

УДК: 004.051, 004.021, 004.023

Сравнение результатов применения различных эволюционных алгоритмов для решения задачи оптимизации маршрута беспилотных аппаратов

А. А. Федина^а, А. И. Нурғалиев^б, Д. А. Скворцова^с

ФГБОУ ВО «Московский государственный технический университет имени Н. Э. Баумана
(национальный исследовательский университет)»,
Россия, 105005, г. Москва, 2-я Бауманская ул., 5, стр. 1

E-mail: ^а faa.bmstu@yandex.ru, ^б fkmf2001@mail.ru, ^с darya_skv@mail.ru

*Получено 15.11.2021, после доработки — 24.01.2022.
Принято к публикации 02.02.2022.*

В данной работе проводится сравнительный анализ точного и эвристических алгоритмов, представленных методом ветвей и границ, генетическим и муравьиным алгоритмами соответственно, для поиска оптимального решения задачи коммивояжера на примере работа-курьера. Целью работы является определение времени работы, длины полученного маршрута и объема памяти, необходимого для работы программы, при использовании метода ветвей и границ и эволюционных эвристических алгоритмов. Также определяется наиболее целесообразный из перечисленных методов для применения в заданных условиях. В настоящей статье используются материалы проведенного исследования, реализованного в формате программы для ЭВМ, программный код для которой реализован на языке Python. В ходе исследования был выбран ряд критериев применимости алгоритмов (время работы программы, длина построенного маршрута и объем необходимой для работы программы памяти), получены результаты работы алгоритмов в заданных условиях и сделаны выводы о степени целесообразности применения того или иного алгоритма в различных заданных условиях работы работа-курьера. В ходе исследования выяснилось, что для малого количества точек (≤ 10) метод ветвей и границ является наиболее предпочтительным, так как находит оптимальное решение быстрее. Однако при вычислении маршрута этим методом, при условии увеличения точек более 10, время работы растет экспоненциально. В таком случае более эффективные результаты дает эвристический подход с использованием генетического и муравьиного алгоритмов. При этом муравьиный алгоритм отличается решениями, наиболее близкими к эталонным, при увеличении точек более 16. Относительным недостатком его является наибольшая ресурсоемкость среди рассматриваемых алгоритмов. Генетический алгоритм дает схожие результаты, но при увеличении точек более 16 растет длина найденного маршрута относительно эталонного. Преимущество генетического алгоритма — его меньшая ресурсоемкость по сравнению с другими алгоритмами.

Практическая значимость данной статьи заключается в потенциальной возможности использования полученных результатов для оптимального решения логистических задач автоматизированной системой в различных сферах: складская логистика, транспортная логистика, логистика «последней мили» и т. д.

Ключевые слова: беспилотные аппараты, алгоритмы оптимизации, метод ветвей и границ, генетический алгоритм, муравьиный алгоритм, задача коммивояжера, логистические системы

UDC: 004.051, 004.021, 004.023

Comparison of the results of using various evolution algorithms to solve the problem of route optimization of unmanned vehicles

A. A. Fedina^a, A. I. Nurgaliev^b, D. A. Skvortsova^c

Bauman Moscow State Technical University (National Research University),
5/1, 2nd Bauman st., Moscow, 105005, Russia

E-mail: ^a faa.bmstu@yandex.ru, ^b fkmaf2001@mail.ru, ^c darya_skv@mail.ru

Received 15.11.2021, after completion — 24.01.2022.

Accepted for publication 02.02.2022.

In this paper, a comparative analysis of the exact and heuristic algorithms presented by the method of branches and boundaries, genetic and ant algorithms, respectively, is carried out to find the optimal solution to the traveling salesman problem using the example of a courier robot. The purpose of the work is to determine the running time, the length of the obtained route and the amount of memory required for the program to work, using the method of branches and boundaries and evolutionary heuristic algorithms. Also, the most appropriate of the listed methods for use in the specified conditions is determined. This article uses the materials of the conducted research, implemented in the format of a computer program, the program code for which is implemented in Python. In the course of the study, a number of criteria for the applicability of algorithms were selected (the time of the program, the length of the constructed route and the amount of memory necessary for the program to work), the results of the algorithms were obtained under specified conditions and conclusions were drawn about the degree of expediency of using one or another algorithm in various specified conditions of the courier robot. During the study, it turned out that for a small number of points ≤ 10 , the method of branches and boundaries is the most preferable, since it finds the optimal solution faster. However, when calculating the route by this method, provided that the points increase by more than 10, the operating time increases exponentially. In this case, more effective results are obtained by a heuristic approach using a genetic and ant algorithm. At the same time, the ant algorithm is distinguished by solutions that are closest to the reference ones and with an increase of more than 16 points. Its relative disadvantage is the greatest resource intensity among the considered algorithms. The genetic algorithm gives similar results, but after increasing the points more than 16, the length of the found route increases relative to the reference one. The advantage of the genetic algorithm is its lower resource intensity compared to other algorithms.

The practical significance of this article lies in the potential possibility of using the results obtained for the optimal solution of logistics problems by an automated system in various fields: warehouse logistics, transport logistics, «last mile» logistics, etc.

Keywords: unmanned vehicles, optimization algorithms, branches and bounds method, genetic algorithm, ant algorithm, traveling salesman problem, logistics systems

Citation: *Computer Research and Modeling*, 2022, vol. 14, no. 1, pp. 45–62 (Russian).

