

УДК 004.021

Doi: 10.31772/2712-8970-2022-23-2-168-176

**Для цитирования:** Крутько Д. А., Калашников А. С., Буряченко В. В. Методы построения маршрутов вне населенных пунктов на основе GPS-данных // Сибирский аэрокосмический журнал. 2022. Т. 23, № 2. С. 168–176. Doi: 10.31772/2712-8970-2022-23-2-168-176.

**For citation:** Krutko D. A., Kalashnikov A. S., Buryachenko V. V. [Methods for constructing routes outside of settlements on the basis of GPS data]. *Siberian Aerospace Journal*. 2022, Vol. 23, No. 2, P. 168–176. Doi: 10.31772/2712-8970-2022-23-2-168-176.

## Методы построения маршрутов вне населенных пунктов на основе GPS-данных

Д. А. Крутько\*, А. С. Калашников, В. В. Буряченко

Сибирский государственный университет науки и технологий имени академика М. Ф. Решетнева  
Российская Федерация, 660037, г. Красноярск, просп. им. газ. «Красноярский Рабочий», 31  
\*E-mail: krutko.d00@gmail.com

*Методы построения маршрутов включают задачу поиска кратчайшей траектории между двумя или несколькими объектами, которая может изменяться в зависимости от погодных условий, координат высоты и других параметров. Методы, которые рассматриваются в статье, позволяют выполнять построение маршрутов с использованием GPS-треков для различных областей знаний: проектирование маршрутов в рамках города, региона, страны либо при дистанционном зондировании земли. Рассматриваемые алгоритмы используются в сфере мониторинга окружающей среды при чрезвычайных ситуациях, для поиска оптимальных маршрутов передачи данных в спутниковых системах и их валидации, а также в организационно-экономических системах. Наиболее широко для построения маршрутов применяются подходы теории графов и поиска в пространстве состояний, где любой траектории между объектами ставится свой вес. Однако до сих пор не существует универсальной системы, позволяющей построить оптимальный маршрут по пересеченной местности. В статье рассмотрены такие методы, как алгоритм Дейкстры, Левита, Флойда – Уоршелла, а также выполнено сравнение их эффективности по времени работы и вычислительной сложности. Целью является разработка алгоритма поиска кратчайшего пути и построения туристического маршрута от заданной точки А до точки Б, что откроет большие возможности для горожан самостоятельно посещать новые интересные районы, активно проводить свободное время и узнавать окрестности города. Система апробирована на территории Торгашинского хребта, включает более 38 точек маршрута, расположенных на расстоянии более 25 км, и позволяет построить желаемые маршруты в течение менее 15 мс с учетом перепада весов и расстояния между объектами. При этом система допускает ввод собственных координат, которые учитываются при построении маршрутов.*

*Ключевые слова:* кратчайший путь, теория графов, построение маршрутов, gpx-треки, алгоритм Дейкстры.

## Methods for constructing routes outside of settlements on the basis of GPS data

D. A. Krutko\*, A. S. Kalashnikov, V. V. Buryachenko

Reshetnev Siberian State University of Science and Technology  
31, Krasnoyarskii rabochii prospekt, Krasnoyarsk, 660037, Russian Federation  
\*E-mail: krutko.d00@gmail.com

*Route building methods include the task of finding the shortest trajectory between two or more objects, which may vary depending on weather conditions, altitude coordinates, and other parameters. The methods discussed in the article allow building routes using GPS tracks for various fields of knowledge: designing routes within a city, region, country, or with remote sensing of the earth. The considered algorithms are used in the field of environmental monitoring in emergency situations, to search for optimal data transmission routes in satellite systems and their validation, as well as in organizational and economic systems. The most widely used approaches for building routes are graph theory and search in the state space, where any trajectory between objects is given its own weight. However, there is still no system that allows you to make a tourist route over rough terrain. The most widely used approaches are graph theory and search in the state space, where any trajectory between objects has its own weight. The article discusses such methods as the Dijkstra, Levitt, Floyd-Warshell algorithm, and also compares their effectiveness in terms of running time and complexity. The aim of the work is to develop an algorithm for finding the shortest path and building a tourist route from a given point A to point B. This development will open up new opportunities for citizens to independently visit new interesting areas, actively spend their free time and get to know the surroundings of the city. The system has been tested on the territory of the Torgashinsky ridge, includes more than 38 route points located at a distance of more than 25 kilometers, and allows you to build the desired routes within less than 15 milliseconds. At the same time, the system allows you to enter your own coordinates, which are considered when building routes.*

