

УДК 004.023

DOI: [10.26102/2310-6018/2024.45.2.043](https://doi.org/10.26102/2310-6018/2024.45.2.043)

## Модифицированный муравьиный алгоритм для построения туристического маршрута с ограничением по времени

О.А. Медведева, А.Ю. Минакова 

*Воронежский государственный университет, Воронеж, Российская Федерация*

**Резюме.** В статье рассматривается задача построения туристического маршрута с заранее заданными точками начала и конца маршрута. Объекты делятся на два типа. Первые – обязательные, которые наверняка должны войти в результирующий маршрут. Вторые – дополнительные, посещение которых не является необходимым. Маршрут формируется с учетом приоритетов, заранее заданных для объектов туристом исходя из его интересов и предпочтений, при этом суммарное время посещения объектов не должно превышать заданного крайнего срока прибытия в конечную точку маршрута. Для решения поставленной задачи в статье предлагается подход, основанный на построении известными методами маршрута по основным объектам и дальнейшем его расширении с использованием муравьиных стратегий. С этой целью вводится понятие «сытости» муравья и вероятности возврата к основному маршруту, чтобы была возможность контроля оставшегося времени. В завершении статьи приводятся результаты вычислительного эксперимента, направленного на оценку влияния параметров муравьиного алгоритма на полученный маршрут и разработку рекомендаций по настройке этих параметров в зависимости от размерности задачи. Кроме того, проводится сравнительный анализ маршрутов, полученных предложенным алгоритмом и точным методом ветвей и границ по заданному набору объектов, по результатам которого сделан вывод об эффективности предложенного алгоритма.

**Ключевые слова:** туристический маршрут, муравьиный алгоритм, приоритет, задача коммивояжера, вероятностный выбор.

**Для цитирования:** Медведева О.А., Минакова А.Ю. Модифицированный муравьиный алгоритм для построения туристического маршрута с ограничением по времени. *Моделирование, оптимизация и информационные технологии.* 2024;12(2). URL: <https://moitvvt.ru/ru/journal/pdf?id=1590> DOI: 10.26102/2310-6018/2024.45.2.043

## Modified ant algorithm for constructing a time-limited tourist route

О.А. Medvedeva, A.Yu. Minakova 

*Voronezh State University, Voronezh, the Russian Federation*

**Abstract.** The article considers the task of building a tourist route with predetermined points of the beginning and end of the route. The objects are divided into two types. The first ones are mandatory, which should certainly be included in the resulting route. And the second ones are additional ones, which are not necessary to visit. The route is formed taking into account the priorities set for the objects by the tourist, based on his interests and preferences, while the total time of visiting the objects should not exceed the specified deadline for arrival at the end point of the route. To solve this problem, the article proposes an approach based on the construction of a route by known methods along the main objects and its further expansion using ant strategies. To this end, the concept of "satiety" of the ant and the probability of returning to the main route are introduced, so that it is possible to control the time reserve. At the end of the article, we present the results of a computational experiment aimed at assessing the influence of the ant algorithm parameters on the resulting route and developing recommendations for adjusting these parameters depending on the size of the problem. In addition, a comparative analysis of

the routes obtained by the proposed algorithm and the exact branch-and-bound method for a given set of objects is carried out, based on the results of which a conclusion is drawn about the effectiveness of the proposed algorithm.

**Keywords:** tourist route, ant algorithm, priority, traveling salesman's task, probabilistic choice.

**For citation:** Medvedeva O.A., Minakova A.Yu. Modified ant algorithm for constructing a time-limited tourist route. *Modeling, Optimization and Information Technology*. 2024;12(2). URL: <https://moitvvt.ru/ru/journal/pdf?id=1590> DOI: 10.26102/2310-6018/2024.45.2.043 (In Russ.).

## Введение

Задача построения туристического маршрута остается крайне актуальной в современном мире, где возрастает потребность в планировании маршрута для пользователя с учетом времени, которое закладывает турист на преодоление маршрута. При этом полученный маршрут должен отвечать интересам пользователя, его потребностям, приоритетам.