1. Введение

С каждым годом присутствие роботов в сфере транспортной логистики стремительно увеличивается. Как отмечают исследователи из Оксфордского университета [Снабжение: роботы и дроны. . . , 2020], человечество находится на пороге новой логистической революции. Компании в погоне за снижением издержек и как следствие за снижением себестоимости конечного продукта заменяют присутствие людей роботами. Беспилотные технологии становятся одним из главных драйверов развития логистики. Отсюда следует, что для достижения успеха в технологической гонке компаниям необходимо адаптироваться к быстро изменяющимся внешним условиям. Массовая автоматизация и роботизация процессов отчетливо указывают на общую тенденцию снижения потребности в низкоквалифицированной рабочей силе.

К аналогичному выводу пришли аналитики консалтинговой компании McKinsey. По заверениям специалистов, практически все трудоспособное население Европы в ближайшие годы столкнется с серьезными изменениями рынка труда, связанными в первую очередь с непрерывным развитием технологий, в частности со стремительно развивающимся рынком беспилотных технологий.

По данным исследователей, продолжающийся процесс повсеместной автоматизации и проникновение цифровых технологий во все сферы деятельности человека ставят под угрозу 51 млн рабочих мест в регионе. Исходя из подсчетов McKinsey Global Institute, можно сделать вывод о потенциальной возможности исчезновения от 400 до 800 миллионов рабочих мест к 2030 году [Manyika, 2017; Robots Threaten Bigger Slice of Jobs in US, 2017].

Стремительное развитие получила сфера роботов-курьеров. По заявлению Кеннета Линдгрена, генерального директора «KONE Россия», рынок данного способа доставки к 2027 году достигнет объема в \$38,3 млрд. В свою очередь, в своем исследовании аналитическая компания IDTechEx говорит о том, что к 2040 году емкость рынка роботов-курьеров может достигнуть \$290 млрд, объяснением чему служит удешевление технологий и повышение грузоподъемности [Роботы-курьеры помогут создать безбарьерную среду, 2021].

Существует так называемая проблема последней мили, связанная с доставкой товара, например, от распределительного центра или маркетплейса до двери клиента. Данный этап является наиболее важным и дорогим звеном транспортной логистики. Согласно исследованиям международной консалтинговой компании Cushman & Wakefield, затраты на последнюю милю могут достигать более 50 % общих расходов цепочки поставок [Last Link: Quantifying the Cost, 2019]. Одним из возможных решений может быть использование уличных дронов-курьеров — как наземных, так и воздушных. Многие компании активно разрабатывают и тестируют, а некоторые уже ввели в эксплуатацию таких роботов. Компания FedEx представила свою разработку SameDay и развернула испытания робота-курьера в марте 2019 года. Разработка получила усовершенствованную подвеску, дающую возможность роботу взбираться по лестницам, а аккумулятор позволяет преодолеть путь длиной в 16 км. Грузоподъемность такого робота — 45 кг.

В России технологии роботов-курьеров использует компания «Сбербанк». Они тестируют пилотную версию разработки под названием «Глеб» (это робот для транспортировки документов по офису). В качестве курьеров роботов стала использовать компания «Яндекс»: они назвали свою разработку «Яндекс.Ровер» и в 2019 году начали ее тестирование в районах Москвы и Казани. С апреля 2020 года услугами «Яндекс.Роверов» пользовались в Сколково для доставки документов и посылок. В 2021 году разработка «Яндекса» доставляет заказы из ресторанов и магазинов в московском районе Хамовники. К текущему времени, по заявлению компании «Яндекс», роботы-курьеры доставили более 4 тысяч заказов [Робот-курьер Яндекс.Ровер. . . , 2020].

2. Постановка задачи

Основной задачей роботов-курьеров является доставка товаров из точки отправки в точку заказа за минимальное время по кратчайшему маршруту. Так как роботы работают автономно и без участия человека, самостоятельно распознавая объекты внешнего мира и прокладывая себе путь, возникает проблема оптимизации такого маршрута.

Поиск оптимального маршрута сводится к решению задачи коммивояжера, то есть к поиску кратчайшего гамильтонова цикла в графе. Данная задача является задачей комбинаторной оптимизации и заключается в поиске кратчайшего маршрута в полном взвешенном графе G [Gresco, 2008] с N узлами и $N(N - 1)$ ребрами, так что начальный и конечный узлы идентичны, а все остальные узлы посещаются единожды. В рассматриваемой задаче вершинами графов служат основные точки — адреса домов, а в качестве ребер выступают дорожные улицы. Вспомогательными точками будут являться перекрестки, они будут необходимы для проведения расчетов.

В данной работе в качестве примера предлагается рассмотреть уличного робота-курьера. В качестве места действия выбран микрорайон г. Москвы (рис. 1). Основной задачей робота является доставка товаров из точки отправки (маркетплейса) по нескольким адресам с наименьшими временными затратами.

В целях упрощения изложения опускается взаимодействие робота-курьера с людьми и любыми преградами, а также игнорируются ограничения, накладываемые правилами дорожного движения. Также в данной работе не рассматриваются ограничения, связанные с емкостью аккумулятора и запасом хода робота, а также не учитываются габариты и вес доставляемых объектов и грузоподъемность робота-курьера.