*Keywords: shortest path, graph theory, building routes, gpx-tracks, Dijkstra's algorithm.*

## **Введение**

На сегодняшний день актуальна тема поддержания здорового образа жизни. Люди стремятся следить за своим здоровьем, занимаются спортом, а также путешествуют. Они постоянно находятся в поиске различных мест посещения. Красноярский край обладает очень богатой природой. В нашем городе существует проект «Красноярский Хайкинг», который представляет собой сеть промаркированных троп на территории Торгашинского хребта. Появилось множество граждан, желающих сходить в поход как в окрестностях своего города, так и вне городской местности. Существует множество источников, где можно найти справочную информацию о местах посещения и общие тропы, однако, до сих пор нет возможности создания своего индивидуального маршрута по пересеченной местности с оптимальным для конкретного человека треком. Для разработки такой системы, которая бы позволила в автоматизированном режиме строить свои собственные маршруты по выбранным местам посещения с использованием данных gps-навигации, необходимо разработать собственный алгоритм, позволяющий построить кратчайший маршрута от точки А до точки Б.

Задача построения туристического маршрута сводится к задаче поиска кратчайшего пути от одной вершины графа до другой [1]. Данная проблема является одной из самых популярных задач в теории графов. Граф представляет собой абстрактный объект из множества вершин (узлов) и набора ребер, описывающих связи между парами вершин. Маршруты, которые состоят из связанных между собой перекрестков, являются графом.

Методы построения маршрутов применяются на практике в различных сферах. В работе [2] рассмотрены подходы, связанные с разработкой технологии для комплексной обработки данных дистанционного зондирования и векторных карт с целью мониторинга последствий чрезвычайных ситуаций, выявления критичных территорий и предотвращения негативных последствий. Авторы [3] рассматривают применение дискретного алгоритма решения SOTA (Self-organizing Tree Algorithm) и его параллельной версии с применением CUDA при построении надежного кратчайшего пути с использованием возможностей видеокарт. В работе [4] авторами разработана модификация дискретного алгоритма нахождения маршрута движения в зависящей от времени транспортной сети. Авторы демонстрируют эффективность предлагаемого подхода на примере крупномасштабной дорожной сети г. Самары и достигают скорости построения эффективного маршрута менее 1 с. Также методы построения маршрутов применяются для пере-

дачи данных в спутниковых [5], организационно-экономических системах [6], для мониторинга состояния лесов при организации спутникового наблюдения [7; 8] и в других областях данных. В статье рассмотрены различные методы построения маршрутов и предложена эффективная реализация модификации метода Дейкстры для построения туристических маршрутов на территории Торгашинского хребта.

### Методы построения маршрутов

Классическая проблема поиска кратчайшего пути в открытой местности была направлением большого числа исследований на протяжении многих лет. В связи с этим существует большое количество различных алгоритмов нахождения пути, многие из которых показывают хорошие результаты в своей области. Исследователи в работе [9] выполнили оценку эффективности 15 алгоритмов поиска кратчайшего пути в реальных дорожных сетях. Авторы рассматривают алгоритмы Беллмана – Форда – Мура, алгоритм Дейкстры и его модификации, алгоритм Пейпа – Левитта и другие. На основе оценки определен набор рекомендуемых алгоритмов для вычисления кратчайших путей в реальных дорожных сетях. Также широко известен алгоритм  $A^*$  [10], который направлен на уменьшение времени поиска оптимального маршрута за счет исключения менее перспективных направлений поиска на базе алгоритма Дейкстры. Модификации алгоритма Дейкстры применяются во многих практических задачах, например поиск кратчайшего маршрута для туризма в Бали [11]. Рассмотрим наиболее подробно такие алгоритмы, как алгоритм Дейкстры, алгоритм Левита и алгоритм Флойда – Уоршелла.

