

RESEARCH OF MODELS AND METHODS FOR ROUTING AND PRACTICAL IMPLEMENTATION OF AUTONOMOUS MOVEMENT BY UNMANNED TRANSPORT SYSTEMS FOR CARGO DELIVERY

A. S. Kostin, N. N. Maiorov

Saint-Petersburg State University of Aerospace Instrumentation,
St. Petersburg, Russian Federation

Modern challenges in the field of transport logistics associated with the need to increase the speed of cargo transportation and reduce the cost of transportation require the development of solutions that will allow the introduction of unmanned vehicles to solve logistics problems. Within the framework of this paper, a new problem of routing an unmanned transport system through delivery points using annealing simulation methods and an ant colony algorithm is considered. These methods are chosen taking into account the real situation of complication of the conditions of possible delivery, the emergence of closed areas for delivery, obtaining a large number of points for delivery, or the emergence of other features. The selected algorithms are relatively simple to implement and, at the same time, form acceptable routes of movement, taking into account the initial data. These algorithms are studied in the software simulation environment Gazebo for the possibility of their application in unmanned transport systems and their effectiveness in constructing alternative traffic routes. In the software simulation, a program code for the autonomous movement of an unmanned vehicle is developed and a simulation environment is formed taking into account the test data. In the test environment, the results of the ant colony algorithm are checked against the test data, and special attention is paid to the study of the speed and accuracy of the selected algorithms. It is worth noting that the annealing simulation algorithm allows you to quickly optimize a given route, however, to obtain more accurate results, you should additionally use other optimization methods, such as the ant colony algorithm. As a result of the proposed methods application, it is concluded that it is possible to form the optimal route of movement relative to the initial data within the length of the path and the speed of the generated route.

Keywords: simulated annealing method, ant colony algorithm, unmanned transport system, Gazebo simulation environment.

For citation:

Kostin, Anton S., and Nikolai N. Maiorov. "Research of models and methods for routing and practical implementation of autonomous movement by unmanned transport systems for cargo delivery." *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S. O. Makarova* 15.3 (2023): 524–536. DOI: 10.21821/2309-5180-2023-15-3-524-536.

УДК 656.072

ИССЛЕДОВАНИЕ МОДЕЛЕЙ И МЕТОДОВ МАРШРУТИЗАЦИИ И ПРАКТИЧЕСКОГО ВЫПОЛНЕНИЯ АВТОНОМНОГО ДВИЖЕНИЯ БЕСПИЛОТНЫМИ ТРАНСПОРТНЫМИ СИСТЕМАМИ ДЛЯ ДОСТАВКИ ГРУЗОВ

А. С. Костин, Н. Н. Майоров

ФГАОУ ВО «Санкт-Петербургский государственный университет аэрокосмического приборостроения», Санкт-Петербург, Российская Федерация

Темой исследования являются современные тенденции в сфере транспортной логистики, связанные с потребностью в росте скорости грузоперевозок и снижением их стоимости, требующие проработки решений, которые позволят внедрить беспилотные транспортные средства для решения задач логистики. В работе рассмотрена новая задача маршрутизации беспилотной транспортной системы по точкам

доставки с применением методов имитации отжига и муравьиного алгоритма. Данные методы были выбраны с учетом реальной ситуации усложнения условий возможной доставки, появления закрытых зон для доставки, получения большого количества точек для доставки или возникновения других особенностей. Выбранные алгоритмы относительно просты в реализации и при этом формируют допустимые маршруты движения с учетом исходных данных. Данные алгоритмы исследованы в программной симуляционной среде Gazebo на предмет возможности их применения в беспилотных транспортных системах и их эффективности при построении альтернативных маршрутов движения. В программной симуляции разработан программный код автономного движения беспилотной транспортной и сформирована имитационная среда с учетом тестовых данных. В тестовой среде выполнена проверка результатов работы муравьиного алгоритма относительно тестовых данных, особое внимание уделено исследованию скорости работы и точности выбранных алгоритмов. Следует отметить, что алгоритм имитации отжига позволяет выполнить быструю оптимизацию заданного маршрута, однако для получения более точных результатов необходимо дополнительно применять другие оптимизационные методы (например, муравьиный алгоритм). В результате применения предложенных методов сделан вывод о том, что удалось сформировать оптимальный маршрут движения относительно исходных данных в рамках длины пути и установить скорости его прохождения.

Ключевые слова: метод имитации отжига, муравьиный алгоритм, беспилотная транспортная система, имитационная среда Gazebo.

Для цитирования:

Костин А. С. Исследование моделей и методов маршрутизации и практического выполнения автономного движения беспилотными транспортными системами для доставки грузов / А. С. Костин, Н. Н. Майоров // Вестник Государственного университета морского и речного флота имени адмирала С. О. Макарова. — 2023. — Т. 15. — № 3. — С. 524–536. DOI: 10.21821/2309-5180-2023-15-3-524-536.

