

Математическое моделирование  
Mathematical modeling

УДК 519.857

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

НАУЧНАЯ СТАТЬЯ

## Двухэтапная сплайн-аппроксимация в компьютерном проектировании трасс линейных сооружений

Д.А. Карпов,  
В.И. Струченков <sup>®</sup>

МИРЭА – Российский технологический университет, Москва, 119454 Россия

<sup>®</sup> Автор для переписки, e-mail: [str1942@mail.ru](mailto: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.

## 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](mailto: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, dynamic 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 (in Russ.). <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.

## ВВЕДЕНИЕ

Линейные сооружения – это сооружения, положение которых на местности определяется осью сооружения, называемой трассой. К ним относятся железные и автомобильные дороги, трубопроводы различного назначения, каналы, водоводы и др. Трасса – это трехмерная кривая, которая традиционно представляется двумя плоскими кривыми – планом и продольным профилем.

План трассы – это ее проекция на плоскость  $XOY$ , а продольный профиль – график функции  $Z(s)$ , где  $s$  – длина кривой в плане, отсчитываемая от заданной начальной точки. Продольный профиль получается при развертке на плоскость вертикальной поверхности, проходящей через трассу.

Независимо от вида сооружения, проектирование продольного профиля можно рассматривать как построение сплайна, состоящего из элементов заданного вида. Этот сплайн должен минимально (в заданном смысле) отклоняться от исходной ломаной линии, которая при проектировании новых сооружений является профилем земли, а при проектировании реконструкции – это профиль существующего сооружения.

Простейшим сплайном первого порядка является проектная линия продольного профиля железной дороги. В этом случае задача состоит в преобразовании исходной ломаной линии (профиль земли) в другую ломаную линию, удовлетворяющую целому ряду ограничений: на уклоны элементов и разности уклонов смежных элементов, минимальную длину элементов, высотные ограничения в отдельных точках и зонах [1, 2]. В силу малости проектных уклонов длина элемента и разность абсцисс его концов практически совпадают; разность уклонов смежных элементов отождествляется с углом поворота, а уклон – с углом элемента с осью абсцисс.

При этом число элементов искомого сплайна неизвестно. Это обстоятельство, а также наличие многочисленных ограничений, является существенным отличием рассматриваемой проектной задачи сплайн-аппроксимации от задач, решаемых в теории сплайнов и ее приложениях [3–5], где число узлов сплайна и их абсциссы считаются заданными, а ограничения, как правило, отсутствуют.

В упрощенной постановке задача поиска оптимального сплайна в виде ломаной линии при неизвестном числе элементов и ограничениях была решена в прошлом веке применительно к проектированию продольного профиля новых железных дорог [6, 7].

Задача решалась в два этапа. На первом этапе исходный профиль земли преобразовывался в

ломаную, составленную из коротких элементов, с соблюдением всех ограничений, кроме ограничения на длину элемента. Такой профиль был назван работчиками первых проектирующих алгоритмов – «цепочка» [6].

На втором этапе «цепочка» преобразовывалась в проектную линию с соблюдением всех ограничений, включая ограничения по длине элементов.

В реальной постановке применительно к проектированию в условиях пересеченного рельефа и сложной геологии задача была решена на ЭВМ БЭСМ-4 методом нелинейного программирования. Соответствующая программа получила широкое практическое применение, несмотря на большое время счета из-за крайне низкого быстродействия этой и последующих моделей ЭВМ (Минск 32, ЕС 1020 и др.) прошлого века [1].

В получивших широкое распространение в нашей стране САПР, разработанных западными фирмами [8–10], их отечественными [11] и белорусским аналогами [12], компьютер используется для решения вспомогательных задач, но не для выработки оптимальных проектных решений. В этих системах задача сплайн-аппроксимации решается «через глаза», то есть проектировщик должен задать ту или иную информацию, которая полностью определяет искомую линию. В лучшем случае он рассматривает несколько из возможных решений, которых теоретически бесконечно много.

В настоящее время требуется совершенствование математической модели, алгоритма и ранее разработанных проектирующих программ в связи с изменением технических требований при проектировании высокоскоростных железных дорог. Сплайн первого порядка должен быть заменен на сплайн, состоящий из прямых и дуг окружностей, число которых остается неизвестным.

Подобный сплайн используется и при проектировании трубопроводов большого диаметра.

При проектировании продольного профиля автомобильных дорог возникает задача поиска параболического сплайна второго порядка [13] с уже отмеченными принципиальными особенностями. Эта задача была решена с применением нелинейного программирования [13].

