

МЕТОДЫ ОПТИМИЗАЦИИ | OPTIMIZATION METHODS

Научная статья | Original paper

УДК 004.021:004.032.26

Программный комплекс нахождения оптимального кусочно-прямолинейного маршрута с n поворотами

В.Н. Нефедов ✉, Г.К. Наседкин

Московский авиационный институт

(национальный исследовательский университет) (МАИ)

Москва, Российская Федерация

✉ nefedovvn54@yandex.ru

Резюме

Контекст и актуальность. Существуют практически значимые задачи, в которых требуется соединить две заданные точки плоскости кусочно-прямолинейной ломаной. При этом возможны различные ограничения на искомую ломаную, в числе которых может быть ограничение на количество звеньев n и на абсолютную величину углов поворота в точках излома. В простейшем дискретном случае точки поворота принадлежат конечному множеству Q , содержащему N элементов. При этом полный перебор всех возможных решений представляет собой NP-трудную задачу и может потребовать перебора N^n вариантов. Между тем, возможны подходы, позволяющие значительно сузить множество допустимых решений. Одним из таких подходов является использование точной формулы, описывающей множество всех ломаных (описываемых в виде последовательности n точек поворота), удовлетворяющих заданному ограничению на абсолютную величину углов поворота. В этом случае происходит последовательное значительное сужение допустимой области для каждого следующего поворота. В результате алгоритм перебора элементов именно такого множества будет работать многократно быстрее алгоритма полного перебора. **Цель.** Реализовать алгоритм перебора множества допустимых решений согласно точной формуле. **Гипотеза.** Алгоритм перебора элементов именно такого множества будет работать многократно быстрее алгоритма полного перебора. **Методы и материалы.** Программный комплекс, в котором реализован алгоритм поиска, разработан



для работы с фигурами, представляющими собой карту трудоёмкости маршрута. После выставления ограничений на угол и указания количества поворотов искомого маршрута, программный комплекс применяет алгоритм перебора множества допустимых решений с использованием точной формулы (полученной с использованием метода динамического программирования). **Результаты.** Проведены эксперименты на четырёх картах при количествах поворотов, равных 2, 3 и 4. Алгоритм перебора элементов множества допустимых маршрутов показал существенное преимущество по сравнению с алгоритмом простого многократного (n -кратного) перебора элементов множества Q . Экспериментально установлена кубическая зависимость времени работы алгоритма от максимального угла поворота. **Выводы.** Разработанный алгоритм многократно превосходит метод простого перебора по скорости работы и позволяет решать многие задачи за приемлемое время.

Ключевые слова: оптимизация, кусочно-прямолинейный маршрут, ломаная, угол поворота, программный комплекс

Для цитирования: Нефедов, В.Н., Наседкин, Г.К. (2026). Программный комплекс нахождения оптимального кусочно-прямолинейного маршрута с n поворотами. *Моделирование и анализ данных*, 16(3), 155–168. <https://doi.org/10.17759/mda.2026160307>

A software package for finding an optimal piecewise rectilinear route with n turns

V.N. Nefedov ✉, G.K. Nasedkin

Moscow Aviation Institute (National Research University) (MAI)

Moscow, Russian Federation

✉ nefedovvn54@yandex.ru

Abstract

Context and relevance. There are practically significant problems requiring connecting two given points in a plane with a piecewise polygonal line. Various constraints on the desired polyline are possible, including a limit on the number of links n and on the absolute value of the rotation angles at the breakpoints. In the simplest discrete case, the turning points belong to a finite set Q containing N elements. A complete enumeration of all possible solutions is an NP-hard problem and may require searching through N^n variants. However, approaches are possible that significantly narrow the set of feasible solutions. One such approach is to use an exact formula describing the set of all polylines (described as a sequence of n turning points) that satisfy a given constraint on the absolute value of the rotation angles. In this case, the feasible domain is successively significantly narrowed for each subsequent rotation. As a result, an algorithm for enumerating the elements



of precisely such a set will perform many times faster than an exhaustive search algorithm. **Objective.** Implement an algorithm for enumerating a set of feasible solutions according to a precise formula. **Hypothesis.** To implement an algorithm that enumerates the set of feasible solutions according to the exact formula. **Methods and materials.** The software package that implements the search algorithm was developed to work with images representing a route complexity map. After setting the angle constraint and specifying the required number of turns, the package applies an enumeration algorithm over the set of feasible solutions using the exact formula obtained by dynamic programming. **Results.** Experiments were conducted on four maps with the number of turns equal to 2, 3, and 4. The algorithm enumerating the set of feasible routes showed a significant advantage over simple n -fold enumeration of the elements of Q . A cubic dependence of the algorithm's running time on the maximum turning angle was experimentally established. **Conclusions.** The developed algorithm is many times faster than simple enumeration and allows solving a large number of problems in an acceptable time.