Введение (Introduction)

Современные задачи логистики и управление цепей поставок требуют разработки и внедрения алгоритмов и методов маршрутизации беспилотными транспортными системами различного типа, в том числе морскими и воздушными [1], [2] с учетом существующих моделей управления. Уже сейчас прорабатываются точечные решения для маршрутизации беспилотных транспортных систем. Например, в 2015 г. компания DHL сообщила, что успешно доставила медицинские препараты и другие товары, являющиеся срочными, на небольшой остров на севере Германии с помощью дронов. 20 сентября беспилотное судно BBC BT-440 разработки и производства предприятия АО «НПП «Радар ММС», совершило первую в России коммерческую доставку. Груз для заказчика: ООО «Газпромнефть-Снабжение», массой 80 кг был доставлен на участок Восточно-Мессояхского месторождения из поселка Тазовский (Ямало-Ненецкий автономный округ) [1], [3]. Беспилотное судно с грузом по заданной траектории прошло более 120 км до точки посадки, груз был доставлен и беспилотник вернулся в точку взлета в поселке Тазовский, преодолев еще более 120 км. Движение в одну сторону, выполняемое на высоте порядка 200 м с учетом взлета и посадки, составило менее 1,5 ч, при этом дрон двигался со скоростью около 100 км/ч. Таким образом, был совершен успешный пилотный проект в данной области.

В рамках направления исследования беспилотных транспортных систем был выполнен сбор и анализ наукометрических данных при помощи программы VOSviewer. Данное программное обеспечение позволяет систематизировать и упорядочить ключевые слова в специальные *смысловые кластеры*, отражающие интерес к выбранной тематике за указанный период времени, а также построить наукометрическую картину в той или иной отрасли. В данном программном обеспечении были построены связи по ключевым словам относительно области поиска в сфере беспилотных транспортных систем (рис. 1).

На основе полученного графика связей можно сделать вывод о том, что наиболее важными в данной области исследованиями является разработка беспилотных транспортных систем, исследования в области логистики и доставки грузов при помощи беспилотных транспортных систем, машинное обучение и автоматизация, а также вопросы связи нескольких беспилотных транспортных систем [3], [5].

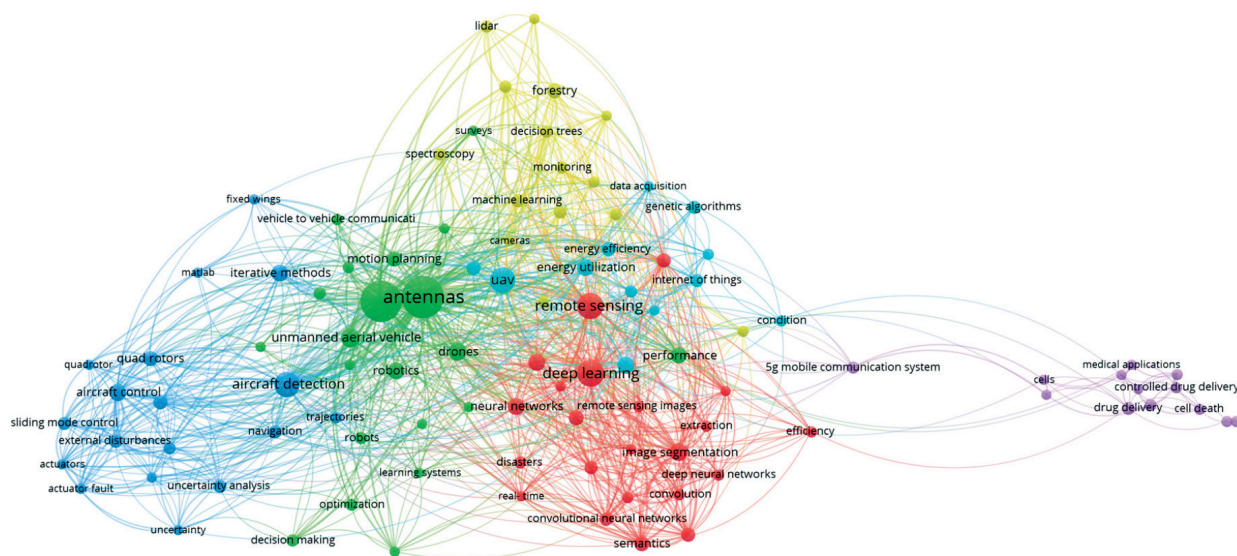


Рис. 1. Режим Network Visualization в VOSviewer

На рис. 2 показаны темпы доставки грузов при помощи беспилотных систем в 2019–2021 гг.

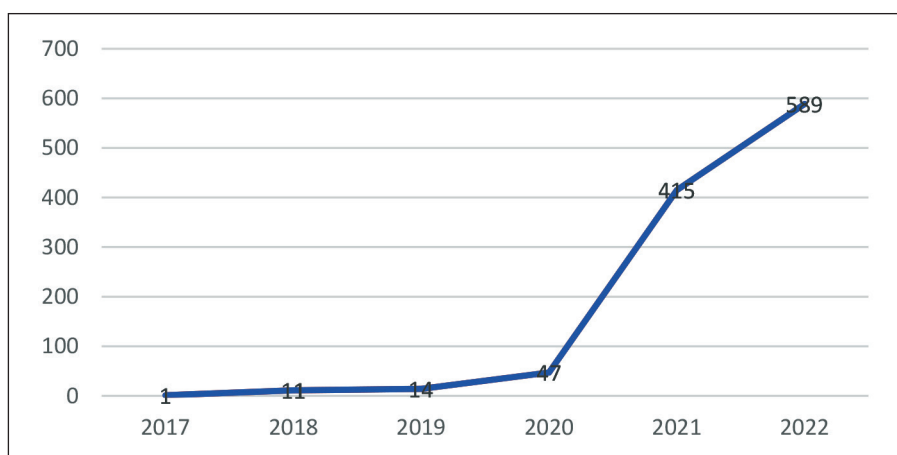


Рис. 2. Темпы роста доставки грузов при помощи использования беспилотных транспортных систем