Алгоритм Дейкстры – один из самых известных алгоритмов для поиска кратчайшего пути [12]. Он позволяет определить кратчайшие пути между вершинами. Реализация заключается в том, что алгоритм на каждом шаге «посещает» одну вершину и пытается уменьшить метки. Работа алгоритма завершается, когда все вершины посещены. У алгоритма Дейкстры имеется ряд достоинств, таких как высокая скорость работы и точность результата. Однако есть и недостаток – сложность в понимании. Вычислительная сложность алгоритма Дейкстры зависит от способа нахождения вершины, хранения множества непосещенных вершин и обновления меток. Отсюда получаем, что реализация в данном методе потребует  $O(N)$  и  $O(1)$  единиц соответственно. Учитывая, что первая операция выполняется  $N$  раз, а вторая в зависимости от построенного графа, получается сложность  $O(N \times N + M)$ , где  $N$  – количество вершин, а  $M$  – константа, зависящая от построенного графа.

Алгоритм Левита – алгоритм на графах, который находит кратчайшее расстояние от одной из вершин графа до всех остальных [13]. Он также работает для графов с ребрами отрицательного веса. В сравнении с методом Дейкстры метод Левита проигрывает в том, что некоторые вершины приходится обрабатывать повторно, а выигрывает на более простых алгоритмах включения и исключения вершин из множества  $M1$  ( $M1$  – вершины, расстояние до которых вычисляется на текущем шаге алгоритма). Установлено, что для графов с «геометрическим» происхождением, построенных на основе транспортных сетей и реальных расстояний, метод Левита оказывается наиболее быстрым. Помимо этого, он выигрывает и по размеру программы. Сложность алгоритма Левита в худшем случае составляет  $O(N^2 \times M)$ . Чтобы достичь такого времени работы необходимо, чтобы в графе ребра располагались в лексикографическом порядке. Более реальной оценкой данного метода является среднее время, а именно сложность  $O(N \times M)$ . Однако на реальных графах алгоритм Левита лишь немногим уступает алгоритму Дейкстры.

Алгоритм Флойда – Уоршелла используется для нахождения кратчайших расстояний между всеми вершинами взвешенного графа без циклов с отрицательными весами с использованием метода динамического программирования [14]. В алгоритме используются две матрицы смежности: матрица расстояний  $D_k$  и матрица предшествования  $S_k$ , после чего, в течение  $n$  итераций, где  $n$  – количество узлов в матрице расстояний, а  $n$ -я итерация, дается

оптимальная / конечная матрица расстояний  $D_k = n$ , а также окончательная матрица предшествования  $S_k = n$ .

Недостаток алгоритма Флойда – Уоршелла в том, что алгоритм определяет только кратчайшее расстояние между всеми парами вершин, но не сохраняет информацию о кратчайших путях, что необходимо для задач построения маршрута.

По итогу рассмотрения наиболее важных алгоритмов для получения кратчайшего пути в графе, а также анализу их преимуществ и недостатков, обозначим требования, которые необходимы для реализации поиска кратчайшего пути в трехмерном пространстве на территории Торгашинского хребта:

- временная сложность работы алгоритма;
- высокая простота реализации алгоритма для мобильного приложения;
- работа для графов с положительными весами;
- точность результата;
- сохранение информации о кратчайших путях.

Проведенный анализ показал, что для поиска кратчайшего пути в трехмерном пространстве на территории Торгашинского хребта целесообразно использовать алгоритм Дейкстры, так как он учитывает особенности рассматриваемого процесса. Основными достоинствами алгоритма Дейкстры является высокая скорость работы и точность результата [15].

### Алгоритм построения маршрутов на основе GPS-данных

Прежде чем переходить к пошаговому описанию, рассмотрим схему работы алгоритма построения маршрутов по пересеченной местности на территории Торгашинского хребта (рис. 1).