Keywords: optimization, piecewise-rectilinear route, broken line, turning angle, software package

For citation: Nefedov, V.N., Nasedkin, G.K. (2026). A software package for finding an optimal piecewise rectilinear route with n turns. *Modelling and Data Analysis*, 16(3), 155–168. (In Russ.). <https://doi.org/10.17759/mda.2026160307>

Введение

Существуют практически значимые задачи, в которых требуется соединить две заданные точки двумерной плоскости кусочно-прямолинейной ломаной. При этом имеются ограничения на количество звеньев ломаной (например, их ровно n штук) и на углы поворота в каждой точке, соединяющей соседние отрезки ломаной. Примеры таких задач описаны в (Свойкин, Королько и др., 2024) и (Королько, 2025). Решение задачи построения маршрута с минимальным количеством поворотом рассматривались в работах (Junjie Lu, Bi Zeng, Jingtang Tang, Tin Lun Lam, 2022), (Ivanov, 2025). В последнее время активно исследуются также численные методы оптимизации маршрутов с ограничениями на углы поворота (Петров, Михайлова, 2023), а также вопросы вычислительной сложности подобных задач при наличии геометрических ограничений (Hoffmann, Juan-Arinyo, 2022). Алгоритмы динамического программирования для кусочно-прямолинейной оптимизации траекторий с ограничениями на повороты предложены в (Garcia, Rodriguez, 2021). Фундаментальные аспекты планирования траекторий в присутствии угловых ограничений, включая анализ конфигурационного пространства и условия существования допустимых ломаных, подробно рассмотрены в (LaValle, 2006)

Требуется описать множество допустимых ломаных для того, чтобы иметь возможность их алгоритмически реализуемого перечисления в дискретном случае или аппроксимации конечным множеством ломаных в непрерывном случае



(с последующим их перечислением). Имея алгоритмы такого перечисления, можно затем решать задачи оптимизации некоторой целевой функции, в которой учитывается стоимость прохождения по каждому звену ломаной и стоимость совершения каждого поворота. Предполагается, что значения целевой функции достаточно легко вычисляются, исходя из имеющейся информации относительно области, на которой ищется оптимальная ломаная. Вычисление целевой функции на отрезке происходит в соответствии с алгоритмом описанном в (Нефедов, Свойкин и др., 2025). В данной работе используются два основных утверждения: Теорема 1 и Лемма 1 из (Нефедов, Наседкин, 2025), доказанные в (Нефедов, 2025), с помощью которых описывается интересующее нас множество допустимых ломаных.

Основной целью настоящей работы является исследование эффективности разработанного алгоритма по сравнению с алгоритмом полного перебора, а также анализ зависимости быстродействия алгоритма от значения максимального угла поворота.

Материалы и методы

Постановка задачи нахождения оптимального кусочно-прямолинейного маршрута

На целочисленном прямоугольнике $Q = \{0, 1, \dots, N-1\} \times \{0, 1, \dots, M-1\}$ заданы значения функции $F(x, y)$, где $(x, y) \in Q$. Функция $F(x, y)$ отражает трудоёмкость преодоления указанного участка области Q . Для удобства доопределим функцию $F(x, y)$ на координатном прямоугольнике

$$\bar{Q} = \{(x, y) \in \mathbb{R}^2 \mid 0 \leq x \leq N, 0 \leq y \leq M\}$$

следующим образом. Любая точка из Q соответствует левой нижней угловой точке одного из $N \cdot M$ квадратов, на которые множество Q разбивает прямоугольник $\bar{Q}$. Будем считать, что функция $F(x, y)$ является константной на этом квадрате, т.е.

$$\forall (x, y) \in Q, \forall (\bar{x}, \bar{y}) \in \{(u, v) \in \mathbb{R}^2 \mid x \leq u \leq x+1, y \leq v \leq y+1\} \quad F(\bar{x}, \bar{y}) = F(x, y).$$

Введём обозначение угла между векторами:

$$|\angle(A, B)| = \arccos\left(\frac{(A, B)}{|A| \cdot |B|}\right),$$

где (A, B) — скалярное произведение векторов A и B , а $|A|$ и $|B|$ — длины этих векторов. Считаем, что $\angle(A, B) \in (-\pi, \pi]$, положительное направление углов — по часовой стрелке.