Сплайн с дугами окружностей используется, как альтернатива сплайну с параболой, при проектировании продольного профиля автодорог, а также плана трасс различных линейных сооружений [14].

Целью настоящей статьи является рассмотрение упомянутых проектных задач с единых теоретических позиций, как задач сплайн-аппроксимации, и изложение принципиальных пунктов и особенностей алгоритмов их решения.

## 1. ПОСТАНОВКА ЗАДАЧИ АППРОКСИМАЦИИ СПЛАЙНОМ С ДУГАМИ ОКРУЖНОСТЕЙ И ЕЕ ФОРМАЛИЗАЦИЯ

Рассмотрим задачу проектирования продольного профиля прямолинейными элементами, которые сопрягаются дугами окружностей.

Если проектируется реконструкция, то исходным является профиль существующего сооружения. Если проектируется новое сооружение, то исходным является профиль земли. «Цепочный» продольный профиль (пунктирная линия на рис. 1), для получения которого имеются проектирующие программы [14], используется для поиска числа элементов сплайна. Равенство длин элементов «цепочки» не требуется, но абсциссы ее узлов и абсциссы узлов исходной ломаной линии совпадают.

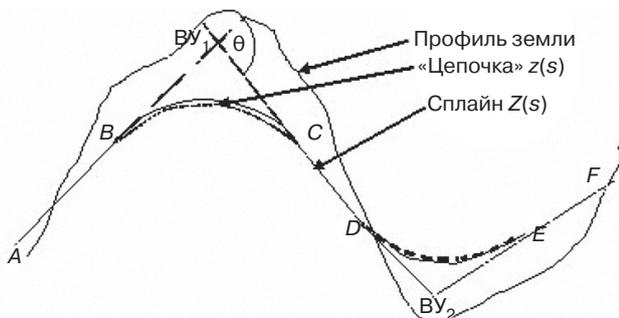


Рис. 1. Сплайн с дугами окружностей

Итак, имеем ломаную линию  $z(s)$ , которую требуется преобразовать с минимальными отклонениями в сплайн  $Z(s)$ , состоящий из отрезков прямых, сопрягаемых дугами окружностей (рис. 1).

Ограничения:

- 1) на уклоны  $I_j$  прямолинейных элементов сплайна:  $-I_{\min} \leq I_j \leq I_{\max}$ ,  $j = 1, 2, \dots, N - 1$ , где  $N$  – число узлов сплайна (вершин искомой ломаной, далее – ВУ). Фактически это ограничение на первую производную функции  $Z(s)$ .
- 2) на радиусы (кривизну) вписанных выпуклых и вогнутых кривых:  $1/R_{\text{вып}} \leq 1/R_j \leq 1/R_{\text{вог}}$ ,  $j = 1, 2, \dots, N$ ,  $R_{\text{вып}} < 0$ , а  $R_{\text{вог}} > 0$ .
- 3) на длины дуг окружностей ( $BC$ ,  $DE$  на рис. 1):  $L_{\text{кр}j} \geq L_{\text{кр} \min}$ .
- 4) на длины прямых вставок между кривыми ( $CD$  на рис. 1):  $L_{\text{пр}j} \geq L_{\text{пр} \min}$ .

Дополнительно могут быть заданы ограничения на ординаты отдельных точек (высотные ограничения в местах пересечения водотоков, других коммуникаций и т.д.)

### Целевая функция

Равные отклонения в разные стороны от исходной линии часто бывают неравноценны. Поэтому традиционно используемая минимизация суммы

квадратов отклонений в заданных точках (в том числе и с различными весами) оказывается неприемлемой.

При проектировании новых дорог в качестве целевой функции на данном этапе могут быть приняты суммарные объемы земляных работ в насыпях и в выемках. Строительные затраты могут быть приняты в качестве целевой функции, если в насыпях и в выемках нет взаимосвязи элементов, которая возникает при использовании грунтов выемок для сооружения насыпей и требует рассмотрения проектной линии как единого целого [14], как в нелинейном программировании.

На этапе преобразования исходной ломаной («цепочки» или существующего профиля) в сплайн нужного вида отклонения по ординатам (рабочие отметки) невелики (порядка 0.5 м [14]), что позволяет использовать упрощенные критерии оптимальности, поскольку цель данного этапа – определить число элементов и их примерное расположение, то есть построить начальное приближение для применения нелинейного программирования.

При проектировании реконструкции продольного профиля дороги на данном этапе целесообразно использовать моделирующие функции, в которых учтены конкретные особенности задачи.