Классическая задача планирования маршрута широко известна и хорошо изучена. Вопросы поиска оптимального маршрута поднимаются в трудах математиков Уильяма Роуэна Гамильтона, Джорджа Данцига, Ричарда Мэннинга Карпа, Дэвида Аплгейта, Герхарда Райнельта [1].

Но и в нынешнее время доработка и создание новых алгоритмов для построения туристического маршрута актуальна. Практический интерес представляют различные модификации, связанные с внесением дополнительных условий на маршрут и на характеристики посещаемых объектов (достопримечательностей). Так, в работе [2] рассматривается задача поиска туристического маршрута с помощью алгоритма TRP, он основан на роевом интеллекте для многоцелевого планирования маршрута (SIGPA). Его основная цель – учесть влияние таких факторов, как безопасность, предпочтения пользователей, наличие условий для людей с ограниченными возможностями. В [3] рассматривается задача составления туристических маршрутов на основе тематических предпочтений пользователя. Для решения данной задачи используются методы решения задачи о рюкзаке и метод ветвей и границ. В задаче выделяются ключевые критерии, учитывающие подмножества характеристик достопримечательностей, что помогает найти маршрут, отвечающий требованиям туристов.

Для решения задачи нахождения туристического маршрута актуальной остается проблема построения быстрых и эффективных эвристических алгоритмов решения. Довольно часто для решения данной проблемы предлагается использовать подходы, основанные на генетических алгоритмах. Так, в [4] при моделировании за основу взята задача коммивояжера, а решение ищется с помощью генетического алгоритма. При этом достопримечательности делятся по определенным группам, которые пользователь может выбрать исходя из своих интересов. В работе [5] предлагается метод MASA – модифицированный алгоритм A\*, который позволяет объединить многочисленные факторы с разными весами в многовзвешенном графике, чтобы предложить лучший туристический маршрут. Для построения маршрута используется также генетический алгоритм. Для разработки индивидуального туристического маршрута однодневной экскурсии по достопримечательностям в работе [6] рассмотрены и доработаны генетический алгоритм и алгоритм разностной эволюции, на основе которого построен четырехэтапный эвристический алгоритм.

Также существуют методы, которые объединяют несколько известных подходов в один. Так, в [7] рассматривается задача планирования туристической поездки для групп. В статье предложен гибридный алгоритм GRASP-VND, основанный на жадной адаптивной процедуре поиска, оптимальности по Парето и переменном поиске соседей.

Алгоритм учитывает интересы пользователей, и групповое планирование должно осуществляться с учетом максимальной вместимости объекта.

В данной статье будет рассмотрена задача построения туристического маршрута с фиксированным набором объектов, обязательных к посещению. При этом возможно добавление дополнительных объектов в маршрут с учетом приоритетов пользователя и заданного ограничения по времени прохождения маршрута.

### Постановка задачи

Рассмотрим следующую задачу. Пусть задана совокупность неподвижных обязательных объектов (типа А) и дополнительных объектов (типа В), координаты которых изначально известны. Мобильный объект должен обязательно посетить все неподвижные объекты типа А и по возможности объекты типа В, при этом каждый неподвижный объект может быть посещен только один раз. Кроме того, маршрут мобильного объекта должен начинаться и заканчиваться в некоторых фиксированных обязательных объектах типа А. Требуется построить маршрут для мобильного объекта с целью максимизации суммарных приоритетов неподвижных объектов, входящих в маршрут, при этом время, затраченное на маршрут, не должно превышать заданного резерва, которым располагает мобильный объект.

Возможный маршрут мобильного объекта представлен на Рисунке 1. Здесь кружками обозначены обязательные объекты, квадратами – дополнительные объекты. Объекты с номерами А0 и А5 являются начальным и конечным соответственно. Жирными стрелками обозначен минимальный маршрут, который должен пройти мобильный объект, чтобы посетить только обязательные объекты. Пунктирными стрелками обозначены возможные расширения маршрута с учетом добавления дополнительных неподвижных объектов.