В таблице приведены данные роста запросов по беспилотным транспортным системам за 2017–2022 гг.:

Год	Количество
2017	1
2018	11
2019	14
2020	47
2021	415
2022	589

Успешность выполнения задачи доставки грузов беспилотными транспортными системами напрямую зависит от проработки задачи маршрутизации, анализа аппаратных и приборных комплексов беспилотных систем. Данная задача особенно актуальная в ходе доставки грузов для мегаполисов. В условиях плотной городской застройки и увеличения числа автомобилей и транспортных

средств на дорогах доставка мелких посылок беспилотниками выглядит все более привлекательной. Аналогичную модельную ситуацию можно построить для морских транспортных систем. Однако существует ряд ограничений, который необходимо учитывать при выполнении миссий по доставке различных грузов. В данной работе представлено исследование вариантов маршрутизации при помощи различных алгоритмов на основе теории графов.

Методы и материалы (Methods and Materials)

Исследуя организацию маршрутизации беспилотных транспортных систем на микроуровне, можно обнаружить то, что данная задача не прорабатывается должным образом. Оператор в ручном режиме указывает точки доставки грузов без должного использования оптимизационных алгоритмов на основе дальности доставки из стартовой точки в конечную (рис. 3).

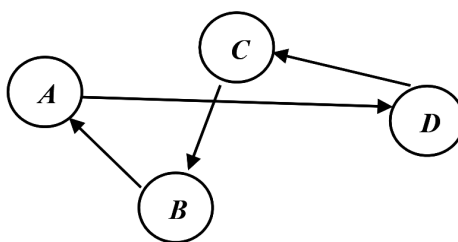


Рис. 3. Пример выполнения доставки грузов беспилотной транспортной системой

В целом задачу организации доставки грузов можно представить в виде графовой модели [4], где есть стартовая точка и точки доставки, в которые беспилотная транспортная система должна попасть, а затем вернуться в стартовую точку для зарядки аккумуляторной батареи и сбора новых грузов. Для решения задачи в случае небольшого количества точек для доставки грузов можно использовать «жадные» оптимизационные алгоритмы (например, алгоритм Дейкстры). За счет его простоты и скорости работы возможно рассчитать допустимый маршрут движения, однако в случае решения задачи с большим количеством точек доставки и большим количеством связей между точками следует рассмотреть другие оптимизационные алгоритмы (например, *метод имитации отжига*, или *муравьиный алгоритм* [6]–[8]). Далее рассмотрим их более подробно.

Метод имитации отжига — это вероятностный метод глобальной оптимизации функций. В анализе данных он известен как популярный метод обучения нейронных сетей, используемый как альтернатива градиентному спуску, которому он уступает в точности, но при этом имеет более высокую скорость сходимости.

Название метода произошло из технологии отжига металла, которая заключается в разогреве и последующем контролируемом охлаждении материала с целью уменьшения дефектов его кристаллической структуры. При нагреве металла его внутренняя энергия возрастает до определенного значения, соответствующего начальному состоянию нейронной сети. При медленном охлаждении (отжиге) с плавным уменьшением температуры тепловые колебания узлов кристаллической решетки около состояния минимума энергии будут плавно уменьшаться, в результате решетка получит более высокую упорядоченность, а энергия системы достигнет глобального минимума. Медленное уменьшение температуры можно интерпретировать как уменьшение вероятности выбора неудачных шагов в пространстве решений. На каждом шаге алгоритм случайным образом выбирает решение, близкое к текущему, оценивает его качество, а затем определяет необходимость либо перехода к нему, либо нахождения в текущем состоянии.

Алгоритм состоит в следующем. Имеет место функция F , которая на основе предыдущего состояния порождает новое, в которое система может перейти или его отбросить. Обозначим его s_c .

1. Прежде всего нужно задать начальное состояние системы. Для этого выбирается любое случайное состояние и задаются начальное и конечное значения температуры t_n, t_k . Далее на каждом k -м шаге $\Delta E = E(s_c) - E(s_{i-1})$.

2. Случайным образом генерируем новое состояние, при котором распределение вероятности должно зависеть от текущего состояния и текущей температуры:

$$s_c = F(s_{i-1}). \quad (1)$$

3. Вычисляем значение функции для сгенерированной точки

$$\Delta E = E(s_c) - E(s_{i-1}). \quad (2)$$

4. Принимаем или не принимаем сгенерированное состояние в качестве текущего. Вероятность этого решения должна зависеть от разности функций сгенерированного и текущего состояний, а также от температуры (чем выше температура, тем больше вероятность принять состояние хуже текущего). Если $\Delta E > 0$, $P(\Delta E) = 100e^{-\Delta E/t_i}$, иначе $s_i = s_c$.

5. Если новое состояние не принято, то генерируем другое, повторяя действия пп. 3, если принято, то переходим к следующей итерации, понизив температуру, $t_k = t_c \cdot \alpha$, где α — коэффициент снижения температуры.