Трудоёмкость пути на области определяется криволинейным интегралом по кусочно-прямолинейному маршруту

$$L = [B^{(0)}, B^{(1)}] \cup [B^{(1)}, B^{(2)}] \cup \dots \cup [B^{(n)}, B^{(n+1)}], \quad (1)$$



при этом

$$\begin{aligned} \angle(B^{(i+1)} - B^{(i)}, B^{(i)} - B^{(i-1)}) &\in [-\varphi_0, \varphi_0], i = 1, \dots, n, \\ B^{(i+1)} &\neq B^{(i)}, i = 0, 1, \dots, n. \end{aligned} \quad (2)$$

Функция трудоёмкости $f(L)$ выглядит следующим образом:

$$f(L) = \sum_{i=0}^n \int_{[B^{(i)}, B^{(i+1)}]} F(x, y) ds.$$

Обозначим $\mathcal{S}^{(n)}(A, B, \varphi)$ — множество всех возможных последовательностей $(B^{(1)}, \dots, B^{(n)})$ внутренних точек ломаной L , удовлетворяющей (1), (2), где $A = B^{(0)}$, $B = B^{(n+1)}$.

Необходимо найти оптимальный кусочно-прямолинейный маршрут L^* , соединяющий заданные две точки $A = B^{(0)}$ и $B = B^{(n+1)}$ и совершающий n поворотов, угол каждого не превосходит по абсолютной величине $\varphi_0 \in \left(0, \frac{\pi}{2}\right)$. Найти L^* можно в ходе решения следующей задачи:

$$f(L) \rightarrow \min; L \in \mathcal{S}^{(n)}(A, B, \varphi_0),$$

где минимум ищется на конечном множестве ломаных L вида (1), где $B^{(i)} \in Q, i = 0, 1, \dots, n+1$.

Отметим, что данная задача остаётся NP-трудной даже при сравнительно небольших n , что подтверждается анализом вычислительной сложности (Hoffmann, Juan-Arinyo, 2022); Sidorov, Kuznetsov, 2024). Тем не менее, как показано в (Yang, Sukkarieh, 2020), для практических приложений (например, планирование траекторий БПЛА) возможно достижение приемлемого времени работы за счёт применения эвристических сужений пространства поиска.

Описание множества допустимых маршрутов

Для решения поставленной выше задачи необходимо иметь удобное для практического использования описание множества допустимых маршрутов $\mathcal{S}^{(n)}(A, B, \varphi_0)$, которое будем использовать для построения алгоритма перебора его элементов. Для этого введём вспомогательные множества

$$\begin{aligned} \mathcal{S}(A, B, \varphi) &= \{C \in \mathbb{R}^2 \setminus \{A, B\} \mid \angle(B - C, C - A) \in (-\varphi, \varphi)\}, \\ \mathcal{C}(E, \varphi) &= \{C \in \mathbb{R}^2 \setminus \{(0, 0)\} \mid \angle(C, E) \in [-\varphi, \varphi]\}. \end{aligned}$$

В случае $\varphi \in (0, \pi)$ множество $\mathcal{S}(A, B, \varphi)$ представляет собой объединение двух сегментов окружностей с общей хордой $[A, B]$. Множество $\mathcal{C}(E, \varphi)$ при $\varphi \in (0, \pi/2)$ представляет собой выпуклый конус. Теперь мы можем выразить множество допустимых положений каждой точки поворота, а вместе с этим и множество возможных маршрутов $\mathcal{S}^{(n)}(A, B, \varphi)$ в самом общем случае, когда отсутствуют какие-либо ограничения на $B^{(i)}, i = 0, 1, \dots, n+1$:



$$\begin{aligned}
 & \left(B^{(1)}, \dots, B^{(n)} \right) \in S^{(n)}(A, B, \varphi) \Leftrightarrow \left(B^{(1)}, \dots, B^{(n)} \right) \in \\
 & \in S(A, B, n\varphi) \times \left\{ S(B^{(1)}, B, (n-1)\varphi) \cap \left[B^{(1)} + C(B^{(1)} - A, \varphi) \right] \right\} \times \\
 & \times \left\{ S(B^{(2)}, B, (n-2)\varphi) \cap \left[B^{(2)} + C(B^{(2)} - B^{(1)}, \varphi) \right] \right\} \times \dots \times \\
 & \times \left\{ S(B^{(n-2)}, B, 2\varphi) \cap \left[B^{(n-2)} + C(B^{(n-2)} - B^{(n-3)}, \varphi) \right] \right\} \times \\
 & \times \left\{ \left[\text{cl } S(B^{(n-1)}, B, \varphi) \setminus \{A, B\} \right] \cap \left[B^{(n-1)} + C(B^{(n-1)} - B^{(n-2)}, \varphi) \right] \right\}.
 \end{aligned} \tag{3}$$

Каждый i -й член декартового произведения (3) будем обозначать $S_i^{(n)}(A, B, \varphi)$.