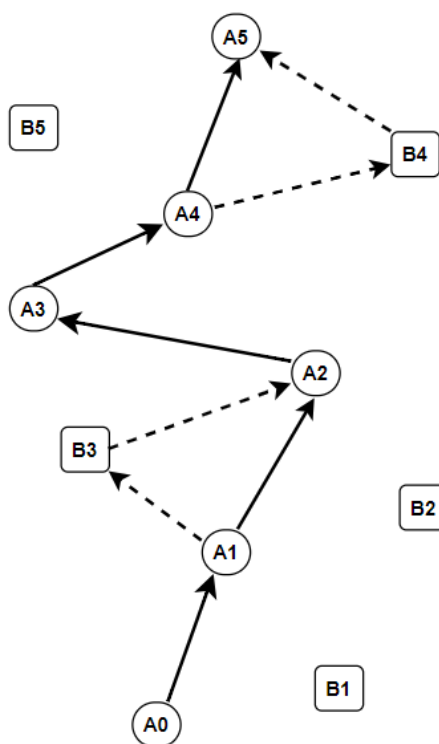


Рисунок 1 – Возможный маршрут мобильного объекта  
 Figure 1 – Possible route of a mobile object

Содержательная постановка задачи может быть, например, представлена как построение маршрута по достопримечательностям для туриста. Пусть дана карта города, на которой выделены достопримечательности. Каждая достопримечательность имеет некоторое значение рейтинга, полученное, например, за счет оценивания этих мест другими туристами. Турист задает набор достопримечательностей, который обязательно хочет посетить, а также место начала и конца маршрута.

Необходимо построить туристический маршрут, который будет включать в себя все достопримечательности, которые интересны туристу, и, при наличии доступного времени, может охватить дополнительные достопримечательности, расположенные рядом. Добавление дополнительных достопримечательностей в маршрут происходит с целью максимизации суммарного приоритета посещенных объектов.

### Математическая модель

Составим математическую модель рассматриваемой задачи в виде задачи математического программирования. Для этого введем следующие обозначения:

$N$  – количество неподвижных объектов.

$C = (c_{ij})$  – матрица временных затрат, где  $c_{ij}$ ,  $i \neq j$  – время, затрачиваемое на переход из  $i$ -го неподвижного объекта в  $j$ -й;  $c_{ii}$  – время, необходимое для посещения  $i$ -го объекта.

$pr = (pr_i)$ ,  $i = \overline{1, N}$  – вектор приоритетов неподвижных объектов, где  $pr = 1$  соответствует значению «не интересно»,  $pr = 2$  – «не очень»,  $pr = 3$  – «средне»,  $pr = 4$  – «интересно» и  $pr = 5$  – «очень интересно».

$T$  – время, которым располагает мобильный объект.

$S$  – множество индексов неподвижных объектов, которые обязательно должны войти в маршрут мобильного объекта  $S \in \{0, 1, \dots, N\}$ .

Отдельно следует сказать о начальном и конечном объектах маршрута. Не ограничивая общности, будем считать, что некоторый объект с индексом 0 будет являться началом маршрута, а объект с индексом  $N$  – концом, при этом они должны быть обязательно посещены мобильным объектом, т. е.  $0 \in S$  и  $N \in S$ .

Введем следующие переменные:

$x_{ij} \in \{0, 1\}$ ,  $i, j = \overline{0, N}$ . Если путь из  $i$ -го объекта в  $j$ -й объект вошел в маршрут мобильного объекта, то  $x_{ij} = 1$ , в противном случае  $x_{ij} = 0$ .