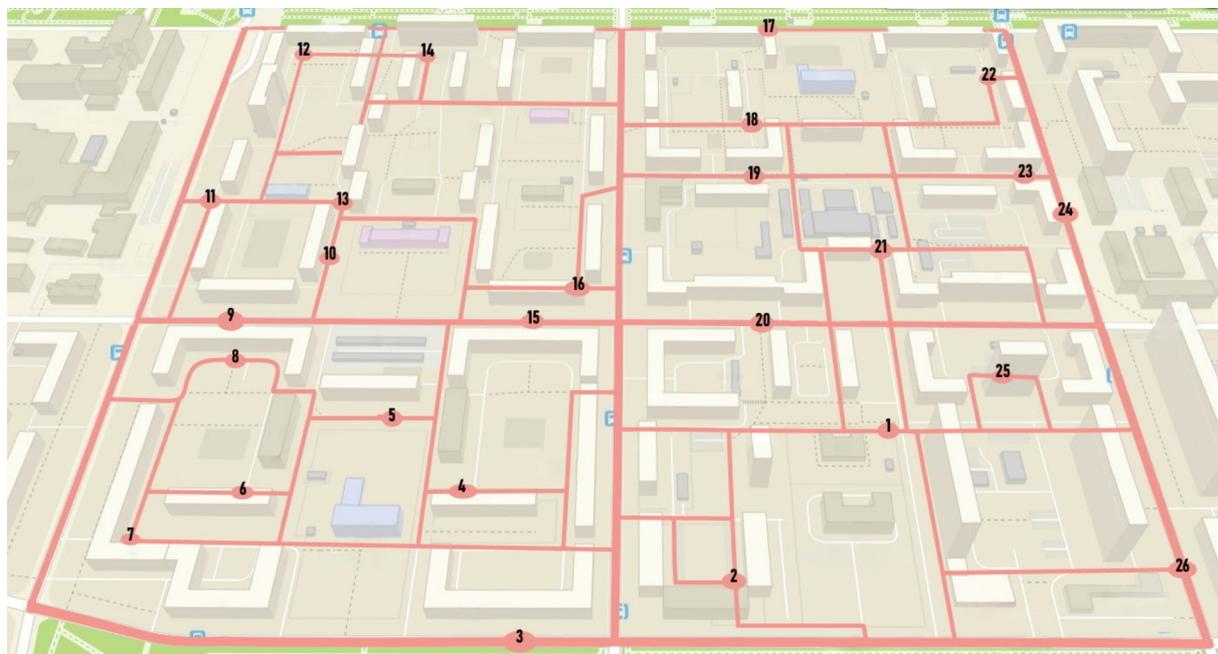


Рис. 1. Схема микрорайона, по которому будет передвигаться робот-курьер

На основе схемы микрорайона составлена матрица расстояний, содержащая 26 точек-адресов, откуда поступил заказ и которые необходимо объехать роботу-курьеру.

3. Математическая модель

Матрицу расстояний, основанную на схеме микрорайона, можно представить в виде полного взвешенного графа $G(X, V)$ [Gutin, Punnen, 2006] порядка N , где $N = \{x_1, x_2, \dots, x_N\}$ — множество вершин, а $V \subseteq X \times X$ — множество ребер. Оптимальным маршрутом в таком случае будет гамильтонов цикл, имеющий наименьший суммарный вес входящих в него ребер.

Формально описать задачу можно следующим образом:

$$F = \sum_{i=1}^m \sum_{j=1}^N d_{ij} x_{ij} \longrightarrow \min, \quad (1)$$

$$\sum_{j=1}^m x_{ij} = 1; \quad j = 1, \dots, N, \quad (2)$$

$$\sum_{i=1}^N x_{ij} = 1; \quad i = 1, \dots, m, \quad (3)$$

$$x_{ij} \geq 0 \quad \{i = \overline{1, m}; j = \overline{1, N}\}, \quad (4)$$

где d_{ij} — расстояние между пунктами i и j , а x_{ij} обозначает двоичные переменные:

$$x_{ij} = \begin{cases} 1, & \text{если коммивояжер переправляется из пункта } i \text{ в пункт } j, \\ 0, & \text{если коммивояжер не переправляется из пункта } i \text{ в пункт } j. \end{cases}$$

Поскольку переменные x_{ij} могут принимать только значение 0 или 1, эта проблема имеет конечное число решений.

Для решения задачи коммивояжера необходимо найти переменные $\{i = \overline{1, m}; j = \overline{1, N}\}$, удовлетворяющие системе ограничений (2)–(3), условиям неотрицательности (4) и обеспечивающие минимум целевой функции (1).

Данная задача является NP-трудной, ее решение зависит от количества вершин N , так как число возможных маршрутов равно числу способов упорядочения вершин графа ($N!$). Решение задачи коммивояжера требует использования эффективных алгоритмов, способных за адекватное время найти приближенный к оптимальному маршрут.

4. Обоснование выбора алгоритмов

Задача коммивояжера является одной и наиболее известных NP-сложных задач, для ее решения не существует точного алгоритма. Существуют разные способы классификации алгоритмов, но основной характеристикой является способность найти точное решение. В данной работе предлагается рассмотреть точные и эвристические [Agatz et al., 2018] методы.

Для поиска оптимального решения в задаче коммивояжера время является экспоненциальным. Исходя из этого, для решения таких задач могут использоваться эвристические методы, главным преимуществом которых является возможность ограничить перебор вариантов решения, т. е. сокращать число вариантов, изучаемых перед выбором окончательного решения.

Эвристические методы обычно существенно быстрее точных, но они не позволяют гарантировать оптимальность найденного решения. Точные методы, в свою очередь, не только находят некоторое решение, но и доказывают его оптимальность. В процессе проведения экспериментов результаты, полученные точным методом, будут считаться эталонными ввиду обеспечения наибольшей точности.

В настоящей статье подробно рассмотрены оба варианта классификации методов решения задачи коммивояжера: точный метод реализуется через метод ветвей и границ, эвристический метод — через генетический и муравьиный алгоритмы. Основными задачами данной работы являются сравнение разных методов и выбор оптимального для решения поставленной задачи.