Очевидно, что при наличии ограничения вида $A, B, B^{(1)}, \dots, B^{(n)} \in Q$ оно должно быть учтено в каждом члене прямого произведения (0.6), т.е. используем $S^{(n)}(Q, A, B, \varphi) = S^{(n)}(A, B, \varphi) \cap Q^n$ вместо $S^{(n)}(A, B, \varphi)$. С учетом сказанного при перечислении элементов множества $S^{(n)}(Q, A, B, \varphi)$ понадобятся алгоритмы, позволяющие перечислять элементы конечных множеств вида $S_i^{(n)}(Q, A, B, \varphi) = S_i^{(n)}(A, B, \varphi) \cap Q$.

Алгоритм построения маршрута

Простейший способ построения маршрута заключается в последовательном добавлении новых точек поворота $B^{(i)}$, $i = 1, 2, \dots, n$ с последующей проверкой ограничений (2). Точки $B^{(i)}$ перебираются из множества Q . Такой подход легко реализуем, но его временная сложность составляет $O((N \cdot M)^n)$.

Более эффективный подход состоит в использовании множества $S_i^{(n)}(A, B, \varphi)$ из формулы (3). При переборе элементов этого множества проверка ограничений не требуется, поскольку они уже учтены при его построении.

Предварительные вычисления и кэширование данных

Для ускорения работы программы перед основным циклом поиска выполняют предварительные вычисления величин, которые иначе пришлось бы многократно пересчитывать. Во-первых, значения синуса и тангенса фиксированного угла φ_0 вычисляются один раз и сохраняются в памяти. В ходе поиска производится обращение к этим значениям, что существенно быстрее непосредственного вычисления тригонометрических функций.

Во-вторых, значения функционала для последнего участка маршрута (от предпоследней точки поворота до конечной точки B) вычисляются заранее, поскольку они наиболее часто используются в процессе поиска. Полное хранение значений функционала для всех пар точек сетки невозможно из-за значительного объема требуемой памяти. Однако хранение значений функционала от всех точек сетки до фиксированной конечной точки B требует памяти, сопоставимой с размером самой сетки Q , что вполне реализуемо.

В-третьих, значение функционала для частично построенного маршрута сохраняется в процессе его построения. Благодаря этому при добавлении новой точки



поворота требуется вычислить только значение функционала для новообразованного прямолинейного участка маршрута.

Интерфейс программного комплекса

Программный комплекс предоставляет графический интерфейс для визуализации и настройки маршрута (рис. 1). Пользователь может добавлять или удалять точки поворота, перемещать их мышью, а также задавать максимальный допустимый угол поворота с помощью слайдера «Max Angle». Начальная, конечная и промежуточные точки маршрута отображаются разными цветами. В окне приложения выводятся координаты всех точек, измеренные углы поворота и текущее значение целевой функции.

В комплексе реализовано три алгоритма поиска:

1. Method 0 — полный перебор всех точек карты для каждой точки поворота (базовый алгоритм);
2. Method 1 — перебор с предварительным вычислением ограничивающего прямоугольника Q' допустимых положений;
3. Method 2 — разработанный алгоритм перебора множества допустимых маршрутов согласно формуле (3).

Значение функции трудоёмкости $F(x, y)$ на карте Q задаётся яркостью пикселя изображения и нормализуется к диапазону $[0, 1]$.



Рис. 1. Программный интерфейс

Fig. 1. Software interface

Результаты

В данном разделе представлены результаты работы программы на различных картах и с различными ограничениями. Также приведены результаты множественных экспериментов, проведённых с целью выяснения зависимости времени работы программы от величины максимального угла поворота.



Результаты исследования зависимости времени работы алгоритма от максимального угла поворота

Было проведено тестирование алгоритма на карте $Q = \{0, 1, \dots, 149\}^2$ с заданной на ней функцией

$$F(x, y) = \max \left(\frac{\sin \frac{x}{10} + \cos \frac{y}{10}}{8} + 0.5, e^{\frac{-2(x-75)^2 - 2(y-75)^2}{1000}} \right). \quad (4)$$

Заданы начальная точка $A = (32, 22)$, конечная $B = (122, 114)$ и максимальный угол в градусах $\varphi_0 = \{1, 2, \dots, 90\}$. Количество углов поворота $n = 2$, пример решения задачи при $\varphi_0 = \frac{\pi}{3}$ представлен на Рис. 1. График зависимости времени работы от максимального угла поворота представлен на Рис. 2. Для наглядности на графике также изображена аппроксимирующая данные функция. На графике время указано в секундах, а угол в радианах.