Так, при проектировании продольного профиля при реконструкции железных дорог прямолинейными элементами без учета влияния круговых кривых, которые вписывались в найденную линию, успешно использовалась гладкая моделирующая функция  $F(h)$  (сплайн второго порядка с дефектом, равным единице), график которой представлен на рис. 2. Здесь  $h$  – рабочая отметка, т.е. разность ординат искомого и исходного сплайна  $h(s) = Z(s) - z(s)$ . Величины  $h_0$ ,  $\Delta$  и параметры элементов  $F(h)$  определялись, исходя из существующих и проектируемых толщины балласта ( $H_{\text{сущ}}$  и  $H_{\text{пр}}$  соответственно), высоты рельсов и шпал.  $\Delta = \max(0, H_{\text{сущ}} - H_{\text{пр}})$ , а участки графика  $F(h)$  соответствуют: 1 – досыпке балласта, 2 – срезке балласта, 3 – срезке земляного полотна.

При  $\Delta = 0$  участок 2 графика  $F(h)$  отсутствует. При равенстве высот существующих и проектируемых рельсов и шпал  $h_0 = 0$ .

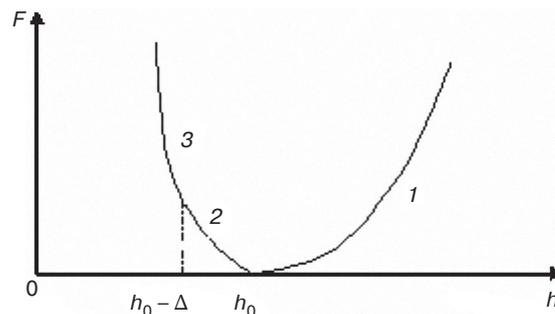


Рис. 2. Моделирующая функция

Значения  $F(h_i)$  вычислялись в узлах исходного сплайна, и целевая функция имела вид:

$$\min \Phi(\mathbf{h}) = \sum_1^k v_i F(h_i), \quad (1)$$

где  $k$  – число узлов исходной ломаной,  $\mathbf{h} = (h_1, h_2, \dots, h_k)$  – вектор неизвестных, а весовые коэффициенты  $v_i$  равны полусуммам длин ее смежных элементов. Аналогичные моделирующие функции использовались при проектировании продольного профиля реконструируемых автодорог параболическими сплайнами [13].

Если целевая функция – объем земляных работ, то  $F(h_i)$  – площадь поперечного сечения в  $i$ -й точке – остается кусочно-квадратической и соответствует вычислению объема как интеграла от площади по формуле трапеций.

## 2. СПЛАЙН-АППРОКСИМАЦИЯ ПО МЕТОДУ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Динамическое программирование при выполнении ряда условий [15–20] позволяет создать алгоритмы построения в несколько этапов оптимальной траектории перемещения некоторой системы из заданного начального состояния в конечное, решая на каждом из этапов однотипные задачи, более простые по сравнению с исходной задачей [21]. Варианты достижения одного и того же состояния различными путями считаются сравнимыми, и в каждом состоянии остается только лучший по заданному критерию вариант.

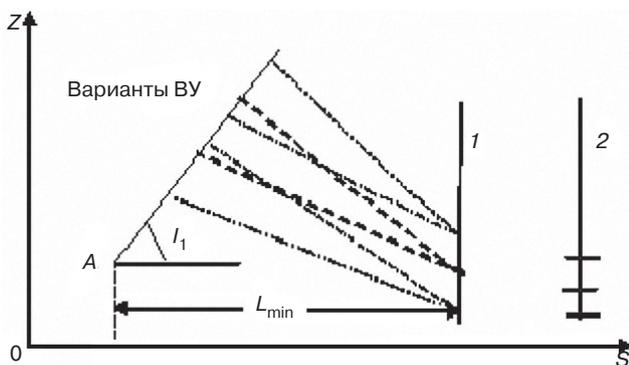


Рис. 3. Первый шаг алгоритма

Ключевое понятие динамического программирования – «состояние системы» для нашей задачи определим следующим образом: точка на прямолинейном элементе сплайна, начиная с которой можно строить дугу окружности или продолжить прямую, плюс угол этого элемента с осью абсцисс  $OS$ . Начальное состояние (точка  $A$  и угол  $I_1$  на рис. 3) считаются заданными.