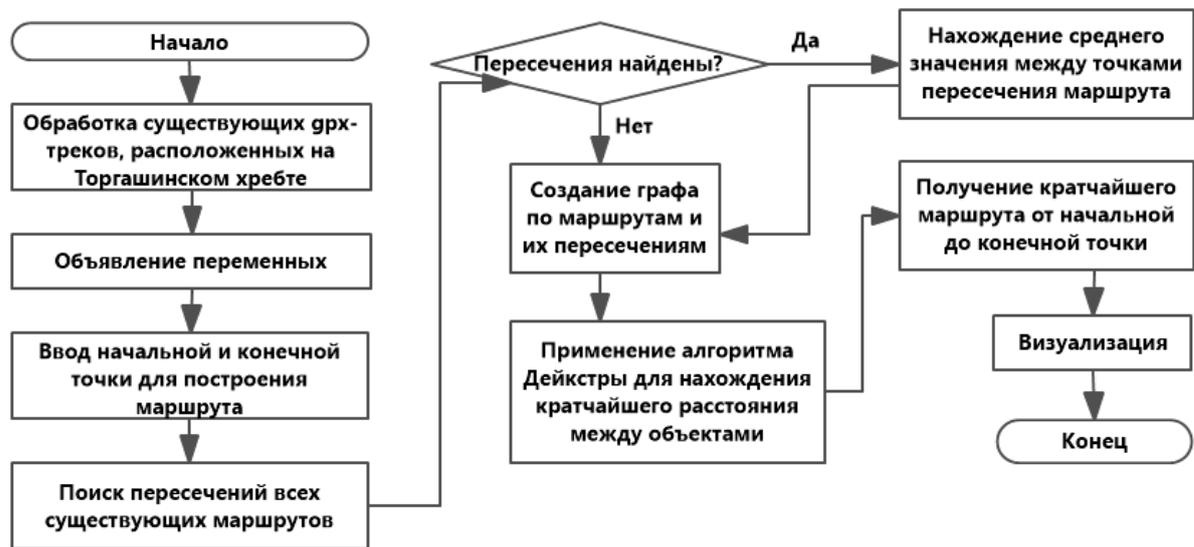


Рис. 1. Схема разработанного алгоритма построения оптимального пути

Fig. 1. Scheme of the developed algorithm for constructing various paths

В программной реализации существует двенадцать грх-треков (свободный текстовый формат хранения и обмена данными GPS) уже существующих маршрутов на территории Торгашинского хребта. Для того чтобы извлечь всю необходимую информацию из данных треков, применяется библиотека minidom, которая позволяет проанализировать xml-разметку и выбрать необходимую информацию [16].

Все треки находятся в соответствующей папке tracks. При проходе циклом по этой папке, выполняется посещение каждого маршрута и в соответствующие поля записывается следующая информация: название маршрута, цвет прорисовки и координаты каждой точки маршрута в формате широта, долгота и высота.

В словаре *vocab* ключам в виде названия маршрута задаются соответствующие *grx*-треки. Далее по этому словарю формируется список названий маршрутов и создаются словари: индексы для точек пересечения и графа (*distances*), список вершин (*nodes*). Короткими задаются начальная (*st*) и конечная (*en*) точки маршрута для составления нового маршрута – они состоят из координат по *y* и *x* – затем они записываются в словарь *point*.

При проходе циклом по *vocab*, выполняется проверка условия, являются ли точки *st* и *en* начальными или конечными точками существующих маршрутов и в таком случае они удаляются из списка *point* – этот список позволяет определить требуется ли создавать новые вершины графа для корректной работы алгоритма Дейкстры.

Следующим этапом является создание графа маршрутов и их пересечений. Для этого необходимо соотнести каждый маршрут со всеми остальными, чтобы определить есть ли пересечения, в случае нахождения пересечений зафиксировать их. Создаются списки по каждому маршруту, содержащие соответственно  $x_1$ ,  $y_1$  и  $z_1$  координаты для первого маршрута и  $x_2$ ,  $y_2$  и  $z_2$  координаты второго маршрута. Проходя циклом по точкам списков  $x_1$  и  $x_2$ , определяется расстояние между точкам. Если это расстояние меньше минимального, то перезаписывается минимальное расстояние на новое и сохраняются индексы найденных точек. Если же маршруты пересекаются, определяется среднеарифметическая точка пересечения ( $x$ ,  $y$ ,  $z$ ). Данный метод позволяет найти точку пересечения маршрута, если *grx*-треки не пересекаются, а проходят очень близко друг к другу.