Процесс прекращается при достижении определенной температуры. Это свидетельствует о том, что вещество остыло в точке с минимальной энергией. После завершения выводится результат работы алгоритма. Данный алгоритм позволяет решить проблему с попаданием в локальный минимум функции, но из-за вероятностного характера дает приближенное нахождение глобального минимума, хотя по сравнению с другими вероятностными алгоритмами процесс происходит достаточно быстро и не требует больших вычислительных ресурсов. Далее рассмотрим муравьиный алгоритм.

Муравьиные алгоритмы представляют собой вероятностную жадную эвристику, когда вероятности устанавливаются на основе информации о качестве решения, полученной из предыдущих решений. Идея муравьиного алгоритма состоит в моделировании поведения муравьев, связанного с их способностью быстро находить кратчайший путь от муравейника к источнику пищи и адаптироваться к изменяющимся условиям, находя новый кратчайший путь. При своем движении муравей метит путь феромоном — веществом, которое служит своеобразной «информацией», используемой другими муравьями для выбора пути. Это элементарное правило поведения и определяет способность муравьев находить новый путь в случае, если старый оказался недоступным. Очевидная положительная обратная связь быстро приведет к тому, что кратчайший путь станет единственным маршрутом движения большинства муравьев. Моделирование испарения феромона — отрицательной обратной связи — гарантирует, что найденное локально оптимальное решение не будет единственным — муравьи будут искать и другие пути. При моделировании такого поведения на некотором графе, ребра которого представляют собой возможные пути перемещения муравьев в течение определенного периода времени, наиболее обогащенный феромоном путь по ребрам этого графа будет являться решением задачи, полученным с помощью *муравьиного алгоритма*.

Пусть муравей находится в узле i , а узел j — это один из узлов S_i , доступных для перехода. Обозначим вес ребра, соединяющего узлы i и j , как $w_{i,j}$, а интенсивность феромона на нем — как $t_{i,j}$. Тогда вероятность перехода муравья из i в j составит

$$P_{i,j} = \frac{t_{ij}^\alpha + \frac{1}{w_{ij}^\beta}}{\sum_{l \in S_i} \left(t_{ij}^\alpha + \frac{1}{w_{ij}^\beta} \right)}, \quad (3)$$

где α, β — регулируемые параметры, определяющие важность составляющих (веса ребра и уровня феромонов) при выборе пути.

После того как муравей успешно проходит маршрут, он оставляет на всех пройденных ребрах след, обратно пропорциональный длине пройденного пути:

$$\Delta_{ij} = \begin{cases} \frac{k}{L}, (ij) \in P \\ 0, (ij) \notin P \end{cases}, \quad (4)$$

где L — длина пути;

k — регулируемый параметр.

Кроме того, следы феромона испаряются, т. е. интенсивность феромона на всех ребрах уменьшается на каждой итерации алгоритма. Таким образом, в конце каждой итерации необходимо обновить значения интенсивностей:

$$\tau_{ij}(t+1) = (1-p)\tau_{ij}(t) + \sum_{k=1}^m \tau_{ij,k}(t). \quad (5)$$

Далее следует рассмотреть и сравнить алгоритмы поиска оптимального пути доставки грузов беспилотной транспортной системы.

Результаты исследования (Results of the Research)

В рамках решения данной задачи был построен граф, состоящий из 35 точек, причем из любой точки можно попасть в другую, расстояние между точками 1 м, по оси X расположено пять меток, по оси Y — семь. Доставка во все точки осуществляется из нулевой точки с координатами 0,0. В рамках данной задачи был реализован программный код на основе метода имитации отжига и муравьиного алгоритма. Затем был реализован программный код навигации беспилотной транспортной системы по полю Аркус-маркеров, расположенных согласно конфигурации точек доставки [9]–[12]. Для успешной навигации дрона необходимо просчитать перемещение дрона по данным точкам.

Исходя из того, что в задаче даны координаты точек, в данном случае необходимо вычислять расстояния между двумя точками по формуле

$$z_{ij} = \sqrt{(x_{i0} - x_{i1})^2 + (y_{i0} - y_{i1})^2}. \quad (6)$$

После этого выполнялись вычисления маршрута движения.

Рассмотрим сначала реализацию метода имитации отжига:

while t_start > t_stop:

y = []

x = []

new_dist = 0

while True:

first_index = randint(1, (len(point) — 2))

second_index = randint(1, (len(point) — 2))

if first_index != second_index:

break

first_elem = new_point[first_index]

second_elem = new_point[second_index]

new_point[first_index] = second_elem

new_point[second_index] = first_elem

for i in range(len(new_point)-1):

dist2 = dist[(new_point[i])-1][(new_point[i + 1])-1]

new_dist = new_dist + dist2

p = 100 * math.e ** (—((start_dist — new_dist) / t_start))

rand = randint(1, 100)

В данном фрагменте исходного кода на языке Python приведена реализация выбора случайных точек с дальнейшей их заменой. Для работы данного алгоритма были заданы минимальные и максимальные температуры, максимальная разница между полученным и лучшим решением на предыдущем шаге, а также заданы точки перемещения и матрица расстояний согласно условию задания (рис. 4 и 5).