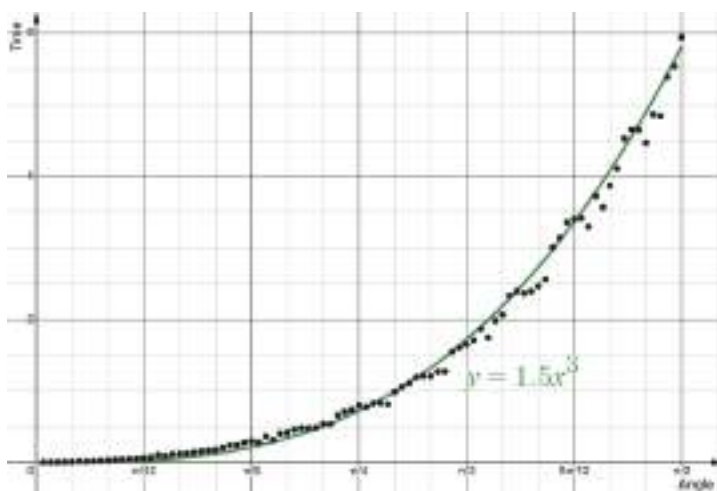


Рис. 2. График зависимости времени работы в секундах от максимального угла в радианах при $n = 2$

Fig. 2. Graph of the dependence of operating time in seconds on the maximum angle in radians for $n = 2$

Рассмотрим теперь зависимость времени работы алгоритма от величины максимального угла поворота при $n = 3$. Было проведено тестирование алгоритма на карте $Q = \{0, 1, \dots, 74\}^2$ с заданной на карте функцией $F(2x, 2y)$, $F(x, y)$ определена как (4).

Заданы начальная точка $A = (16, 11)$, конечная $B = (61, 57)$ и максимальный угол в градусах $\varphi_0 = \{1, 2, \dots, 90\}$. Пример решения представлен на Рис. 3.



График зависимости времени от максимального угла поворота с аппроксимирующей данные функцией представлен на Рис 4. На графике время указано в секундах, а угол в радианах.



Рис. 3. Пример решения при $n = 3, \varphi_0 = \frac{\pi}{2}$

Fig. 3. Example of a solution for $n = 3, \varphi_0 = \frac{\pi}{2}$

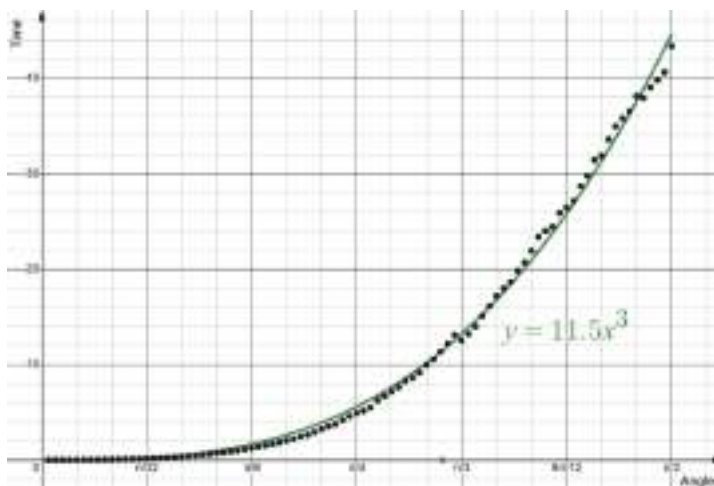


Рис. 4. График зависимости времени работы в секундах от максимального угла в радианах при $n = 3$

Fig. 4. Graph of the dependence of operating time in seconds on the maximum angle in radians for $n = 3$



Результаты исследования эффективности разработанного алгоритма

Было проведено 4 эксперимента, в каждом из которых двум алгоритмам были заданы одинаковые начальные условия. Карты, на которых проводились исследования, представлены на Рис. 5–8. Выполнение работы алгоритмов проводилось при $Q = \{0, 1, \dots, 74\}^2$ (размеры каждой карты равны 75x75 пикселей), при этом значение функции $F(x, y)$ соответствовало яркости пикселей изображения.