Рассмотрим подробнее ограничения задачи. По условию требуется, чтобы обязательные объекты вошли в результирующий маршрут, причем только один раз. Это требование можно описать ограничениями вида:

$$\sum_{i=0}^{N-1} x_{ij} = 1, \forall j \in S \setminus \{0\},$$

$$\sum_{j=1}^N x_{ij} = 1, \forall i \in S \setminus \{N\}.$$

Суммарное время, потраченное на маршрут, должно не превышать времени, которым располагает мобильный объект:

$$\sum_{i=0}^{N-1} \sum_{j=1}^N (c_{ij} + c_{ii}) x_{ij} \leq T.$$

Если мобильный объект приезжает в любой  $i$ -й неподвижный объект, то он должен из этого же объекта выехать.

$$\sum_{i=0}^{N-1} x_{ij} = \sum_{k=1}^N x_{jk}, \forall j = \overline{1, N-1}.$$

Отдельно следует сказать о требовании отсутствия подциклов. Для поставленной задачи оно будет иметь вид, аналогичный таковому в известной задаче коммивояжера [8].

Целевая функция отвечает за максимизацию суммарного приоритета маршрута и будет иметь вид:

$$\sum_{i=0}^N \sum_{j=0}^N pr_i x_{ij} \rightarrow \max.$$

Таким образом итоговая математическая модель поставленной задачи примет вид:

$$\sum_{i=0}^N \sum_{j=0}^N pr_i x_{ij} \rightarrow \max,$$

$$\sum_{i=0}^{N-1} x_{ij} = 1, \forall j \in S \setminus \{0\},$$

$$\sum_{j=1}^N x_{ij} = 1, \forall i \in S \setminus \{N\},$$

$$\sum_{i=0}^{N-1} \sum_{j=1}^N (c_{ij} + c_{ii}) x_{ij} \leq T,$$

$$\sum_{i=0}^{N-1} x_{ij} = \sum_{k=1}^N x_{jk}, \forall j = \overline{1, N-1},$$

+ условие отсутствия подциклов.

### Алгоритм решения

Решение поставленной задачи предлагается разбить на несколько основных этапов:

1. Построение маршрута, включающего только обязательные объекты.
2. Оценка полученного маршрута на допустимость.
3. Расширение построенного маршрута за счет добавления в него дополнительных объектов с учетом их приоритетности и ограничения по доступному времени.

Остановимся на каждом этапе подробнее. По условию задачи искомым маршрутом должен начинаться и заканчиваться в фиксированных обязательных объектах. В зависимости от того, совпадают или нет эти точки, для построения маршрута по обязательным объектам можно использовать разные алгоритмы. Так, если маршрут должен начинаться и заканчиваться в одном объекте, задача сводится к известной задаче коммивояжера и может быть решена методом ветвей и границ<sup>1</sup> [8]. Если же точки начала и конца не совпадают, необходимо построить гамильтонову цепь минимальной длины по заданной матрице  $C$  [9].

При проведении вычислительного эксперимента будет рассматриваться вариант построения замкнутого маршрута.

Следующим этапом необходимо оценить полученный маршрут на соответствие условию задачи, а именно, выполнение временного ограничения. Если суммарное время прохождения маршрута окажется больше заявленного, то исходная задача несовместна. При этом можно, например, уменьшить количество или изменить набор основных объектов, чтобы получить допустимый маршрут.

Если временное ограничение для полученного маршрута выполнено, то вычисляется доступный резерв времени, оставшийся после выполнения найденного маршрута, и начинается процесс расширения маршрута добавлением дополнительных объектов с учетом ограничения по времени.

<sup>1</sup>Kianfar K. Branch-and-Bound Algorithms. В книге: *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons; 2011. <https://doi.org/10.1002/9780470400531.eorms0116>

Для реализации данной процедуры предлагается использовать модифицированный муравьиный алгоритм.

### Модификация муравьиного алгоритма

Идея данного алгоритма заключается в добавлении к основному маршруту, найденному на предыдущем этапе, дополнительных объектов. Расширение маршрута проходит в рамках времени, оставшихся после выполнения основного маршрута. При этом порядок следования объектов основного маршрута нарушать нельзя.

Дополнительный объект может быть добавлен в маршрут, если мобильный объект располагает временем, необходимым, чтобы посетить его и вернуться к следующему объекту основного маршрута. Для реализации данной идеи предлагается использовать муравьиный алгоритм [10].