5. Описание выбранных алгоритмов для поиска оптимального маршрута

Основной идеей *метода ветвей и границ* [Подшивалов и др., 2017; Nilsson, 2013; Костюк, 2013] (далее — МВГ), в отличие от метода полного перебора, является избавление от подмножества решений, которые заведомо не могут содержать оптимальное решение [Костюк, 2013]. Частично построенные маршруты отсекаются в том случае, если их оценка больше или равна длине ранее построенного полного наилучшего маршрута. При поиске оптимального решения строится бинарное дерево. На каждом шаге метода производятся анализ элементов подмножества и оценка текущего маршрута с целью выявления оптимального решения.

Метод является развитием метода полного перебора, в отличие от последнего — с отсеком подмножеств допустимых решений, заведомо не содержащих оптимальных решений, чем лучшее, найденное алгоритмом на данный момент.

Основным преимуществом МВГ является то, что накопление ошибок менее значительное, чем у других алгоритмов, в силу того, что решение идет по разным ветвям. Более того, при принудительной остановке процесса решения высока вероятность получения целочисленного результата, но без установления его оптимальности.

Значительным недостатком МВГ является отсутствие признака оптимальности. Оптимальное решение может быть получено задолго до остановки алгоритма, но обнаружить это в общем случае нельзя. Оптимальность устанавливается только при исчерпании списка задач.

Решение задачи коммивояжера данным алгоритмом начинается с построения матрицы расстояний G_0 размерности $N \times N$, в которой исходными данными будут являться расстояния между пунктами C_{ij} . Далее находятся минимальные значения по строкам (d_i) и столбцам (d_j), проводится редукция матрицы по строкам (5) и столбцам (6):

$$C_{ij} = C_{ij} - d_i, \quad (5)$$

$$C_{ij} = C_{ij} - d_j. \quad (6)$$

После редукции находится корневая нижняя граница H_0 . Для этого вычисляется сумма констант приведения по формуле

$$H_k = \sum_{i=1}^N d_i + \sum_{j=1}^N d_j. \quad (7)$$

Одновременно с этим начинается построение дерева решений – графа, где:

- вершины — это ключевые решения по включению/невключению в итоговый маршрут рассматриваемых отрезков пути;
- ветви — ребра альтернативных вариантов решения;
- корень — локальная нижняя граница H_k , она же — минимальная длина маршрута.

Далее вычисляются оценки нулевых клеток приведенной матрицы по формуле (8) и среди них выбирается максимальная оценка Q :

$$Q_{ij} = \min(d_i) + \min(d_j), \quad (8)$$

где Q_{ij} — это штрафы за неиспользование тех отрезков маршрута, которым соответствуют нулевые клетки.

Матрица редуцируется в соответствии с клеткой, имеющей оценку Q . Одновременно с получением матрицы происходит ветвление дерева решений: первая ветвь — это решения, включенные в маршрут, а вторая — решения, не включенные в маршрут. Для каждой ветви вычисляется новая локальная нижняя граница по формуле (9):

$$H_k = \begin{cases} H_{k-1} + \sum_{i=1}^N d_i + \sum_{j=1}^N d_j & \text{— для первой ветви,} \\ H_{k-1} + Q_{ij} & \text{— для второй ветви.} \end{cases} \quad (9)$$

Далее производится выбор:

- 1) либо выбирается ветвь, которая включает минимальную нижнюю локальную границу;
- 2) либо выбирается ветвь, которая не включает минимальную нижнюю локальную границу, искомый отрезок исключается из маршрута.

Ветвление дерева продолжается, пока не будет найдено оптимальное решение, включающее в себя все точки из начальной матрицы расстояний. Его длина вычисляется как сумма всех C_{ij} , включенных в маршрут.

Блок-схема МВГ, отражающая процесс его реализации, представлена на рис. 2.

Генетический алгоритм [Кремер, Подвальный, 2012; Чмырь, 2018] (далее — ГА) представляет собой эвристический алгоритм, основанный на эволюционном признаке выживания наиболее приспособленных особей. Данный алгоритм применяет методы, аналогичные естественному отбору в природе, например скрещивание, мутация, кроссинговер. Наиболее приспособленные особи имеют выше шансы на выживание и продолжение рода, и, как следствие, потомство в среднем будет более приспособленным, нежели предыдущее поколение [Razali, Geraghty, 2011].

На начальном этапе случайным образом создается некоторое количество начальных векторов, называемых начальной популяцией. Далее к маршрутам множества применяются операции скрещивания и мутации, целью которых является создание нового «поколения». Исключающим фактором попадания в локальный экстремум является тот факт, что ГА использует одновременно несколько точек поискового пространства.

Достоинствами ГА служат большое число свободных параметров, позволяющих эффективно встраивать эвристики, реализация отбора лучших особей методом градиентного спуска и возможность распараллеливания алгоритма.

В качестве недостатка ГА выступает недоказанность сходимости.

Определим основные понятия, используемые в ГА:

- особь — искомый оптимальный путь $S_{\text{опт}}$;
- популяция — набор всех маршрутов $f(x) = F(S_1, S_2, \dots, S_k)$;
- хромосома — любой маршрут S_k ;
- гены — точки на карте (рис. 1) от A до k , которые проходит робот-курьер.