Для нахождения расстояния между двумя точками необходимо изучить форму Земли. Форма Земли может быть описана как сфера, поэтому уравнения для вычисления расстояний на большом круге важны для вычисления кратчайшего расстояния между точками на поверхности Земли и часто используются в навигации [17].

Вычисление расстояния этим методом более эффективно и во многих случаях более точно, чем вычисление его для спроектированных координат, поскольку, во-первых, для этого не надо переводить географические координаты в прямоугольную систему координат и, во-вторых, многие проекции, если неправильно выбраны, могут привести к значительным искажениям длин в силу особенностей проекционных искажений.

Для вычислений используется сфера радиусом 6372795 м, что может привести к меньшей ошибке, чем если считать в прямоугольных системах координат.

Согласно сферической теореме косинусов:

$$\Delta\sigma = \sin^{-1} \{ \sin \phi_1 \times \sin \phi_2 + \cos \phi_1 \times \cos \phi_2 \times \cos \Delta\lambda \}, \quad (1)$$

где  $\phi_1$ ,  $\lambda_1$ ,  $\phi_2$ ,  $\lambda_2$  – широта и долгота двух точек в радианах;  $\Delta\lambda$  – разница координат по долготе;  $\Delta\sigma$  – угловая разница.

Для перевода углового расстояния в метрическое, необходимо угловую разницу умножить на радиус Земли, единицы конечного расстояния будут равны единицам, в которых выражены метры.

Описанный метод реализован в функциях: *Lenfor* и *Meters*. Функция *Lenfor* – функция, рассчитывающая длину пути между точками маршрута по *x* и *y* без учета высоты, и функция *Meters* – функция, рассчитывающая расстояние между точками *gps*-координат с учетом перепада высот, после вычисления расстояния функцией *Lenfor*.

Алгоритм Дейкстры позволяет определить кратчайшие пути между вершинами. В данной реализации принцип этого алгоритма используется для определения весов ребер, нахождения кратчайшего пути и составляет его из частей *grx*-треков. Алгоритм Дейкстры поэтапно формулируется следующим образом:

Шаг 1: задать начальную и конечную точку маршрута.

Шаг 2: создать словарь непосещенных вершин – *unvisited*, словарь путей до каждой вершины – *way*, словарь посещенных вершин – *visited*.

Шаг 3: цикл, пока есть непосещенные вершины. Иначе шаг 10.

Шаг 4: если есть соседи в `unvisited`, то новый вес пути (`newDistance`) = текущий вес пути (`currentDistance`) + вес между вершинами (`distance`). Иначе шаг 3.

Шаг 5: значение `unvisited` – список? Если да, взять первое значение списка: `dist = dist[0]`. Иначе `unvisited[neighbour] = newDistance`, `newPath` (новая часть пути).

Шаг 6: записать в словарь путей `newPath`. А в `visited` для текущей вершины записать `currentDistance`.

Шаг 7: удалить текущую вершину из `unvisited`.

Шаг 8: сформировать кандидатов для новой текущей вершины. Отсортировать `newDistance` по возрастанию. Получаем новую текущую вершину (`path`) = `currentDistance[1]` и текущий путь до нее (`currentDistance`) = `currentDistance[0]`.

Шаг 9: `newPath = path + current` (текущая ячейка).

Шаг 10: получен заполненный словарь путей из стартовой точки.

Шаг 11: выбрать искомый маршрут по ключу конечной точки.

Шаг 11: определить какому маршруту принадлежит очередная пара вершин.

Шаг 12: если индекс начальной точки (`lost`) > индекса конечной точки (`next`), копировать эту часть трека. Иначе, идти в обратную сторону трека.

Шаг 13: собрать оптимальный `gps`-трек, по заданным точкам.

### Экспериментальные исследования

В качестве примера рассмотрим один из вариантов построения маршрута, где начальная точка – гора Тамара с координатами 55.952562 и 92.857231, находящаяся на маршруте «Здоровье», а конечная точка – 2-я Торгашинская видовка с координатами 55.919239 и 92.889939, находящаяся на маршруте «Болгаш». Рассмотрим визуализацию маршрута на рис. 2 (отображен розовым цветом).