Вероятность перехода  $k$ -го муравья из  $i$ -го объекта в  $j$ -й будет задаваться вероятностно-пропорциональным правилом согласно формуле (1):

$$p = \frac{(pheromone_{ij})^\alpha (attractiveness_{ij})^\beta}{\sum (pheromone_{ij})^\alpha (attractiveness_{ij})^\beta}, \quad (1)$$

где  $pheromone_{ij}$  – количество феромона на пути из  $i$ -го объекта в  $j$ -й,  $attractiveness_{ij} = \frac{1}{c_{ij}}$  – привлекательность пути  $(i, j)$ ,  $\alpha, \beta$  – настраиваемые параметры.

Правило обновления феромона рассчитывается согласно следующей формуле:

$$pheromone_{ij} = (1 - p) \cdot pheromone_{ij} + A, \quad (2)$$

где  $p$  – коэффициент испарения феромона,  $A$  – дополнительное количество феромонов.

Кроме того, в классический муравьиный алгоритм предлагается добавить дополнительный параметр – «сытость», который будет отвечать за доступный запас времени у муравья [11]. Изначально «сытость» муравья равна отношению разницы между временем  $T$ , которым располагает мобильный объект, к времени  $tm$ , затраченным на основной маршрут, к общему времени:

$$capacity = \frac{(T - tm)}{T}. \quad (3)$$

Также вводится вероятность возврата муравья к основному маршруту, зависящая от текущей «сытости»:

$$pv = 1 - \frac{1}{1 + e^{-capacity}}. \quad (4)$$

Чем больше значение «сытости», тем с большей вероятностью муравей уйдет с основного маршрута с целью поиска дополнительных объектов для добавления. При выключении нового объекта в маршрут «сытость» муравья будет уменьшаться и соответственно вероятность возврата на основной маршрут расти. Эта идея согласуется с реальностью так муравей стремится вернуться в муравейник, если голоден, не доводя до голодной смерти.

Рассмотрим модифицированный муравьиный алгоритм поэтапно.

### Модифицированный муравьиный алгоритм

Этап 1. Задать количество муравьев  $K$ , количество итераций  $N_{max}$ , основной маршрут зафиксировать в качестве рекордного, задать номер текущего муравья  $k = 1$ , номер итерации  $t = 0$ .



Этап 2. Поместить муравья в начальную точку и восполнить сытость  $capacity = \frac{(T-tm)}{T}$ .

Этап 3. Проверить условия останова по следующим правилам:

Если есть не посещенные объекты, муравей сыт  $capacity > 0$  и текущий объект не является конечным объектом, то перейти к этапу 4, иначе перейти к этапу 7.

Этап 4. Рассчитать вероятности перехода муравья из текущего  $i$ -го объекта к следующему  $j$ -му объекту по формуле (1).

Если текущая сытость  $capacity > \frac{c_{ij}+c_{jd}-c_{id}}{T}$ , где  $d$  – последующий объект, который входит в основной маршрут, то совершить вероятностный переход в  $j$ -й объект и перейти к этапу 5. Иначе перейти к следующему  $d$ -му объекту из основного маршрута и перейти к этапу 3.

Этап 5. Обновить феромоны по формуле (2).

Этап 6. Уменьшить «сытость» муравья согласно времени, потраченному на преодоление пути, т. е.  $capacity = \frac{capacity-c_{ij}}{T}$ , и перейти к этапу 3.

Этап 7. Сравнить полученный маршрут с текущим рекордным маршрутом.

Если суммарный приоритет всех объектов из полученного маршрута больше, чем суммарный приоритет рекордного маршрута, то взять полученный маршрут в качестве рекорда и перейти к этапу 8.

Этап 8. Проверить условие останова по муравьям:  $k < K$ . Если выполнено, то  $k = k + 1$  и перейти к этапу 2. Иначе к шагу 3.

Этап 9. Проверить условие останова по количеству итераций  $t < N_{max}$ . Если выполнено, то  $t = t + 1$  и перейти к этапу 2. Иначе останов. Рекордный маршрут берется в качестве ответа.