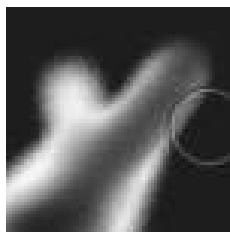


Рис. 5. Карта 1
Fig. 5. Map 1

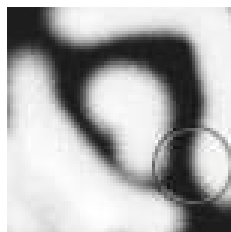


Рис. 6. Карта 2
Fig. 6. Map 2

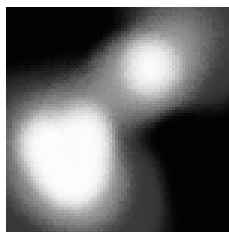


Рис. 7. Карта 3
Fig. 7. Map 3

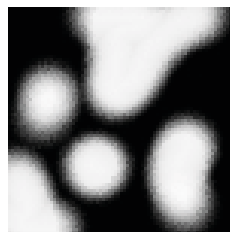


Рис. 8. Карта 4
Fig. 8. Map 4

В табл. 1 представлены результаты экспериментов при $n = 3, \varphi_0 = \frac{\pi}{3}$.

Таблица 1 / Table 1

Результаты работы алгоритмов при $n = 3, \varphi_0 = \frac{\pi}{3}$ Operation of algorithms for $n = 3, \varphi_0 = \frac{\pi}{3}$

Название карты / Name of the map	Начальная точка / The starting point	Конечная точка / The end point	Значение целевой функции / The value of the objective function	Время работы первого алгоритма (сек) / Operating time of the first algorithm (seconds)	Время работы второго алгоритма (сек) / Operating time of the second algorithm (seconds)	Ускорение работы / Speeding up work
1	(66,5)	(7,67)	7.8929	15.148	7.564	2
2	(39,74)	(52,0)	14.7368	33.843	4.854	6.97
3	(69,18)	(4,57)	13.4124	29.353	10.464	2.8
4	(12,1)	(61,55)	15.7091	57.049	11.897	4.8



Обсуждение результатов

Численные эксперименты показали, что алгоритм перебора элементов множества допустимых маршрутов от 2 до 7 раз быстрее алгоритма перебора элементов пространства Q (см. табл. 1). Кроме того, экспериментально установлена кубическая зависимость работы алгоритма от значения максимального угла поворота.

Ускорение алгоритма объясняется тем, что множество возможных положений каждой следующей точки поворота оказывается меньше не только исходного множества Q , но и множества допустимых положений для предыдущей точки поворота. Оценить отношение мер этих множеств можно, сравнивая площади геометрических представлений множеств $S(A, B, i\varphi_0)$. Введём функцию

$$\phi_i(\varphi_0) = \frac{|S(A, B, (i-1)\varphi_0)|}{|S(A, B, i\varphi_0)|}, i = 2, 3, \dots, n, \quad (5)$$

где n — количество поворотов маршрута.

На рис. 9. представлен график функции $\phi_i(\varphi_0)$. Точками отмечены значения при $\varphi_0 = 0.65$ рад. Значения на оси ординат показывают, во сколько раз площадь последующего множества $S(A, B, (i-1)\varphi_0)$ меньше площади предыдущего $S(A, B, i\varphi_0)$.

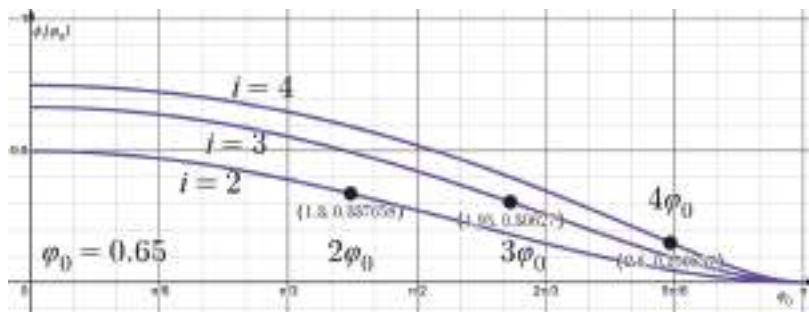


Рис. 9. График функции $\phi_i(\varphi_0)$ для $n = 4$, то есть $i = 2, 3, 4$

Fig. 9. Graph of a function $\phi_i(\varphi_0)$ for $n = 4$, that is, $i = 2, 3, 4$

Для оценки эффективности разработанного метода полезно обратиться к результатам работ (García, Rodríguez, 2021), в которых рассматривались альтернативные стратегии учёта угловых ограничений при планировании кусочно-прямолинейных траекторий. Предложенный в настоящей статье подход, основанный на точной формуле (3), обеспечивает большее уменьшение меры пространства перебора по сравнению с классическими методами динамического программирования без учёта последовательного уменьшения допустимой области для каждой последующей точки поворота. Дополнительное ускорение достигается за счёт кэширования



значений функционала, что согласуется с рекомендациями по оптимизации переборных алгоритмов, приведёнными в (Sidorov, Kuznetsov, 2024).