Рис. 2. Визуализация построенного маршрута

Fig. 2. Visualization of the constructed route

На рис. 2 С.Ш. – северная широта, а В.Д. – восточная долгота. Наблюдаем 13 маршрутов разных цветов, расшифруем слева направо: красный – Тропа «Здоровье», фиолетовый – Тропа «Лыжная», зеленый – тропа «Мокрый лог», розовый – построенный маршрут по указанным точкам, оранжевый – тропа «Рыжая», зеленый – тропа «Синильга», желтый – тропа «Сквозная», синий – тропа «Сивая», фиолетовый – тропа «Топ», оранжевый – тропа «Болгаш», коричневый – тропа Спелеологов, фиолетовый – тропа «Кузнецово», синий – тропа «Сказка».

Помимо визуализации, алгоритм построения туристических маршрутов вычисляет длину нового маршрута, максимальную и минимальную высоту на протяжении всего трека и время построения. Таким образом, по построенному маршруту получено: длина маршрута = 10676 м, максимальная высота = 584,53 и минимальная высота = 179,76. На текущий момент в программной системе имеется 38 различных объектов Торгашинского хребта, что охватывает более 95 % точек посещения туристами. При этом алгоритм (система) допускает ввод собственных координат, которые учитываются при построении маршрутов. В таблице приведены характеристики некоторых построенных маршрутов.

#### Примеры построенных маршрутов и их характеристики

	Время построения, с	Расстояние, м	Перепад высот, м
От г. Тамара до 2-й Торгашинской видовки	0,005457	10 676	404,77
От ск. Сивые до Черной сопки	0,005655	21 642	376,68
От г. Синильга до г. Луч	0,004734	10 555	404,77
От СНТ «Загорье-1» до пещ. Мокрая	0,005604	6 371	381,67
От ск. Пропасть до ск. Красный гребень	0,013433	4 188	229,81
От ск. Магда до СНТ «Загорье-2»	0,003422	9 849	266,07

#### Заключение

В результате работы разработан алгоритм построения кратчайшего маршрута от начальной до конечной точки, основанный на алгоритме Дейкстры, позволяющий получить оптимальный грх-трек, а также имеющий базу существующих маршрутов. Алгоритм применяется в программной системе, реализованной в виде web-сайта и мобильного приложения, который позволяет туристам построить удобные маршруты для посещения Торгашинского хребта. В системе имеется 38 туристических объектов, и построение любого маршрута занимает в среднем 15 мс.

#### Библиографические ссылки

1. Крутько Д. А. Проблема автоматизации построения схем маршрутов пешего туризма в горных массивах Красноярского края // Актуальные проблемы авиации и космонавтики : материалы VII Междунар. науч.-практ. конф. Красноярск, 2021. Т. 2. С. 260–262.
2. Иванова К. А., Маликова О. В. Технология комплексной обработки геопространственных данных для мониторинга последствий чрезвычайных ситуаций // Региональные проблемы дистанционного зондирования Земли : материалы II Междунар. науч. конф. (22–25 сентября 2015, г. Красноярск) / науч. ред. Е. А. Ваганов ; отв. ред. М. В. Носков. Красноярск : Сиб. федер. ун-т, 2015. С. 62–65.
3. Агафонов А. А., Максимов А. И., Бородинов А. А. Исследование эффективности вычисления надежного кратчайшего пути с использованием GPU // VI Междунар. конф. и молодёжная школа «Информационные технологии и нанотехнологии» (ИТНТ-2020), 2020. С. 1–7.
4. Агафонов А. А., Мясников В. В. Метод определения надежного кратчайшего пути в стохастической сети с использованием параметрически заданных устойчивых распределений вероятностей // СПИИРАН. 2020. Т. 3, вып. 18. С. 558–582.
5. Федоров А. А., Сошилов И. В., Логинов В. Н. Эвристические алгоритмы поиска маршрутов передачи данных в спутниковых системах и их валидация // ТРУДЫ МФТИ. 2020. Т. 12, № 3. С. 1–9.
6. Рамзаев В. М., Хаймович И. Н., Мартынов И. В. Методы поиска кратчайших путей на графах в организационно-экономических системах и их реализация // V Междунар. конф. и молодёжная школа «Информационные технологии и нанотехнологии» (ИТНТ-2019), 2019. С. 1–8.
7. Стыщенко Ф. В., Сайгин И. А., Барталев С. А. Исследование возможностей многолетнего мониторинга состояния поврежденных пожарами лесов на основе спутниковых данных // Ин-