### Вычислительный эксперимент и обсуждение результатов

Для исследования предложенного алгоритма решения задачи и анализа параметров модифицированного муравьиного алгоритма был проведен вычислительный эксперимент на основе реализованного на языке Python программного комплекса.

Цели вычислительного эксперимента:

1. Исследование влияния параметров муравьиного алгоритма на результирующий маршрут;

2. Сравнение результатов работы предложенного алгоритма и точного метода ветвей и границ для заданного набора объектов.

В работе предполагается, что матрица временных затрат  $C$  заполняется случайным образом значениями из отрезка  $[1, 100]$ . Количество обязательных объектов фиксировано и равно 10, резерв времени  $T = 720$  минут для всех размерностей задачи.

Вектор приоритетов  $pr$  заполняется следующим образом: приоритет у основных объектов задается равным 5, а у остальных объектов он получает целое случайное значение от 1 до 4.

Для каждой размерности задачи задаются 100 матриц временных затрат и вычисляются средний суммарный приоритет маршрутов  $prior$ , среднее количество дополнительных объектов  $dop$ , добавленных в маршрут, а также общее время маршрута и неизрасходованный остаток времени.

В Таблице 1 представлены результаты работы алгоритма для задач разной размерности при разном количестве муравьев в колонии. Количество итераций фиксировано и равно 100.

Таблица 1 – Результаты первого вычислительного эксперимента  
Table 1 – Results of the first computational experiment

Количество объектов	Количество муравьев									
	20		50		100		200		500	
	<i>dop</i>	<i>prior</i>	<i>dop</i>	<i>prior</i>	<i>dop</i>	<i>prior</i>	<i>dop</i>	<i>prior</i>	<i>dop</i>	<i>prior</i>
20	5	59	7	62	10	73	10	73	10	73
30	6	61	7	66	11	79	11	79	11	79
50	6	60	8	68	12	83	12	83	12	83
100	11	65	11	73	14	90	14	90	14	90
200	16	74	19	81	22	116	24	119	25	121
500	27	90	31	112	38	186	41	192	41	197

На основе данных, представленных в Таблице 1, можно сделать следующие выводы:

1. При увеличении количества муравьев в колонии увеличивается суммарный приоритет полученного маршрута. Иными словами, добавляются дополнительные объекты с более высоким приоритетом. При малом же количестве муравьев результаты хуже, так как рассматривается меньше допустимых маршрутов.

2. Для задач размерности меньше 100 объектов предпочтительно задавать количество муравьев в колонии, равное 100. При таких размерностях дальнейшее увеличение количества муравьев не приводит к улучшению суммарного приоритета маршрута. Заметим, что данная рекомендация применима для решения практических задач формирования маршрута туриста, так как в среднем карта местности включает не более 100 достопримечательностей.

3. При увеличении размерности более 100 объектов количество муравьев необходимо брать порядка размерности задачи.

В Таблице 2 представлены результаты работы предложенного алгоритма при разном количестве итераций и фиксированном количестве муравьев, равном 100. При этом оценивается суммарный приоритет полученного маршрута и оставшееся неизрасходованное время.

По данным, представленным в Таблице 2, можно сделать вывод, что увеличение количества итераций алгоритма более 100 не оказывает существенного влияния на суммарный приоритет полученного маршрута. При этом остаток неизрасходованного времени принимает значения, не превышающие 1 % от общего доступного времени. Это означает, что предложенный алгоритм позволяет в полной мере использовать допустимый резерв времени при расширении основного маршрута.

Таблица 2 – Результаты второго вычислительного эксперимента  
Table 2 – Results of the second computational experiment

Количество объектов	Количество итераций	Основной маршрут		Конечный маршрут		Остаток неизрасходованного времени
		<i>prior</i>	<i>Время</i>	<i>prior</i>	<i>Общее время</i>	
20	100	50	352	73	713	7
	500	50	352	73	714	6
	1000	50	352	73	714	6
30	100	50	358	79	717	3
	500	50	358	79	718	2
	1000	50	358	79	718	2



Таблица 2 (продолжение)

Table 2 (continued)