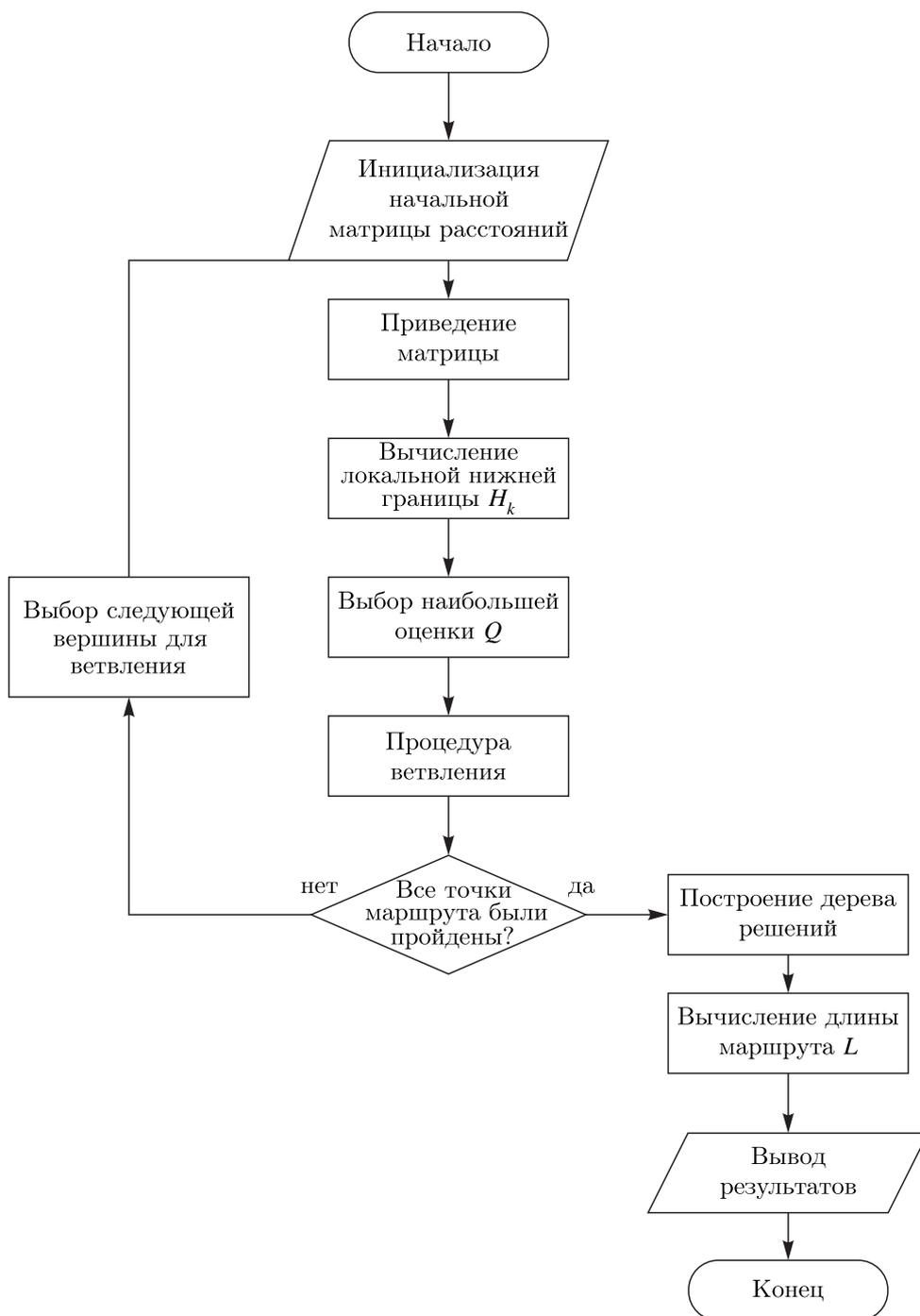


Рис. 2. Блок-схема МВГ

Реализация ГА начинается с создания первой матрицы маршрутов — начальной популяции. Далее генерируются n маршрутов путем перемешивания генов, рассчитывается приспособленность (длина маршрута) каждой хромосомы. Выбираются два родителя (маршрута) и проводится процесс размножения.

При размножении сперва генерируется точка разрыва, с которой начинается процесс скрещивания, после чего формируются первый и второй потомки.

Пусть хромосомы первого родителя будут $X = \{x_i, i = \overline{1, L}\}$, а хромосомы второго — $Y = \{y_i, i = \overline{1, L}\}$. Если точка разрыва будет перед n -м геном, то до скрещивания хромосомы обоих родителей будут выглядеть следующим образом:

$$\begin{aligned} X_1, X_2, X_3, \dots, X_{n-1}, X_n, X_{n+1}, \dots, X_L & \text{— гены I родителя,} \\ Y_1, Y_2, Y_3, \dots, Y_{n-1}, Y_n, Y_{n+1}, \dots, Y_L & \text{— гены II родителя.} \end{aligned}$$

Первый потомок формируется использованием генов I родителя до точки разрыва и генов II родителя после нее:

$$X_1, X_2, X_3, \dots, X_{n-1}, Y_n, Y_{n+1}, \dots, Y_L \text{— гены I потомка.}$$

Второй потомок формируется аналогичным образом:

$$Y_1, Y_2, Y_3, \dots, Y_{n-1}, X_n, X_{n+1}, \dots, X_L \text{— гены II потомка.}$$

Процесс мутации проводится с помощью обмена между случайными генами: выбираются два случайных гена и меняются местами. Операция выполняется для обоих потомков:

$$\begin{aligned} X_1, X_2, Y_n, \dots, X_{n-1}, X_3, Y_{n+1}, \dots, Y_L & \text{— гены I потомка после мутации,} \\ Y_1, X_n, Y_3, \dots, Y_{n-1}, Y_2, X_{n+1}, \dots, X_L & \text{— гены II потомка после мутации.} \end{aligned}$$

Выжившие потомки добавляются в популяцию, отсортированную по возрастанию приспособленности хромосом. Из популяции удаляются наименее приспособленные особи в размере $(K_i - K_{i-1})$, где K_i — размер текущей популяции, K_{i-1} — размер старой популяции.