Относительно узлов исходного сплайна (ломаной  $z(s)$ ) с заданным шагом разбивается сетка

варьирования (на рис. 3 – точки на вертикалях 1, 2). В каждой из этих точек, исходя из уклона исходного сплайна, назначаются углы с осью  $OS$  (сектор поиска). Шаг назначения точек и углов на вертикали и число шагов по вертикали и углам в каждой из них являются исходными параметрами алгоритма поиска проектного сплайна и задаются до начала расчета. При необходимости расчеты могут выполняться сначала с крупными дискретами, а затем с использованием полученного решения с меньшими дискретами. Это обычный прием сокращения времени счета при использовании динамического программирования, который успешно применялся при параболической сплайн-аппроксимации [13, 14].

В качестве первой выбирается вертикаль, отстоящая от начальной точки  $A$  (по абсциссам) на расстоянии  $L_{\min}$ , которое задается исходя из ограничений 3 и 4 на длины кривых и прямых вставок. Кроме того задается расстояние  $L_{\max}$  – максимальная длина кривой плюс длина прямой вставки. Для реализации динамического программирования используются углы поворота (рис. 1) и координаты их вершин.

При поиске первой кривой проектного сплайна левая сторона угла задана (точка  $A$  и угол  $I_1$ ). Точки на каждой вертикали в пределах от  $L_{\min}$  до  $L_{\max}$  в совокупности с назначаемыми в каждой из них направлениями образуют возможные варианты правой стороны первого угла поворота (рис. 3) и определяют соответствующие варианты первой ВУ. Для каждого варианта с использованием минимального радиуса выпуклой или вогнутой кривой (в зависимости от знака угла поворота) анализируется возможность выполнения всех ограничений, и остаются только те варианты, для которых все ограничения выполнены. Кроме того, рассматривается возможность без нарушения ограничений увеличить радиус вписанной дуги с использованием разности уклонов смежных элементов исходного сплайна, попадающих в рассматриваемый угол. Выбирается радиус, при котором значение целевой функции для соответствующей ВУ минимально.

На первом шаге сравнения путей и отбраковки вариантов достижения одного и того же состояния не происходит. Каждое из допустимых состояний первого шага вместе с соответствующими значениями целевой функции (затрат на его достижение из начальной точки), координат ВУ, радиуса и угла поворота запоминается.

### Общий шаг алгоритма

Рассматриваются узлы исходного сплайна, абсциссы которых (и соответствующие вертикали) находятся в пределах от  $S_A + 2L_{\min}$  до  $S - L_{\min}$ , где  $S$  – абсцисса конца профиля. Для каждой вертикали

рассматриваются последовательно все предшествующие узлы, отстоящие от нее не менее чем на  $L_{\min}$  и не более чем на  $L_{\max}$ , и проходящие через них прямые. Для каждого пересечения последовательно выполняются те же действия, что и на первом шаге. Отличие состоит в том, что для рассматриваемого состояния (правая сторона угла, линия  $BC$  на рис. 4) возможно много пересечений со сторонами предшествующих углов, исходящих из одной или из разных ВУ (точки  $A_i$  и  $A_{i+1}$  на рис. 4). Как и на первом шаге, рассматриваются и сравниваются только допустимые по ограничениям соединения. В итоге в каждое состояние на каждой вертикали (точка  $C$  плюс угол) приводит один вариант или ни одного. Для каждого из таких вариантов дополнительно запоминается точка и направление (точка  $A_i$  и угол  $A_iB$  с осью абсцисс на рис. 4), которые соответствуют левой стороне угла, то есть для каждого нового состояния запоминается связь с лучшим из предшествующих состояний.

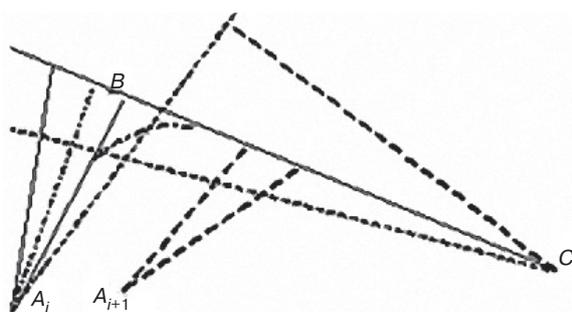


Рис. 4. Сравнение и отбраковка вариантов

### Последний шаг алгоритма

На последнем шаге правая сторона угла поворота известна. Это конечная точка и угол, задающий конечное направление. Выполняются те же действия, что и на общем шаге, и сравнением допустимых соединений определяется минимальное значение целевой функции. Обычным для динамического программирования разворотом по запомненным связям [21, 22] восстанавливается оптимальный сплайн и его параметры.