50	100	50	325	83	716	4
	500	50	325	83	716	4
	1000	50	325	85	717	3
100	100	50	318	90	712	8
	500	50	318	90	714	6
	1000	50	318	92	715	5
500	100	50	291	186	718	2
	500	50	291	195	717	3
	1000	50	291	201	718	2

В следующем эксперименте исследуется эффективность работы предложенного алгоритма. С этой целью сначала по заданной матрице временных затрат предложенным алгоритмом строится маршрут с 10 обязательными объектами. Далее фиксируются все объекты, попавшие в результирующий маршрут, и по полученному набору объектов решается задача коммивояжера точным методом ветвей и границ. При этом суммарный приоритет маршрута, очевидно, не изменится, однако может поменяться остаток неизрасходованного времени.

Таблица 3 – Результаты третьего вычислительного эксперимента

Table 3 – Results of the third computational experiment

Количество объектов	Предложенный алгоритм		Метод ветвей и границ	
	<i>prior</i>	<i>Оставшиеся время</i>	<i>prior</i>	<i>Оставшиеся время</i>
20	73	7	73	12
30	79	3	79	11
50	83	4	83	18
100	90	8	90	31
500	186	2	186	84

Из данных, представленных в Таблице 3, можно сделать вывод, что за счет изменения порядка обхода объектов в маршруте, построенном методом ветвей и границ, он получается короче. Однако при этом может нарушиться порядок следования обязательных объектов в маршруте, что для предложенного в статье алгоритма является недопустимым. При этом для задач размерности менее 100 объектов, отношение разницы оставшихся неизрасходованными промежутков времени предложенного и точного алгоритмов к общему запасу времени не превышает 3%. Таким образом, можно сделать вывод, что рассмотренный в статье алгоритм позволяет получать близкое к оптимальному решение поставленной задачи.

### Заключение

В статье рассмотрена задача поиска туристического маршрута с заданным набором обязательных объектов и возможностью его расширения дополнительными объектами с учетом ограничения по времени. Данная задача актуальна в современном мире, так как достаточно высок спрос в разработке маршрута, который учитывал бы время, выделенное туристом на преодоление дистанции. Для добора объектов к основному маршруту использовался модифицированный муравьиный алгоритм. В

статье представлены результаты вычислительного эксперимента, направленного на настройку параметров предложенного алгоритма и анализ его работы.