Процессы размножения, скрещивания, мутации и сортировки популяции повторяются до тех пор, пока не сменяют друг друга заданное число поколений и не будет найдена лучшая особь (искомый путь).

Блок-схема, описывающая ГА, представлена на рис. 3.

Муравьиный алгоритм [Uğur, Aydin, 2009; Branke, Guntzsch, 2004; Ouyang, Yan, 2004; Семенов и др., 2016] (далее — МА) — один из эффективных алгоритмов для нахождения приближенных решений задачи коммивояжера. Также МА подходит для нахождения решения аналогичных задач поиска маршрутов на графах. Основой для данного алгоритма служит коллективная адаптация внутри колонии муравьев (агентов). МА основан на низкоуровневом взаимодействии простых агентов единой системы. Муравей опирается на опыт других муравьев [Семенов и др., 2016] посредством феромона — вещества, оставляемого муравьями на пройденном пути, концентрация которого определяет желание особи выбрать тот или иной путь. Стоит отметить случайный характер движения агента системы при выборе пути [Uğur, Aydin, 2009; Курейчик, Кажаров, 2011] на начальном этапе. Данное явление приводит к попаданию в локальный оптимум.

В качестве основных достоинств МА можно отметить:

- сравнительную эффективность для решения оптимизационных задач и задач размещения;
- возможность использования в динамических приложениях;
- использование памяти обо всей колонии вместо памяти только о предыдущем поколении, ввиду чего данный алгоритм менее подвержен неоптимальным начальным решениям (в сравнении с ГА).

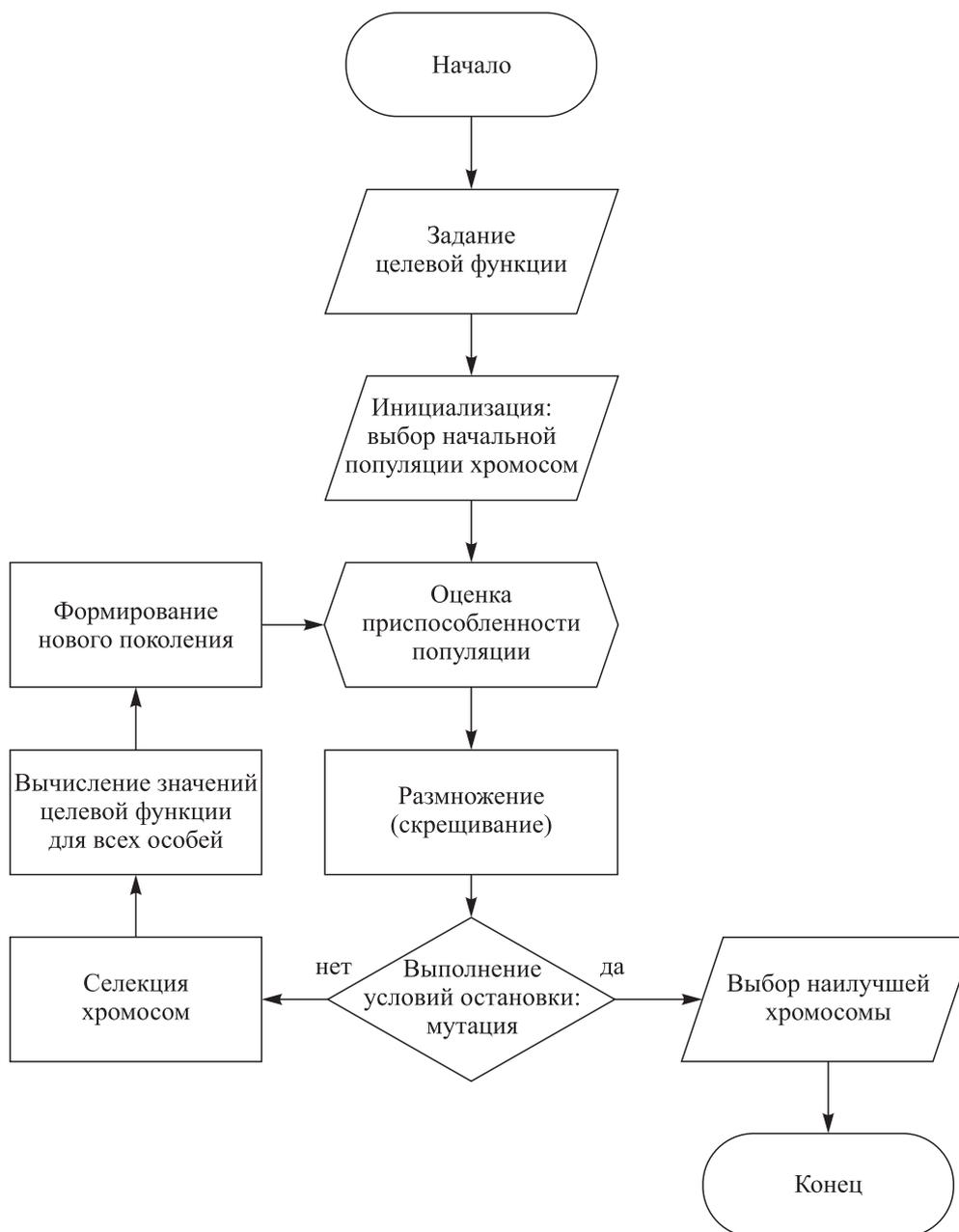


Рис. 3. Блок-схема ГА

Несмотря на перечисленные преимущества, МА имеет следующие недостатки:

- затрудненный теоретический анализ ввиду того, что распределение вероятностей меняется при итерациях, а время сходимости не определено;
- значительная зависимость от настроечных параметров.