формационные технологии в дистанционном зондировании Земли – RORSE 2018. ИКИ РАН, 2019. С. 185–190. Doi: 10.21046/rorse2018.185.

8. Миклашевич Т. С., Барталев С. А., Плотников Д. Е. Интерполяционный алгоритм восстановления длинных временных рядов данных спутниковых наблюдений растительного покрова // Современные проблемы дистанционного зондирования Земли из космоса. 2019. Т. 16. № 6. С. 1–10.

9. Zhen B., Noon C. Shortest Path Algorithms: An Evaluation using Real Road Networks Transportation Science. 1998. P. 1–10.

10. Algorithm to find tourism place shortest route: a systematic literature review / E. Madyatmadja, H. Nindito, R. Bhaskoro et. al. // Journal of Theoretical and Applied Information Technology. 2021. Vol. 99, No. 4. P. 1–10.

11. Fitriansyah A., Parwati N., Wardhani D., Kustian N. Dijkstra’s Algorithm to Find Shortest Path of Tourist Destination in Bali ICASMI, 2018.

12. Базовые алгоритмы нахождения кратчайших путей во взвешенных графах [Электронный ресурс]. URL: <https://habr.com/ru/post/119158/>.

13. Алгоритм Левита – алгоритмы поиска на графах [Электронный ресурс]. URL: <https://amp.www.google-info.org/3957083/1/algorithm-levita.html>.

14. A Study on Different Algorithms for Shortest Route Problem / Dr. Roopa, R. Apoorva, H. Srinivasu, M. Viswanatah // International Journal of Engineering Research & Technology (IJERT). 2013. Vol. 2, Iss. 9. P. 1–13.

15. Крутько Д. А. Проблема поиска кратчайшего пути в трехмерном пространстве на территории Торгашинского хребта // Актуальные проблемы авиации и космонавтики : материалы VIII Междунар. науч.-практ. конф. Красноярск, 2022.

16. xml.dom.minidom – Minimal DOM implementation [Электронный ресурс]. URL: <https://docs.python.org/3/library/xml.dom.minidom.html>.

17. Вычисление расстояния и начального азимута между двумя точками на сфере [Электронный ресурс]. URL: <https://gis-lab.info/qa/great-circles.html>.

## References

1. Krutko D. A. [The problem of automating the construction of hiking route schemes in the mountains of the Krasnoyarsk Territory]. *Proceedings of the VII International Scientific and Practical Conference “Actual Problems of Aviation and Cosmonautics”*. Krasnoyarsk, 2021, Vol. 2, P. 260–262 (In Russ.).

2. Ivanova K. A., Malikova O. V. [Technology of integrated processing of geospatial data for monitoring the consequences of emergency situations]. *Regional problems of remote sensing of the Earth: materials of the II Intern. scientific conference*. September 22–25, 2015. Krasnoyarsk, Sib. feder. un-t Publ., 2015, P. 62–65 (In Russ.).

3. Agafonov A. A., Maksimov A. I., Borodinov A. A. [Study of the efficiency of calculating a reliable shortest path using GPU]. *VI International Conference and Youth School “Information Technologies and Nanotechnologies” (ITNT-2020)*. 2020, P. 1–7 (In Russ.).

4. Agafonov A. A., Myasnikov V. V. [A method for determining a reliable shortest path in a stochastic network using parametrically specified stable probability distributions]. *SPIIRAN*. 2020. Vol. 3, Iss. 18, P. 558–582 (In Russ.).

5. Fedorov A. A., Soshilov I. V., Loginov V. N. [Heuristic algorithms for searching for data transmission routes in satellite systems and their validation]. *Proceedings OF MIPT*. 2020, Vol. 12, No. 3, P. 1–9 (In Russ.).