### СПИСОК ИСТОЧНИКОВ / REFERENCES

1. Geunes J. *Operations Planning: Mixed Integer Optimization Models (Operations Research Series)*. Boca Raton: CRC Press; 2014. 218 p. <https://doi.org/10.1201/b17414>
2. Ntakolia C., Iakovidis D.K. A swarm intelligence graph-based pathfinding algorithm (SIGPA) for multi-objective route planning. *Computers & Operations Research*. 2021;133. <https://doi.org/10.1016/j.cor.2021.105358>
3. Сучкова Т.М., Волкова Л.Л. О разработке рекомендательной системы для составления туристических маршрутов на основе тематических предпочтений пользователя. В сборнике: *Искусственный интеллект в автоматизированных системах управления и обработки данных: Сборник статей II Всероссийской научной конференции: в 5 томах: Том 1, 27-28 апреля 2023 года, Москва, Россия*. Москва: Издательский дом «КДУ»; 2023. С. 145–150.  
Suchkova T.M., Volkova L.L. On developing a recommender system for touristic routes basing on user's thematic preferences. In: *Artificial Intelligence in Management, Control, and Data Processing Systems: Proceedings of the II All-Russian scientific conference: in 5 volumes: Volume 1, 27-28 April 2023, Moscow, Russia*. Moscow: Publishing House «KDU»; 2023. P. 145–150. (In Russ.).
4. Полетайкин А.Н. Компьютерная система составления туристических маршрутов методом генетических алгоритмов. В сборнике: *Техническая эксплуатация водного транспорта: проблемы и пути развития: Материалы Второй международной научно-технической конференции, 23-25 октября 2019 года, Петропавловск-Камчатский, Россия*. Петропавловск-Камчатский: Камчатский государственный технический университет; 2020. С. 158–162.  
Poletajkin A.N. Computer system for compiling tourist routes by method of genetic algorithms. In: *Tekhnicheskaya ekspluatatsiya vodnogo transporta: problemy i puti razvitiya: Materialy Vtoroi mezhdunarodnoi nauchno-tekhnicheskoi konferentsii, 23-25 October 2019, Petropavlovsk-Kamchatsky, Russia*. Petropavlovsk-Kamchatsky: Kamchatka State Technical University; 2020. P. 158–162. (In Russ.).
5. Pasandi L., Hooshmand M., Rahbar M. Modified A\* Algorithm integrated with ant colony optimization for multi-objective route-finding; case study: Yazd. *Applied Soft Computing*. 2021;113. <https://doi.org/10.1016/j.asoc.2021.107877>
6. Zheng W., Liao Zh., Qin J. Using a four-step heuristic algorithm to design personalized day tour route within a tourist attraction. *Tourism Management*. 2017;62:335–349. <https://doi.org/10.1016/j.tourman.2017.05.006>
7. Ruiz-Meza J., Brito J., Montoya-Torres J.R. A GRASP-VND algorithm to solve the multi-objective fuzzy and sustainable Tourist Trip Design Problem for groups. *Applied Soft Computing*. 2022;131. <https://doi.org/10.1016/j.asoc.2022.109716>
8. Корбут А.А., Финкельштейн Ю.Ю. *Дискретное программирование*. Москва: Наука; 1969. 368 с.  
Korbut A.A., Finkel'shtein Yu.Yu. *Diskretnoe programmirovaniye*. Moscow: Nauka; 1969. 368 p. (In Russ.).
9. Žak A. Growth order for the size of smallest hamiltonian chain saturated uniform hypergraphs. *European Journal of Combinatorics*. 2013;34(4):724–735. <https://doi.org/10.1016/j.ejc.2012.10.009>

10. Родзин С.И., Родзина О.Н., Эль-Хатиб С.А. Гибридный муравьиный алгоритм сегментации медицинских изображений. *Вестник Чувашского университета*. 2017;(3):262–272.  
Rodzin S., Rodzina O., El-Khatib S. Hybrid segmentation ant algorithms of medical images. *Vestnik Chuvashskogo universiteta*. 2017;(3):262–272. (In Russ.).
11. Medvedev S.N., Medvedeva O.A. Modified ant colony optimization algorithm for solving the vehicle routing problem with a given load capacity. *Journal of Physics: Conference Series*. 2021;1902. <https://doi.org/10.1088/1742-6596/1902/1/012123>

### ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

**Медведева Ольга Александровна**, кандидат физико-математических наук, доцент, доцент кафедры вычислительной математики и прикладных информационных технологий факультета прикладной математики, информатики и механики Воронежского государственного университета, Воронеж, Российская Федерация.

*e-mail:* [o\\_a\\_medvedeva@mail.ru](mailto:o_a_medvedeva@mail.ru)

**Olga A. Medvedeva**, Candidate of Physical and Mathematical Sciences, Associate Professor, Associate Professor of the Department of Computational Mathematics and Applied Information Technologies, Faculty of Applied Mathematics, Computer Science and Mechanics, Voronezh State University, Voronezh, the Russian Federation.

**Минакова Александра Юрьевна**, магистрант кафедры вычислительной математики и прикладных информационных технологий факультета прикладной математики, информатики и механики Воронежского государственного университета, Воронеж, Российская Федерация.

*e-mail:* [lminakova7@mail.ru](mailto:lminakova7@mail.ru)

**Aleksandra Yu. Minakova**, undergraduate student of the Department of Computational Mathematics and Applied Information Technologies of the Faculty of Applied Mathematics, Computer Science and Mechanics of Voronezh State University, Voronezh, the Russian Federation.

*Статья поступила в редакцию 29.05.2024; одобрена после рецензирования 10.06.2024; принята к публикации 18.06.2024.*

*The article was submitted 29.05.2024; approved after reviewing 10.06.2024; accepted for publication 18.06.2024.*