Настроечными параметрами для МА являются коэффициент α , определяющий, насколько сильно количество феромона оказывает влияние на выбор муравья, и коэффициент β , определяющий, насколько сильно муравей при принятии решения будет ориентироваться на близость следующей вершины, а также m — число муравьев в колонии.

Вероятность того, что муравей k перейдет из пункта i в пункт j , определяется отношением его желания совершить переход $i \rightarrow j$ $(\tau_{ij,k})^\alpha * (\eta_{ij,k})^\beta$ к сумме его желаний перейти в любую доступную точку, что отражено в формуле (10):

$$P_{ij,k} = \frac{(\tau_{ij,k})^\alpha * (\eta_{ij,k})^\beta}{\sum_{m \in N} (\tau_{ij,k})^\alpha * (\eta_{ij,k})^\beta}, \quad (10)$$

где $P_{ij,k}$ — вероятность перехода из пункта i в пункт j ; k — индекс муравья; $\tau_{ij,k}$ — начальное количество феромона, откладываемого муравьем k при переходе; $\eta_{ij,k}$ — близость пункта i к пункту j ; α и β — настроечные коэффициенты; m — число муравьев в колонии; N — множество доступных пунктов.

Далее случайным образом определяется вероятность. Пункт выбирается исходя из того, в какой диапазон попало значение вероятности.

Пройдя в выбранный пункт, муравей отложит некоторое количество феромона на пути к нему, определяемое по формуле (11). В первом случае выбранная дорога попадает в итоговый

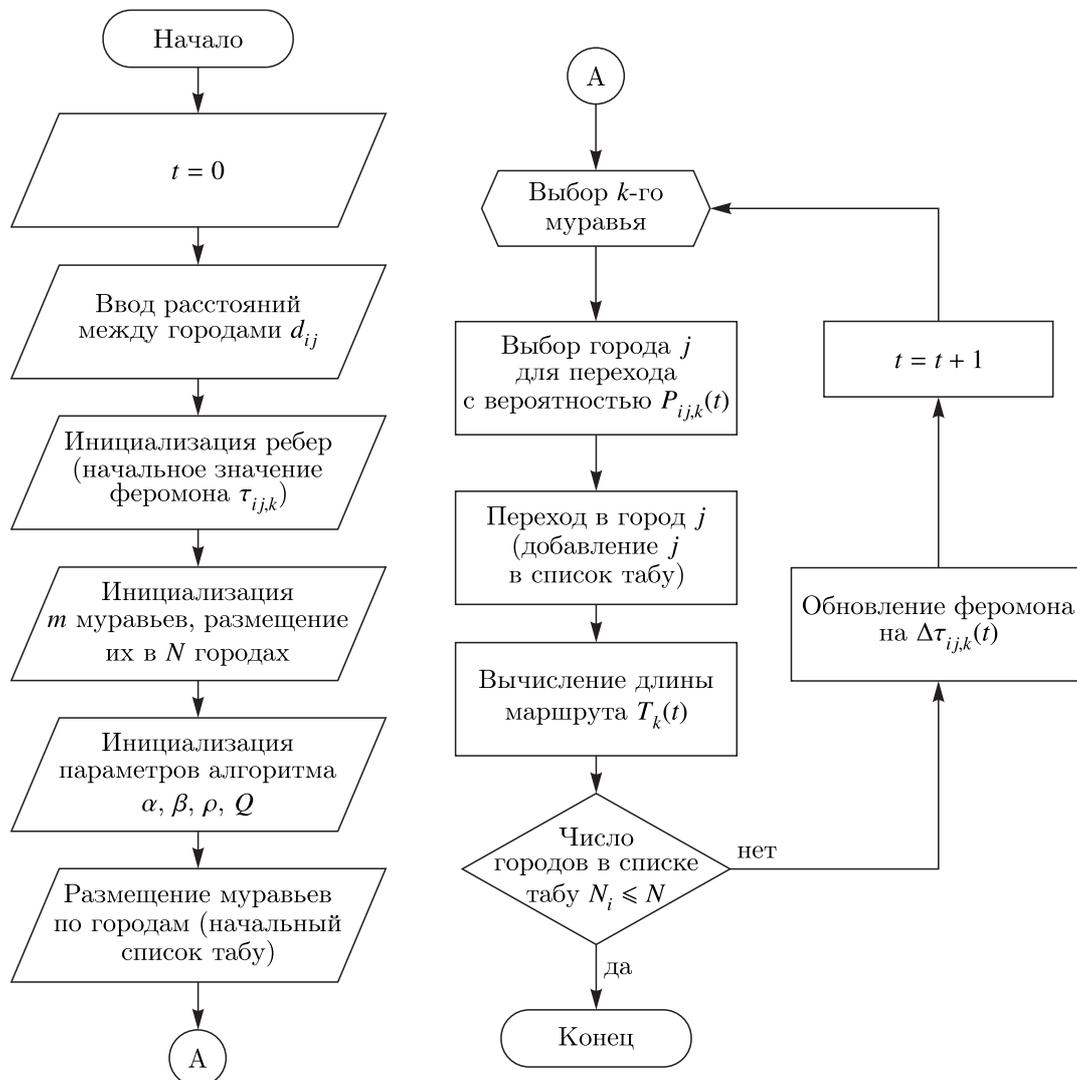


Рис. 4. Блок-схема МА

маршрут, во втором – не попадает [Dorigo, Member, 1997].

$$\tau_{ij,k}(t) = \begin{cases} \frac{Q}{L_k(t)}, & \text{если } (i, j) \in T_k(t); \\ 0 & \text{– в ином случае.} \end{cases} \quad (11)$$

Стоит учитывать, что феромон на протяжении всего времени испаряется по закону, определяемому формулой (12) [Uğur, Aydin, 2009; Dorigo, Member, 1997; Guntsch et al., 2001]:

$$\Delta\tau_{ij}(t+1) = (1-p) * \tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij,k}(t). \quad (12)$$

Блок-схема, описывающая работу МА, представлена на рис. 4.