Конечно, можно рассматривать и несколько конечных направлений и точек и выполнять те же действия для каждой из них с последующим выбором лучшего варианта.

При реализации алгоритма встретились сложности при наличии длинных прямых. Получались малые углы поворота. В зависимости от конкретной задачи такие углы или не допускаются вообще или в них не вписываются кривые (например, при проектировании автодорог низких категорий). В первом случае вместо двух пересекающихся прямых можно было в процессе перебора вариантов образовывать

одну (по крайним точкам). Но поскольку алгоритм предназначен только для построения начального приближения, то во избежание излишних усложнений такие преобразования производились только для полученного сплайна. Это оправдано еще и тем, что далеко не все такие пересечения под малыми углами попадают в окончательное решение.

### 3. ОПТИМИЗАЦИЯ ПАРАМЕТРОВ СПЛАЙНА

Проектная линия полностью определяется координатами вершин углов поворота и радиусами вписанных окружностей, полученными по методу динамического программирования (рис. 5).

Для начала считаем неизменными абсциссы ВУ, то есть рассматриваем возможность оптимизации положения сплайна за счет перемещения ВУ по фиксированным вертикалям. В силу малости уклонов (они не превышают нескольких десятков промилле) длины сторон каждого угла считаются равными разности абсцисс ВУ, которые не изменяются. Поскольку начальная и конечная точка и направления в них заданы, ординаты первой и последней ВУ не могут измениться. Поэтому переменными считаем только  $Z_j, j = 1, 2, \dots, n$ , ординаты варьируемых ВУ (их число  $n = N - 2$ ) и радиусы вписанных кривых  $R_j$ . Заданные граничные условия учитываются вычислением предельных значений уклонов  $I_1$  и  $I_n$  и затем ординат  $Z_1$  и  $Z_n$  [14].

Чтобы получить задачу нелинейного программирования с целевой функцией  $\Phi(\mathbf{h})$  (1), через эти неизвестные нужно выразить рабочие отметки в узлах исходной ломаной, т.е. разности ординат проектного сплайна и исходной ломаной ( $B'B''$  на рис. 5), и все ограничения.

При проектировании продольного профиля новых дорог целевая функция соответствует минимуму затрат на сооружение земляного полотна и на искусственные сооружения. Соответствующие модели – те же, что и при использовании параболических сплайнов при совместном проектировании продольного и поперечных профилей с учетом распределения земляных масс [14].

При наличии таких выражений вычисление градиента целевой функции (1) сводится к простому пересчету производных [14], так как ординаты точек ( $D, B$  на рис. 5) на прямолинейных элементах линейно зависят от ординат ВУ. В силу малости уклонов угол поворота считается равным разности смежных уклонов ( $\Delta I_j$  на рис. 5).

Это позволяет с достаточной точностью выразить отклонения точек кривой от соответствующих точек прямых ( $CC'', BB''$  на рис. 5), то есть поправки к рабочим отметкам, вычисленным по сторонам угла поворота («стрелы»).

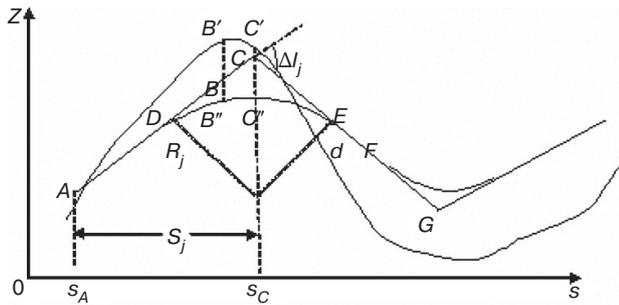


Рис. 5. К пересчету производных при наличии дуг окружностей

На рис. 5  $CC'' = \delta_j = R_j \Delta_j^2/8$ ;  $BB'' = \delta_B = \delta_j - t_B \Delta_j/2 + t_B^2/(2R_j)$ , где  $t_B = |s_C - s_B|$  – разность абсцисс ВУ и точки на кривой;  $\Delta_j = I_{j+1} - I_j$ , а  $I_j = (Z_j - Z_{j-1})/S_j$ . Здесь  $Z_j$  – неизвестные проектные отметки вершин углов поворота, а  $S_j$  – разности абсцисс, которые несущественно отличаются от длин сторон угла;  $s_C - s_A = S_j \approx AC$ .