Исходя из анализа зависимости работы программы от максимального угла поворота, можно рекомендовать следующий подход к применению алгоритма. Поскольку существует некоторое максимальное $\varphi' \leq \varphi_0$, такое, что при любом значении $\varphi \in [\varphi', \varphi_0]$ решение не меняется, имеет смысл сначала применять алгоритм для углов меньших 90 градусов. Так как время кубически пропорционально углу, то даже при $\varphi_0 = 45$ градусов алгоритм будет работать в 8 раз быстрее, чем при $\varphi_0 = 90$ градусов.

Заключение

Экспериментально подтверждено, что разработанный алгоритм перебора множества допустимых маршрутов решает задачу нахождения оптимального кусочно-прямолинейного маршрута с n поворотами быстрее алгоритма полного перебора (ускорение в 2–7 раз). Установлена кубическая зависимость времени работы от максимального угла поворота φ_0 .

Можно отметить, что результаты исследования могут быть применены в областях робототехники (Enric Galceran and Marc Carreras, 2013) и беспилотных летательных аппаратов (Guangjun Gao and Jijian Lu, 2025).

Ограничения. Задача остаётся NP-трудной, поэтому приемлемое время работы программы достигается при $n \leq 3$ и размере карты, не превышающем 10^5 элементов. Рекомендуется выдерживать баланс между размером карты и количеством поворотов, а при малых φ_0 алгоритм эффективен даже на картах размером больше 10^5 или при $n > 3$.

Limitations. The problem remains NP-hard. Acceptable running time of the program is achieved when $n \leq 3$ and the map size does not exceed 10^5 elements. It is recommended to maintain a balance between the map size and the number of turns. For small values of φ_0 , the algorithm remains efficient even on maps larger than 10^5 elements or when $n > 3$.

Список источников / References

1. Королько, Н.С., Свойкин, Ф.В., Рего Г.Э. [и др.]. (2025). *Оптимальная схема разработки лесосеки канатной трелёвочной установкой*, 252, 317–334. (In Russ.). <https://www.elibrary.ru/item.asp?id=80515845>
Korolko, N.S., Svoikin, F.V., Rego G.E. [et al.]. (2025). *Optimal logging development scheme using a cable skidding rig*, 252, 317–334. (In Russ.). <https://www.elibrary.ru/item.asp?id=80515845>
2. Нефедов, В.Н., Свойкин, Ф.В., Гарибян, Б.А., Ряпухин, А.В., Королько, Н.С. (2025). *Методы аппроксимации двумерных множеств конечными множествами и их приложение к некоторым геометрическим задачам оптимизации*, 29(1), 129–157. (In Russ.). <https://www.elibrary.ru/item.asp?id=82963398>