Исходные данные задания:

```
Начальные точки перемещения:
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 1]
Начальная длина маршрута:
61.529999999999994
```

Рис. 4. Начальные условия перемещения

Начальный график перемещения:

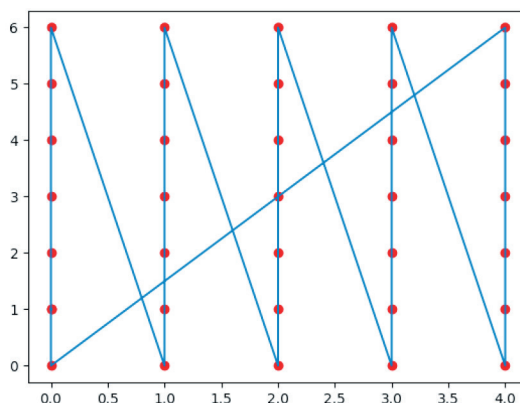


Рис. 5. Изначальный маршрут перемещения

В результате выполнения программного кода были получены следующие результаты (рис. 6 и 7):

```
Метод имитации отжига, итоговое расстояние: 41.300000000000004
Точки перемещения:
[1, 2, 3, 4, 5, 6, 7, 14, 21, 20, 13, 12, 11, 10, 9, 16, 17, 18, 19, 26, 32, 31, 23, 24, 33, 34, 35, 28, 27, 25, 22, 29, 30, 15, 8, 1]
```

Рис. 6. Вывод в терминал результата работы метода имитации отжига

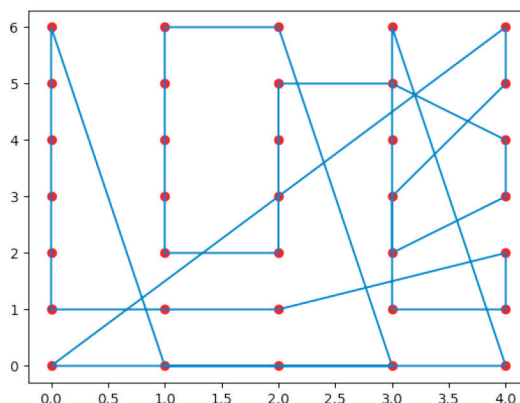


Рис. 7. Маршрут перемещения
при помощи алгоритма имитации отжига

Данный алгоритм позволяет достаточно быстро найти допустимый маршрут с выполнением оптимизации исходных данных. Далее рассмотрим реализацию муравьиного алгоритма. В данном алгоритме были введены переменные, отвечающие за количество итераций поиска маршрутов (му-

равьи), коэффициенты феромона, коэффициент испарения, степени влияния феромонов и исходные данные, аналогичные алгоритму метода имитации отжига.

Фрагмент программного кода:

```
def graph(k):
    global kNach
    x = [(XCoord[kNach], XCoord[k])]
    y = [(YCoord[kNach], YCoord[k])]
    plt.plot(x, y, color = 'red')
    kNach = k
    x = np.array([])
    y = np.array([])

def PPerehoda(k, TownList):
    global Summa
    for i in range(len(TownList)):
        r = int(TownList[i])
        if k!= r:
            Summa = Summa + ((pow(Feramonus[k][r], Alfa)) * (pow((Const/RastBtw[k][r]), Beta)))
    for i in range(len(TownList)):
        r = int(TownList[i])
        if k!= r:
            PMove[k][r] = ((pow(Feramonus[k][r], Alfa)) * (pow((Const/RastBtw[k][r]), Beta))) / Summa
```

В приведенном фрагменте программного кода выполняется расчет испарения феромона в зависимости от пройденного маршрута (городов) [8]–[11]. Если один из маршрутов был использован, то тогда коэффициент феромона увеличивается, вместе с этим растет вероятность повторного выбора данного маршрута, иначе данный маршрут постепенно будет угасать. Также в данном фрагменте программного кода строятся предварительные графики построения маршрута беспилотной транспортной системы для визуализации работы алгоритма. Результаты работы программы приведены на рис. 8 и 9.

```
Муравьиный алгоритм
минимальное расстояние:
35.414 на маршруте:
[ 0.  7. 14. 21. 28. 29. 30. 23. 22. 15.  8.  9. 16. 17. 24. 31. 32. 25.
18. 19. 26. 33. 34. 27. 20. 13.  6.  5. 12. 11.  4.  3. 10.  2.  1.]
```

Рис. 8. Вывод в терминал результата работы муравьиного алгоритма

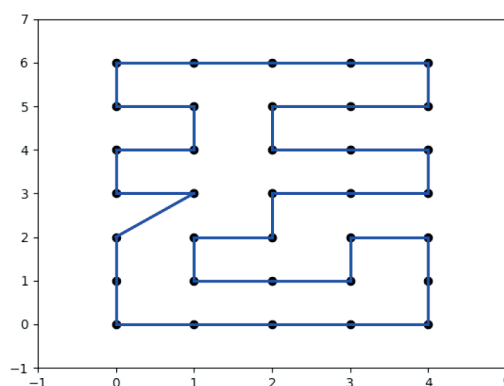


Рис. 9. Результат построения маршрута перемещения при помощи муравьиного алгоритма

В результате программной реализации оба алгоритма помогли оптимизировать маршрут доставки грузов беспилотной транспортной системой, имея при этом каждый свои особенности.

Обсуждение (Discussion)

В рамках решения задачи доставки грузов в небольшое количество точек для доставки грузов можно использовать «жадные» оптимизационные алгоритмы (например, алгоритм Дейкстры), однако в данном примере приведен сложный граф с большим количеством точек, при этом из любой одной точки можно попасть в любую другую, что накладывает свои особенности, поэтому были выбраны метод имитации отжига и муравьиный алгоритм для решения транспортной задачи.