Вместо ограничений на разность уклонов есть ограничения на минимальные длины кривых:  $R_j \Delta_j \geq L_{j \min}$  и на минимальную длину прямой вставки, то есть суммы  $CE + FG$  (рис. 5) должны удовлетворять условию:

$$R_j \Delta_j/2 + R_{j+1} \Delta_{j+1}/2 + L_{\text{пр min}} \leq S_{j+1}, j = 1, 2, \dots, n. \quad (2)$$

Здесь  $L_{\text{пр min}}$  – заданная минимальная длина прямой вставки,  $n$  – число ВУ.

При малых  $\Delta_j$  для изменения длины прямой вставки на 10 м требуется изменить радиус на 1000 м и более, что при оптимизации сплайна, полученного по методу динамического программирования, маловероятно. Поэтому условие (2) можно упростить и исключить взаимосвязь переменных, относящихся к смежным ВУ, используя полученный на первом этапе сплайн как начальное приближение.

Это можно сделать, выполняя следующие действия:

1. Вычисляем все  $T_j = R_j \Delta_j/2$  (в проектной практике они называются тангенсами).
2. Вычисляем все прямые вставки  $d_j = S_j - (T_{j-1} + T_j)$ ,  $j = 2, \dots, n$  и  $c_j = d_j - L_{\text{пр min}}$  («запас»).
3. Если  $d_j = L_{\text{пр min}}$ , то  $T_{j-1}$  и  $T_j$  фиксируем как максимальные значения  $R_{j-1} \Delta_{j-1}/2$  и  $R_j \Delta_j/2$ . Зафиксированные значения далее не меняем.
4. Последовательно рассматриваем прямые вставки в порядке возрастания, начиная с наименьшей  $d_k$ . Значения  $T_{k-1} + T_k$  можно увеличить на величину  $c_k = d_k - L_{\text{пр min}}$ , не рискуя нарушить ограничение по прямой вставке на соседних элементах. Если максимальные значения  $R_{k-1} \Delta_{k-1}/2$  и  $R_k \Delta_k/2$  еще не зафиксированы, то в качестве их принимаем значения  $T_{k-1} + c_k/2$  и  $T_k + c_k/2$ , соответственно. Величины  $c_{k-1}$  и  $c_{k+1}$  уменьшаем

на  $c_k/2$ . Если было зафиксировано значение  $T_{k-1}$ , то величина  $\max(R_k \Delta_k/2) = T_k + c_k$ , и  $c_{k+1}$  уменьшаем на  $c_k$ . Если зафиксировано значение  $T_k$ , то  $\max(R_{k-1} \Delta_{k-1}/2) = T_{k-1} + c_k$ .

5. Переходим к п. 3 и продолжаем процесс, пока есть не зафиксированные максимальные значения  $R_j \Delta_j/2$ . При необходимости учитывается положение начальной и конечной точек профиля и корректируются (в сторону уменьшения) максимальные значения для  $R_1 \Delta_1/2$  и  $R_n \Delta_n/2$ .

Учитывая, что  $R_j \Delta_j$  – это длина  $j$ -й кривой и  $L_{\text{кр min}}$  – ее минимальное значение, обозначая вычисленные максимальные значения  $R_j \Delta_j$  как  $L_{j \max}$ , получаем систему двусторонних неравенств

$$L_{\text{кр min}} \leq R_j \Delta_j \leq L_{j \max}, j = 1, 2, \dots, n.$$

Эту систему нелинейных неравенств преобразуем в линейную систему, переходя от переменных радиусов к кривизнам  $\sigma_j = 1/R_j$ . Для ограничения по  $L_{j \max}$  имеем:  $\Delta_j \leq L_{j \max} \sigma_j$  при  $R_j > 0$  и  $L_{j \max} \sigma_j \leq \Delta_j$  при  $R_j < 0$ . Для ограничения по  $L_{\text{кр min}}$  получаем  $L_{\text{кр min}} \sigma_j \leq \Delta_j$  при  $R_j > 0$  и  $\Delta_j \leq L_{\text{кр min}} \sigma_j$  при  $R_j < 0$ .

Знаки  $R_j$  известны, следовательно, имеем линейную систему вида:

$$\alpha_j \sigma_j \leq \Delta_j \leq \beta_j \sigma_j, j = 1, 2, \dots, n. \quad (3)$$

При  $R_j > 0$   $\beta_j = L_{j \max}$  и  $\alpha_j = L_{\text{кр min}}$ . При  $R_j < 0$ , наоборот,  $\beta_j = L_{\text{кр min}}$  и  $\alpha_j = L_{j \max}$ .