- Nefedov, V.N., Svoikin, F.V., Garibyan, B.A., Ryapukhin, A.V., Korolko, N.S. (2025). *Methods of approximation of two-dimensional sets by finite sets and their application to some geometric optimization problems*, 29(1), 129–157. (In Russ.). <https://www.elibrary.ru/item.asp?id=82963398>
3. Нефедов, В.Н., Наседкин, Г.К. (2025). *Задача нахождения оптимального кусочно-прямолинейного маршрута с n поворотами*, 24, 282–284. (In Russ.). <https://www.elibrary.ru/item.asp?id=83139651>
- Nefedov, V.N., Nasedkin G.K. (2025). *The problem of finding an optimal piecewise rectilinear route with n turns*, 24, 282–284. (In Russ.). <https://www.elibrary.ru/item.asp?id=83139651>
4. Петров, А.В., Михайлова, Е. В. (2023). *Численные методы оптимизации маршрутов с ограничениями на углы поворота*, 45(3), 211–228. (In Russ.). <https://www.elibrary.ru/item.asp?id=51234567>
- Petrov, A.B., Mikhailova, E.V. (2023). *Numerical Methods for Route Optimization with Constraints on Turning Angles*, 45(3), 211–228. (In Russ.). <https://www.elibrary.ru/item.asp?id=51234567>
5. Свойкин, Ф.В., Королько, Н.С., Угрюмов, С.А., Россихин, К.В. (2024). *Построение трассы канатной дороги математически-программными методами*, 250, 252–272. (In Russ.). <https://www.elibrary.ru/item.asp?id=75137965>
- Svoikin, F.V., Korolko, N.S., Ugryumov, S.A., Rossikhin, K.V. (2024). *Construction of a cableway route using mathematical and software methods*, 250, 252–272. (In Russ.). <https://www.elibrary.ru/item.asp?id=75137965>
6. Сидоров, И.И., Кузнецов, Д.А. (2024). *Эффективные алгоритмы перебора для задач дискретной оптимизации маршрутов*, 18(2), 88–103. (In Russ.). <https://www.elibrary.ru/item.asp?id=67890123>
- Sidorov, I.I., Kuznetsov, D.A. (2024). *Efficient enumeration algorithms for discrete route optimization problems*, 18(2), 88–103. (In Russ.). <https://www.elibrary.ru/item.asp?id=67890123>
7. Enric Galceran and Marc Carreras. (2013). *A survey on coverage path planning for robotics. Robotics and Autonomous systems*, 61(12), 1258–1276. <https://www.sciencedirect.com/science/article/abs/pii/S092188901300167X>
8. Guangjun Gao, Jijian Lu. (2025). *CE-Bi-RRT*: Enhanced Bidirectional RRT* with Cooperative Expansion Strategy for Autonomous Drone Navigation*, 9(12), 831. <https://www.mdpi.com/2504-446X/9/12/831>
9. Garcia, M.J., Rodriguez, L.P. (2021). *Dynamic programming for piecewise-linear path optimization with turn constraints*, 15(4), 332–347. <https://doi.org/10.1007/s10472-021-09765-9>
10. Hoffmann, C. M., Juan-Arinyo, R. (2022). *On the complexity of polygonal path planning under angular constraints*, 58, 102–119. <https://doi.org/10.1016/j.comgeo.2022.101966>
11. Ivanov, V. M. (2025). *Simulation model of spline interpolation of piecewise linear trajectory for CNC machine tools*, 17(2), 225–242
12. Junjie Lu, Bi Zeng, Jingtao Tang, Tin Lun Lam. (2022). *TMSTC*: A Turn-minimizing Algorithm For Multi-robot Coverage Path Planning*, 2212, 02231. <https://arxiv.org/abs/2212.02231>
13. Nefedov V.N. (2025). *Problem of Finding an Optimal Piecewise Linear Path Connecting Two Given Points with the Possibility of Making n Turns*, 2605, 15449. <https://doi.org/10.48550/arXiv.2605.15449>



14. Yang, K., Sukkarieh, S. (2020). *Real-time piecewise-linear path planning for UAVs in dynamic environments*, 108, 104–118. <https://doi.org/10.1016/j.robot.2020.104118>

Информация об авторах

Виктор Николаевич Нефедов, кандидат физико-математических наук, доцент кафедры математической кибернетики, Московский авиационный институт (Национальный исследовательский университет) (МАИ), Москва, Российская Федерация, ORCID: <https://orcid.org/0000-0001-6053-2066>, e-mail: nefedovvn54@yandex.ru

Георгий Константинович Наседкин, ассистент, институт «Компьютерные науки и прикладная математика», Московский авиационный институт (Национальный исследовательский университет) (МАИ), Москва, Российская Федерация, ORCID: <https://orcid.org/0009-0006-5173-4110>, e-mail: gnk_02@mail.ru

Information about the authors

Viktor N. Nefedov, Candidate of Physico-Mathematical Sciences, Associate Professor of the Department of Mathematical Cybernetics, Moscow Aviation Institute (National Research University) (MAI), Moscow, Russian Federation, ORCID: <https://orcid.org/0000-0001-6053-2066>, e-mail: nefedovvn54@yandex.ru

Georgy K. Nasedkin, Assistant, Institute of Computer Science and Applied Mathematics, Moscow Aviation Institute (National Research University) (MAI), Moscow, Russian Federation, ORCID: <https://orcid.org/0009-0006-5173-4110>, e-mail: gnk_02@mail.ru

Вклад авторов

Все авторы приняли участие в обсуждении результатов и согласовали окончательный текст рукописи.

Contribution of the authors

All authors participated in the discussion of the results and approved the final text of the manuscript.

Конфликт интересов

Авторы заявляют об отсутствии конфликта интересов.

Conflict of interest

The authors declare no conflict of interest.

Поступила в редакцию 15.05.2026

Поступила после рецензирования 07.08.2026

Принята к публикации 08.08.2026

Опубликована 30.09.2026

Received 2026.05.15

Revised 2026.08.07

Accepted 2026. 08.08

Published 2026.09.30