Метод имитации отжига показал свою относительную простоту с точки зрения программной реализации, а также с точки зрения требований к ресурсам ЭВМ, однако, из-за особенностей поиска допустимого маршрута движения, данный алгоритм обладает низкой сходимостью. В данном случае было выполнено три запуска оптимизационного алгоритма и в качестве примера показан лучший результат. Оставшиеся два запуска показали результат, приведенный на рис. 10.

```
Метод имитации отжига, итоговое расстояние: 48.730000000000004
Точки перемещения:
[1, 2, 3, 4, 5, 6, 7, 10, 9, 8, 15, 16, 29, 22, 23, 24, 18, 19, 13, 14, 21, 20, 12, 11, 27, 34, 35, 28, 25, 30, 31, 32, 33, 26, 17, 1]
Метод имитации отжига, итоговое расстояние: 47.33
Точки перемещения:
[1, 2, 3, 4, 5, 6, 7, 12, 11, 10, 9, 8, 15, 22, 23, 16, 17, 24, 25, 20, 21, 14, 13, 19, 26, 27, 28, 35, 29, 30, 31, 32, 33, 34, 18, 1]
```

Рис. 10. Результат работы метода имитации отжига
после нескольких запусков

Муравьиный алгоритм, напротив, показал высокую сходимость результатов и сумел найти лучший маршрут движения беспилотной транспортной системы, однако время выполнения поиска маршрута было довольно продолжительным. Обладая относительно простым математическим описанием, тем не менее программная реализация оказалась достаточно сложной из-за применения подхода поиска оптимального маршрута несколькими «муравьями», а также пересчета вероятностей движения по старым маршрутам исходя из коэффициента испарения феромона. Для ускорения работы оптимизации, возможно, следует скомбинировать алгоритмы поиска. Несмотря на то, что метод имитации отжига обладает низкой сходимостью, однако решение обычно находится близко к экстремуму функции, а муравьиный алгоритм позволит оптимизировать полученное решение в течение короткого периода времени для беспилотных транспортных систем.

Результаты (Results)

Представленные алгоритмы позволяют оптимизировать время и маршрут доставки грузов за счет сокращения общего объема пути. Для проверки эффективности данных операций был реализован программный код автономного движения беспилотной транспортной системы в симуляционной среде Gazebo с созданием тестового поля, соответствующего условиям данной задачи. Среда симуляции позволяет запускать и отлаживать программы автономного движения, используя большинство функций, доступных на реальном дроне, без риска повреждения оборудования. Такой подход позволяет оперативно вносить изменения в программный код, проверять его корректность и улучшать качество.

Под автономным управлением беспилотной транспортной системы подразумевается выполнение поставленной миссии без участия оператора (пилота). Навигация беспилотной транспортной системы базируется на технологии машинного зрения и поле Агисо-маркеров. Камера считывает изображение, находит на нем Агисо-маркеры (рис. 11), каждый из которых имеет свой уникальный идентификатор и опираясь на эту информацию, беспилотная транспортная система позиционирует себя в пространстве.



Рис. 11. Поле Aruco-маркеров

В памяти беспилотной транспортной системы указано расположение всех маркеров согласно условиям задания и возможно выполнение анализа лучшего полученного результата при помощи муравьиного алгоритма относительно исходных данных. Результаты сравнения представлены на скриншотах (рис. 12):

```
clover@clover-dev:~$ /bin/python3 /home/clover/Desktop/1.py
Время выполнения полета в симуляционной среде с применением муравьиного алгоритма:
0:03:20.805222
clover@clover-dev:~$ /bin/python3 /home/clover/Desktop/1.py
Время выполнения полета в симуляционной среде без применения оптимизационных алгоритмов:
0:03:38.562681
```

Рис. 12. Результаты работы программы

Фрагмент программного кода:

```
def navigate_wait(x=0, y=0, z=1, yaw=float('nan'), speed=0.5, frame_id='', auto_arm=False,
tolerance=0.2):
    navigate(x=x, y=y, z=z, yaw=yaw, speed=speed, frame_id=frame_id, auto_arm=auto_arm)
    while not rospy.is_shutdown():
        telem = get_telemetry(frame_id='navigate_target')
        if math.sqrt(telem.x ** 2 + telem.y ** 2 + telem.z ** 2) < tolerance:
            break
        rospy.sleep(0.2)
    navigate(x=0, y=0, z=1, frame_id='body', auto_arm=True)
    rospy.sleep(3)
    points=
    [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,0]
    new_points = [7, 14, 15, 22, 21, 28, 29, 30, 23, 24, 31, 32, 33, 34, 27, 20, 13, 6, 5, 4, 11, 12, 19, 26, 25,
    18, 17, 16, 9, 10, 3, 2, 1, 8, 0]
```

В данном фрагменте программного кода приведен пример функции навигации беспилотной системы с применением id Aruco-маркеров. В данном случае это точки, которые должно посетить беспилотное транспортное средство. После выполнения программного кода в терминал выводится итоговое время движения беспилотной транспортной системы по всем указанным координатам. В зависимости от сложности транспортной миссии и разницы в расстояниях между точками доставки разница между исходными данными и оптимизированным маршрутом будет увеличиваться. Данные алгоритмы возможно применять также для беспилотных морских судов. В дальнейшем данную задачу можно усложнить, добавляя запрещенные маршруты движения, которые будут динамически появляться по мере выполнения автономного задания по доставке грузов беспилотными транспортными системами по заданным координатам. В данной работе были продемонстрированы методы оптимизации и показана их важность для внедрения в беспилотные транспортные системы.

