
17. He S., Shin H.-S., Tsourdos A. Computational guidance using sparse Gauss-Hermite quadrature differential dynamic programming. *IFAC-PapersOnLine*. 2019;52(12):13−18. https://doi.org/10.1016/j.ifacol.2019.11.062

15. Lezhnev A.V. *Dinamicheskoe programmirovanie v ekonomicheskikh zadachakh* (*Dynamic programming in economic problems*). Moscow: Binom; 2016. 176 p. (in Russ). ISBN 5-94774-344-2
16. Cavagnari G., Marigonda A., Piccoli B. Generalized dynamic programming principle and sparse mean-field control problems. *Journal of Mathematical Analysis and Applications*. 2020;481(1):123437. https://doi.org/10.1016/j.jmaa.2019.123437
17. He S., Shin H.-S., Tsourdos A. Computational guidance using sparse Gauss-Hermite quadrature differential dynamic programming. *IFAC-PapersOnLine*. 2019;52(12):13−18. https://doi.org/10.1016/j.ifacol.2019.11.062
18. Fayaed S.S., Fiyadh S.S., Khai W.J., Ahmed A.N., Afan H.A., Ibrahim R.K. Improving dam and reservoir operation rules using stochastic dynamic programming and artificial neural network integration model. *Sustainability*. 2019;11(19):5367. https://doi.org/10.3390/su11195367
19. Işik H., Sintunavarat W. An investigation of the common solutions for coupled systems of functional equations arising in dynamic programming. *Mathematics*. 2019;7(10):977. https://doi.org/10.3390/math7100977
20. Karpov D.A., Struchenkov V.I. Dynamic Programming as a Method of Spline Approximation in the CAD Systems of Linear Constructions. *Rossiiskii tekhnologicheskii zhurnal = Russian Technological Journal*. 2019;7(3):77−88 (in Russ.). https://doi.org/10.32362/2500-316X-2019-7-3-77-88
21. Karpov D.A., Struchenkov V.I. *Metody i algoritmy resheniya prikladnykh zadach diskretnoi optimizatsii* (*Methods and algorithms for solving applied discrete optimization problems*). Moscow: Solon-Press; 2020. 201 p. (in Russ.). ISBN 978-5-91359-399-3
22. Kochenderfer M., Wheeler T. *Algorithms for Optimization*. The MIT Press; 2019. 521 p.
[Kokhenderfer M., Uiler T. *Algoritmy optimizatsii*. Moscow: Vil'yams; 2020. 528 p. (in Russ.). ISBN 978-5-907144-76-7]
23. Chernorutskii I.G. *Metody optimizatsii. Komp'yuternye tekhnologii* (*Optimization methods. Computer technologies*). St. Petersburg: BHV-Petersburg; 2011. 370 p. (in Russ.). ISBN 978-5-9775-0784-4
24. Ovchinnikov V.A. *Modeli i metody diskretnoi optimizatsii* (*Models and methods of discrete optimization*). Moscow: MGTU im. N.E. Baumana; 2019. 278 p. (in Russ.). ISBN 978-5-7038-5105-0
25. Struchenkov V.I. *Prikladnye zadachi optimizatsii* (*Applied optimization problems*). Moscow: Solon-Press; 2016. 314 p. (in Russ.). ISBN 978-5-91359-191-3
26. Karikh Yu.S. Evaluation of existing longitudinal profile design methods. In: Collection of works of GiprodorNII: *Improving the economic efficiency of investment in the construction, repair and maintenance of roads*. Moscow: Izdanie GiprodorNII; 1976. P. 105−112. (in Russ.).

18. Fayaed S.S., Fiyadh S.S., Khai W.J., Ahmed A.N., Afan H.A., Ibrahim R.K. Improving dam and reservoir operation rules using stochastic dynamic programming and artificial neural network integration model. *Sustainability*. 2019;11(19):5367. https://doi.org/10.3390/su11195367
19. Işik H., Sintunavarat W. An investigation of the common solutions for coupled systems of functional equations arising in dynamic programming. *Mathematics*. 2019;7(10):977. https://doi.org/10.3390/math7100977
20. Карпов Д.А., Струченков В.И. Динамическое программирование как метод сплайн- аппроксимации в САПР линейных сооружений. *Российский технологический журнал*. 2019;7(3):77−88. https://doi.org/10.32362/2500-316X-2019-7-3-77-88
21. Карпов Д.А., Струченков В.И. *Методы и алгоритмы решения прикладных задач дискретной оптимизации*. М.: Солон-Пресс; 2020. 201 с. ISBN 978-5-91359-399-3
22. Кохендерфер М., Уилер Т. *Алгоритмы оптимизации*. М.: Вильямс; 2020. 528 с. ISBN 978-5-907144-76-7
23. Черноруцкий И.Г. *Методы оптимизации. Компьютерные технологии*. СПб.: БХВ-Петербург; 2011. 370 с. ISBN 978-5-9775-0784-4
24. Овчинников В.А. *Модели и методы дискретной оптимизации.* М.: МГТУ им. Н.Э. Баумана; 2019. 278 с. ISBN 978-5-7038-5105-0
25. Струченков В.И. *Прикладные задачи оптимизации*. М.: Солон-Пресс; 2016. 314 с. ISBN 978-5-91359-191-3
26. Карих Ю.С. Оценка существующих методов проектирования продольного профиля. В сб. трудов ГипродорНИИ. Вып. 17. *Повышение экономической эффективности капиталовложений в строительство, ремонт и содержание автомобильных дорог*. М.: Издание ГипродорНИИ; 1976. С. 105−112.

About the authors

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

Valery I. Struchenkov, Dr. Sci. (Eng.), Professor, General Informatics Department, Institute of Cybernetics, MIREA – Russian Technological University (78, Vernadskogo pr., Moscow, 119454 Russia). E-mail: srt1942@mail.ru. https://orcd.org/0000-0002-9801-7454

Об авторах

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

Струченков Валерий Иванович, д.т.н., профессор, кафедра общей информатики Института кибернетики, ФГБОУ ВО «МИРЭА – Российский технологический университет» (119454, Россия, Москва, пр-т Вернадского, д. 78). E-mail: srt1942@mail.ru. https://orcid.org/0000-0002-9801-7454

Translated by V. Glyanchenko