6. Ramzaev V. M., Khaimovich I. N., Martynov I. V. [Methods for finding shortest paths on graphs in organizational and economic systems and their implementation]. *V International Conference and Youth School “Information Technologies and Nanotechnologies” (ITNT-2019)*. 2019, P. 1–8 (In Russ.).

7. Stytsenko F. V., Saigin I. A., Bartalev S. A. [Study of the possibilities of long-term monitoring of the state of forests damaged by fires based on satellite data]. *Information technologies in remote sensing of the Earth. RORSE 2018*. IKI RAS, 2019, P. 185–190. Doi: 10.21046/rorse2018.185.
8. Miklashevich T. S., Bartalev S. A., Plotnikov D. E. [Interpolation algorithm for restoring long time series of satellite observations of vegetation cover]. *Modern problems of remote sensing of the Earth from space*. 2019, Vol. 16, P. 1–10 (In Russ.).
9. Zhen B., Noon C. Shortest Path Algorithms: An Evaluation using Real Road Networks Transportation Science, 1998, P. 1–10,
10. Madyatmadja E., Nindito H., Bhaskoro R. et. al. Algorithm to find tourism place shortest route: a systematic literature review. *Journal of Theoretical and Applied Information Technology*. 2021. Vol. 99, No. 4, P. 1–10.
11. Fitriansyah A., Parwati N., Wardhani D., Kustian N. Dijkstra's Algorithm to Find Shortest Path of Tourist Destination in Bali ICASMI, 2018.
12. Basic algorithms for finding shortest paths in weighted graphs. Available at: <https://habr.com/ru/post/119158/>.
13. Levit's algorithm – search algorithms on graphs. Available at: <https://amp.www.google-info.org/3957083/1/algorithm-levita.html>.
14. Roopa Dr., Apoorva R., Srinivasu H., Viswanatah M. A Study on Different Algorithms for Shortest Route Problem International. *Journal of Engineering Research & Technology (IJERT)*. 2013, Vol. 2, Iss. 9, P. 1–13.
15. Krutko D. A. [The problem of finding the shortest path in three-dimensional space on the territory of the Torgashinsky ridge]. *Materials of the VIII International Scientific and Practical Conference "Actual problems of aviation and cosmonautics"*. Krasnoyarsk, 2022 (In Russ.).
16. xml.dom.minidom – Minimal DOM implementation. Available at: <https://docs.python.org/3/library/xml.dom.minidom.html>.
17. Calculation of the distance and initial azimuth between two points on the sphere. Available at: <https://gis-lab.info/qa/great-circles.html>.

© Крутько Д. А., Калашников А. С., Буряченко В. В., 2022

---

**Крутько Диана Андреевна** – студентка группы БПИ18-01; Сибирский государственный университет науки и технологий имени академика М. Ф. Решетнева. E-mail: [krutko.d00@gmail.com](mailto:krutko.d00@gmail.com).

**Калашников Алексей Сергеевич** – студент группы А17-02; Сибирский государственный университет науки и технологий имени академика М. Ф. Решетнева. E-mail: [fangy.ko@gmail.com](mailto:fangy.ko@gmail.com).

**Буряченко Владимир Викторович** – кандидат технических наук, доцент; Сибирский государственный университет науки и технологий имени академика М. Ф. Решетнева. E-mail: [buryachenko@sibsau.ru](mailto:buryachenko@sibsau.ru).

**Krutko Diana Andreevna** – student of the BPI18-01 group; Reshetnev Siberian State University of Science and Technology. E-mail: [krutko.d00@gmail.com](mailto:krutko.d00@gmail.com).

**Kalashnikov Aleksey Sergeevich** – student of group A17-02; Reshetnev Siberian State University of Science and Technology. E-mail: [fangy.ko@gmail.com](mailto:fangy.ko@gmail.com).

**Buryachenko Vladimir Viktorovich** – Cand. Sc., Associate Professor; Reshetnev Siberian State University of Science a Technology. E-mail: [buryachenko@sibsau.ru](mailto:buryachenko@sibsau.ru).

---