Выводы (Summary)

На основе проведенного исследования моделей и методов маршрутизации беспилотных транспортных систем можно сделать следующие выводы:

1. Сфера беспилотных транспортных систем в последние годы активно развивается, что отражается в наукометрическом исследовании при помощи программного обеспечения VOSviewer. В работах, посвященных глобальной теме изучения беспилотных транспортных систем, выполнены также и другие важные исследования, связанные с навигацией подобных систем, работой машинного зрения, элементов искусственного интеллекта и особенностями организации связи между группой беспилотных систем.

2. В настоящее время проводятся испытания данных систем в сфере транспортной логистики, однако следует рассматривать оптимизационные алгоритмы, которые позволяют дополнительно сократить время доставки грузов, сэкономить аккумуляторные батареи, а также построить оптимальный маршрут с учетом ограничительных мер.

3. Для оптимизации исходного маршрута движения были рассмотрены метод имитации отжига и муравьиный алгоритм, которые позволили оптимизировать путь беспилотной транспортной системы. Данные алгоритмы были реализованы на языке программирования Python с построением графика нового маршрута движения. Наиболее точным оказался муравьиный алгоритм. В дальнейшем полученный вариант был проверен на оптимальность в симуляционной среде Gazebo.

4. Для имитации доставки грузов в множество точек был сформирован тестовый полигон с размещением опорных точек — Агисо-маркеров, к которым должна добраться беспилотная транспортная система и выполнить доставку груза. Для реализации данной задачи был реализован программный код автономного перемещения и выполнено сравнение двух маршрутов: исходного и оптимизированного по времени выполнения доставки. Исходя из полученных результатов муравьиный алгоритм позволил построить более короткий и быстрый маршрут движения беспилотной транспортной системы. В случае расширения задачи, добавления закрытых зон данный алгоритм позволит построить допустимый маршрут движения.

СПИСОК ЛИТЕРАТУРЫ

1. Костин А. С. Эксплуатация беспилотных авиационных систем / А. С. Костин, Н. Н. Майоров, Т. Ю. Карпова. — СПб.: ГУАП, 2021. — 169 с.
2. Kostin A. S. Models and methods for implementing the autonomous performance of transportation tasks using a drone / A. S. Kostin // 2021 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF). — IEEE, 2021. — Pp. 1–4. DOI: 10.1109/WECONF51603.2021.9470584.
3. Костин А. С. Вопросы современного развития рынка беспилотных летательных аппаратов / А. С. Костин, Н. В. Богатов // Системный анализ и логистика. — 2019. — № 4 (22). — С. 65–72.
4. Прокушев Л. А. Дискретная математика / Л. А. Прокушев. — СПб.: ГУАП, 2000. — 80 с.
5. Gonzalez-R P. L. Truck-drone team logistics: A heuristic approach to multi-drop route planning / P. L. Gonzalez-R, D. Canca, J. L. Andrade-Pineda, M. Calle, J. M. Leon-Blanco // Transportation Research Part C: Emerging Technologies. — 2020. — Vol. 114. — Pp. 657–680. DOI: 10.1016/j.trc.2020.02.030.
6. Костин А. С. Разработка автоматизированных решений для исследования вариантов маршрутов доставки при совместном использовании транспортного средства и беспилотной авиационной системы в границах города / А. С. Костин, Н. Н. Майоров // Известия Тульского государственного университета. Технические науки. — 2022. — № 7. — С. 348–356. DOI: 10.24412/2071-6168-2022-7-348-357.
7. Свидетельство о государственной регистрации программы для ЭВМ № 2022615497 Российская Федерация. Программа автономного полета беспилотной авиационной системы для реализации многоадресной доставки грузов / Н. Н. Майоров, А. С. Костин, Е. А. Вознесенский; заяв. и патентообл. Санкт-Петербургский государственный университет аэрокосмического приборостроения. — № 2022615305; заявл. 31.03.2022; опубл. 31.03.2022.
8. Свидетельство о государственной регистрации программы для ЭВМ № 2023612873 Российская Федерация. Программа формирования допустимого маршрута перемещения задачи коммивояжера на основе муравьиного алгоритма / Н. Н. Майоров, А. С. Костин, Д. В. Кучко; заяв. и патентообл. Санкт-Петербургский государственный университет аэрокосмического приборостроения. — № 2023611992; заявл. 08.02.2023; опубл. 08.02.2023.

9. Engel J. Accurate Figure Flying with a Quadcopter Using Onboard Visual and Inertial Sensing / J. Engel, J. Sturm, D. Cremers // *Proc. of the Workshop on Visual Control of Mobile Robots (ViCoMoR) at the IEEE/ RJS Intern. Conference on Intelligent Robot Systems (IROS)*. Vilamoura, Algarve, Portugal. — 2012. — Pp. 43–48.
10. Drexel M. Synchronization in vehicle routing — a survey of VRPs with multiple synchronization constraints / M. Drexel // *Transportation Science*. — 2012. — Vol. 46. — Is. 3. — Pp. 297–316. DOI: 10.1287/trsc.1110.0400.
11. Kim J. Accurate modeling and robust hovering control for a quad-rotor VTOL aircraft / J. Kim, M. S. Kang, S. Park // *Selected papers from the 2nd International Symposium on UAVs, Reno, Nevada, USA June 8–10, 2009*. — Springer Netherlands, 2010. — Pp. 9–26. DOI: 10.1007/978-90-481-8764-5_2.
12. Outay F. Applications of unmanned aerial vehicle (UAV) in road safety, traffic and highway infrastructure management: Recent advances and challenges / F. Outay, H. A. Mengash, M. Adnan // *Transportation research part A: policy and practice*. — 2020. — Vol. 141. — Pp. 116–129. DOI: 10.1016/j.tra.2020.09.018.