Алгоритм решения задачи нелинейного программирования: найти  $\min \Phi(\mathbf{x})$ , где  $\mathbf{x}$  – вектор неизвестных,  $\Phi(\mathbf{x})$  – целевая функция, при линейных ограничениях  $\mathbf{Ax} \leq \mathbf{b}$  состоит из следующих пунктов:

1. Построение допустимого начального приближения.
2. Вычисление антиградиента  $\mathbf{f}$ .
3. Формирование матрицы активных ограничений  $\mathbf{A}_k$  и построение направления спуска  $\mathbf{p}$ .
4. Проверка условий окончания счета.

Если длина вектора спуска больше заданного  $\epsilon$ , переходим к п. 5. Иначе проверяем возможность исключения ограничений из активного набора. Если таких ограничений нет, то процесс окончен. Иначе исключаем одно из ограничений и переходим к п. 3.

5. Поиск шага по направлению спуска как минимального из шагов до границы и до точки минимума. При этом решается одномерная задача поиска минимума.
6. Переход в новую точку. Далее – к пункту 3, если антиградиент в новой точке уже вычислялся при поиске шага, иначе – к пункту 2.

Алгоритм в общем случае обеспечивает попадание в окрестность точки локального минимума. Поэтому важно получение хорошего начального приближения по методу динамического программирования.

Ключевыми являются два пункта: построение направления спуска и исключения ограничений из активного набора [22–24]. Задача может быть решена стандартными алгоритмами, которые требуют на каждой итерации решения систем линейных уравнений (обращения матрицы). Так, проекцию градиента на  $k$ -й итерации можно вычислить по формуле Розена:

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

Для решения вопроса об исключении ограничений из активного набора надо вычислить вектор  $\mathbf{u} = (\mathbf{A}_k \mathbf{A}_k^T)^{-1} \mathbf{A}_k \mathbf{f}$ , для чего требуется обращение матрицы  $\mathbf{A}_k \mathbf{A}_k^T$ .

Вместо этого рассмотрим возможности построения направления спуска, используя простую структуру системы ограничений [25]. Для этого нужно для любого активного набора уметь строить базис в нуль-пространстве матрицы  $\mathbf{A}_k$ , что было реализовано в действующей программе оптимизации сплайна в виде ломаной без вписанных кривых [14].

Так, если построена базисная матрица  $\mathbf{C}$ , то вектор спуска  $\mathbf{p} = \mathbf{C} \mathbf{C}^T \mathbf{f}$ , где  $\mathbf{f}$  – антиградиент.

Ограничения (3) содержат дополнительно переменные  $\sigma_j$ , но построенные ранее базисные векторы [25] можно преобразовать и для этой системы.

Если в нашей задаче некоторая переменная  $z_j$  не входит ни в одно из активных ограничений, то  $p_j = f_j$ . Наличие подобных свободных точек позволяет разбить профиль на участки независимого построения базисных векторов и соответствующих компонент вектора спуска. Например, для системы активных ограничений по прямой вставке (3) на участке с ВУ от  $(m+1)$ -й до  $(m+r-1)$ -й, схематично представленной на рис. 6, переменные  $z_{m-1}, z_m, \dots, z_{m+r-1}, z_{m+r}$  и  $\sigma_m, \sigma_{m+1}, \dots, \sigma_{m+r-1}$ ; свободные переменные  $z_{m-2}$  и  $z_{m+r+1}$ .

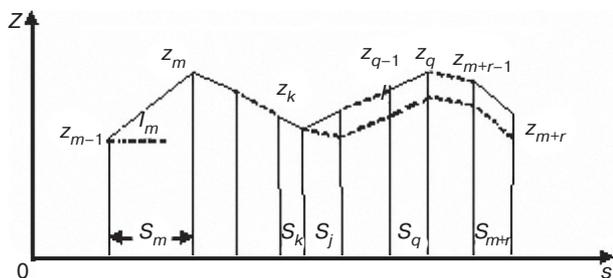


Рис. 6. Пример построения базисных векторов

Активные ограничения:

$$\begin{aligned} -\Delta I_m + \alpha_m \sigma_m &\leq 0, \\ -\Delta I_{m+1} - \beta_{m+1} \sigma_{m+1} &\leq 0, \\ \dots \\ -\Delta I_{m+r-1} - \alpha_{m+r-1} \sigma_{m+r-1} &\leq 0. \end{aligned} \quad (4)$$

В переменных ординатах эта система имеет вид:

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

Искомые базисные векторы должны обращать неравенства этой системы в равенства и быть линейно независимыми. Например, вектор  $\mathbf{c}_1 = (1 \ 1 \ \dots \ 1 \ 1 \ 0 \ 0)^T$  ( $r+2$  единицы и  $r$  нулей) задает сдвиг всех ВУ по оси ординат без изменения уклонов и радиусов. Очевидно, при этом не меняется и разность смежных уклонов и кривизна.