REFERENCES

1. Kostin, A. S., N. N. Maiorov, and T. Yu. Karpova. *Ekspluatatsiya bespilotnykh aviatsionnykh sistem*. SPb.: GUAP, 2021.
2. Kostin, A. S. “Models and methods for implementing the autonomous performance of transportation tasks using a drone.” *2021 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF)*. IEEE, 2021. DOI: 10.1109/WECONF51603.2021.9470584.
3. Kostin, Anton Sergeevich, and Nikita Vladimirovich Bogatov. “Questions of modern development of the drones aircraft market.” *Sistemnyi analiz i logistika* 4(22) (2019): 65–72.
4. Prokushev, L. A. *Diskretnaya matematika*. SPb.: GUAP, 2000.
5. Gonzalez-R, Pedro L., David Canca, Jose L. Andrade-Pineda, Marcos Calle, and Jose M. Leon-Blanco. “Truck-drone team logistics: A heuristic approach to multi-drop route planning.” *Transportation Research Part C: Emerging Technologies* 114 (2020): 657–680. DOI: 10.1016/j.trc.2020.02.030.
6. Kostin, Anton Sergeyevich, and Nikolai Nikolaevich Maiorov. “Development of automated solutions for the research of delivery route options in the joint use of a vehicle and an unmanned aerial system within the boundaries of the city.” *Izvestiya Tula State University* 7 (2022): 348–356. DOI: 10.24412/2071-6168-2022-7-348-357.
7. Maiorov, N. N., A. S. Kostin, and E. A. Voznesenskii. RU 2022615497. Programma avtonomnogo poleta bespilotnoi aviatsionnoi sistemy dlya realizatsii mnogoadresnoi dostavki грузов. Russian Federation, assignee. Publ. 31 March 2022.
8. Maiorov, N. N., A. S. Kostin, and D. V. Kuchko. RU 2023612873. Programma formirovaniya dopustimogo marshruta peremeshcheniya zadachi kommivoyazhera na osnove murav'inogo algoritma. Russian Federation, assignee. Publ. 8 Feb. 2023.
9. Engel, Jakob, Jurgen Sturm, and Daniel Cremers. “Accurate Figure Flying with a Quadcopter Using Onboard Visual and Inertial Sensing.” *Proc. of the Workshop on Visual Control of Mobile Robots (ViCoMoR) at the IEEE/ RJS Intern. Conference on Intelligent Robot Systems (IROS)*. 2012. 43–48.
10. Drexel, Michael. “Synchronization in vehicle routing — a survey of VRPs with multiple synchronization constraints.” *Transportation Science* 46.3 (2012): 297–316. DOI: 10.1287/trsc.1110.0400.
11. Kim, Jinhyun, Min-Sung Kang, and Sangdeok Park. “Accurate modeling and robust hovering control for a quad-rotor VTOL aircraft.” *Selected papers from the 2nd International Symposium on UAVs, Reno, Nevada, USA June 8–10, 2009*. Springer Netherlands, 2010. DOI: 10.1007/978-90-481-8764-5_2.
12. Outay, Fatma, Hanan Abdullah Mengash, and Muhammad Adnan. “Applications of unmanned aerial vehicle (UAV) in road safety, traffic and highway infrastructure management: Recent advances and challenges.” *Transportation research part A: policy and practice* 141 (2020): 116–129. DOI: 10.1016/j.tra.2020.09.018.

ИНФОРМАЦИЯ ОБ АВТОРАХ

Костин Антон Сергеевич — аспирант, ассистент
Научный руководитель:
Майоров Николай Николаевич
ФГАОУ ВО «Санкт-Петербургский
государственный университет аэрокосмического
приборостроения»
190000, Российская Федерация, Санкт-Петербург,
ул. Большая Морская, 67, лит. А
e-mail: anton13258@mail.ru

INFORMATION ABOUT THE AUTHORS

Kostin, Anton S. — Postgraduate, assistant
Supervisor:
Maiorov, Nikolai N.
Saint-Petersburg State University
of Aerospace Instrumentation
67/A Bol'shaya Morskaya Str.,
St. Petersburg, 190000,
Russian Federation
email: anton13258@mail.ru

Майоров Николай Николаевич —
доктор технических наук, доцент
ФГАОУ ВО «Санкт-Петербургский
государственный университет аэрокосмического
приборостроения»
190000, Российская Федерация, Санкт-Петербург,
ул. Большая Морская, 67, лит. А
e-mail: nmsoft@yandex.ru

Maiorov, Nikolai N. —
Dr. of Technical Sciences, associate professor
Saint-Petersburg State University
of Aerospace Instrumentation
67/A Bol'shaya Morskaya Str.,
St. Petersburg, 190000,
Russian Federation
e-mail: nmsoft@yandex.ru

*Статья поступила в редакцию 12 апреля 2023 г.
Received: April 12, 2023.*