Если ко всем уклонам прибавить поровну, например, 1 (т.е. выполнить поворот с центром в  $(m-1)$ -й ВУ) и не менять радиусы, то ограничения системы (4) и соответствующей ей системы (5) останутся активными. Поэтому вектор  $\mathbf{c}_2 = (0 \ S_m \ S_m + S_{m+1} \ S_m + S_{m+1} + S_{m+2} \ \dots \ S_m + S_{m+1} + S_{m+2} + \dots + S_{m+r} \ 0 \ \dots \ 0)^T$  также можно включить в искомый базис. Еще  $r$  базисных векторов получим, выбирая последовательно центрами поворота  $m, m+1, \dots, (m+r-1)$ -е ВУ, меняя уклоны справа на 1 и компенсируя изменение разности уклонов в центре вращения изменением соответствующей кривизны (6).

Линейная независимость полученных векторов следует из способа их построения.

Если на таком участке некоторая кривизна  $\sigma_j$  приняла предельное значение, то соответствующая компонента вектора спуска равна нулю,  $\sigma_j$  исключается из числа переменных, учитываемых при построении базиса, и вектор, соответствующий ее изменению, не включается в базис.

Если предельное значение принял уклон некоторого элемента  $I_k$ , то вектор  $\mathbf{c}_1$  остается в базисе, но

$$\begin{aligned} \mathbf{c}_3 &= (0 \ 0 \ S_{m+1} \ S_{m+1} + S_{m+2} \ \dots \ S_{m+1} + S_{m+2} + \dots + S_{m+r} \ 1/\alpha_m \ 0 \ \dots \ 0)^T, \\ \mathbf{c}_4 &= (0 \ 0 \ 0 \ S_{m+2} \ \dots \ S_{m+2} + S_{m+3} \ \dots \ S_{m+2} + S_{m+3} + \dots + S_{m+r} \ 0 \ 1/\beta_{m+1} \ \dots \ 0)^T, \\ \dots \\ \mathbf{c}_{r+2} &= (0 \ 0 \ 0 \ 0 \ \dots \ 0 \ \dots \ S_{m+r} \ 0 \ \dots \ 0 \ 1/\alpha_{m+r-1})^T. \end{aligned} \quad (6)$$



инстанций, заинтересованных в снижении сметной стоимости строительства и реконструкции за счет повышения качества проектов. Сменившие их САПР западных фирм ускорили процесс подготовки и выдачи многочисленных чертежей и другой проектной документации. Однако они не содержат проектирующих программ. По меткому выражению одного из опытных проектировщиков это «удобные рисовалки и никакой оптимизации там нет». С другой стороны, как усовершенствованные старые системы проектирования продольного профиля, так и новые программы проектирования плана трасс решают сложные задачи оптимизации и визуализации компьютерных проектных решений, но они не могут полностью заменить используемые западные САПР интерактивного проектирования. Дело в том, что разработка

проектирующих систем ведется «по инерции», в порядке личной инициативы, без источников финансирования, и поэтому они не содержат подсистем подготовки и выдачи чертежей и разнообразных выходных документов.

Наметившийся в других областях техники и технологий поворот в сторону искусственного интеллекта дает основание надеяться, что проектирующие программы будут востребованы и при проектировании трасс линейных сооружений, что позволит существенно снизить затраты труда и средств в строительстве за счет применения интеллектуальных систем проектирования.

**Вклад авторов.** Все авторы в равной степени внесли свой вклад в исследовательскую работу.

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

### СПИСОК ЛИТЕРАТУРЫ

1. Струченков В.И. Использование математических методов оптимизации и ЭВМ при проектировании продольного профиля железных дорог; под ред. Б.К. Малавского. 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. Михалеви́ч 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

### 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/>

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. İşik 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.
13. Struchenkov V.I. The use of parabolic splines 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
15. Lezhnev A.V. *Dinamicheskoe programmirovaniye 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. İşik 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. [Kochenderfer 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. Bauman; 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.).

### 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>

### 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://orcid.org/0000-0002-9801-7454>