6. Эксперимент и анализ результатов

Для сравнения выбранных алгоритмов было проведено несколько экспериментов, варьирующих сложность и тип задачи коммивояжера. На основе матрицы расстояний были составлены наборы по 3–26 точек заказа.

Все эксперименты были выполнены на персональном компьютере с 64-разрядной ОС Windows 10 с процессором Intel Core i3-7100U с тактовой частотой 2,40 ГГц и 4 ГБ оперативной памяти. Алгоритмы были реализованы на языке Python.

Во время расчета каждым из алгоритмов фиксируются время работы алгоритма T , протяженность полученного маршрута L . Также замеряются значения пикового и текущего объема памяти, необходимого для решения задачи при малой (10×10) и большой (26×26) выборке точек, и рассчитываются средние отклонения от эталонного значения длины пути $L_{\text{откл}}$ (эталонном считаем длину пути, рассчитанную МВГ).

Полученные результаты экспериментов представлены в таблице 1. Эксперименты с МВГ проводились до 20-й точки включительно, так как время на выполнение программы экспоненциально росло с увеличением количества точек N . Общее количество вычислительных экспериментов для получения средних значений показателей:

- 20 экспериментов для МВГ;
- 26 экспериментов для ГА и МА.

Из таблицы 1 заметна тенденция: при увеличении количества точек время, необходимое для поиска маршрута, между алгоритмами распределяется неравномерно. На малых выборках (до 10 точек включительно) быстрее всего работает МВГ, он же дает оптимальное значение. Медленнее всего работает ГА, однако рассчитанные им длины маршрутов практически совпадают с эталонными значениями, средние отклонения минимальны. МА выбирает самые длинные маршруты, но работает быстрее, чем ГА.

На больших выборках (от 11 точек) МА находит оптимальные маршруты быстрее, чем ГА, а длина маршрутов становится приближенной к оптимальной (до тех пор, пока возможно определение времени работы программы для МВГ, т. е. до 20 точек).

Временная сложность работы алгоритмов представлена на рис. 5. Явно заметен экспоненциальный рост времени, необходимого МВГ для поиска оптимального маршрута.

По рис. 6 можно проследить работу алгоритмов в отношении длины маршрута. Видна конкуренция между МА и ГА в промежутке от 10 до 16 точек, однако с дальнейшим ростом N маршруты, которые находит МА, заметно короче тех, которые находит ГА. Отметим, что до 16 точек

Таблица 1. Результаты экспериментов

N	МВГ		ГА			МА		
	L (м)	T (с)	L (м)	T (с)	L _{откл.} (м)	L (м)	T (с)	L _{откл.} (м)
3	570	0,02607	570	0,25271	0	727	0,04774	78,5
4	720	0,02593	720	0,32320	0	904	0,08210	92
5	830	0,02248	840	0,59409	5	1091	0,13346	130,5
6	940	0,02892	960	0,86195	10	1181	0,16414	120,5
7	1060	0,04510	1060	0,90781	0	1377	0,25260	158,5
8	1170	0,04487	1170	1,21250	0	1579	0,30717	204,5
9	1370	0,08776	1390	1,24365	10	1686	0,38746	158
10	1440	0,14204	1470	1,40111	15	1752	0,48546	156
11	1520	1,35972	1850	1,75593	165	1850	0,70067	165
12	1660	4,89539	1700	1,97140	20	1770	0,90237	55
13	1700	8,48968	2240	2,28792	270	2380	1,71265	340
14	1840	11,44264	2010	2,68779	85	2040	1,59288	100
15	1910	16,09650	2260	2,82111	175	2350	1,33069	220
16	1940	24,06163	2180	3,19507	120	2240	2,18089	150
17	2180	44,66739	3740	3,41825	780	2580	1,95853	200
18	2270	56,87480	4170	3,46712	950	2710	2,59369	220
19	2340	72,76961	4410	3,47459	1035	2610	2,04207	135
20	2360	224,96907	4760	3,76969	1200	2780	2,72900	210
21	—	∞	3590	4,13085	—	3190	2,84529	—
22	—	∞	4970	4,39425	—	2950	3,39234	—
23	—	∞	5660	4,51261	—	3490	3,03589	—
24	—	∞	5440	4,72925	—	3850	3,80022	—
25	—	∞	5810	5,09400	—	3590	4,97836	—
26	—	∞	6700	5,54418	—	3500	5,50137	—
Среднее значение T (с) и L (м) для малой выборки точек (≤ 10)								
	1012,5	0,05289	1022,5	0,84962		1287,125	0,23252	
Среднее значение T (с) или L (м) для большой выборки точек (10 < N ≤ 26)								
	1545,555	46,562643	2902,916	2,66879		2257,375	1,79821	

Таблица 2. Средние значения пикового и текущего объема памяти

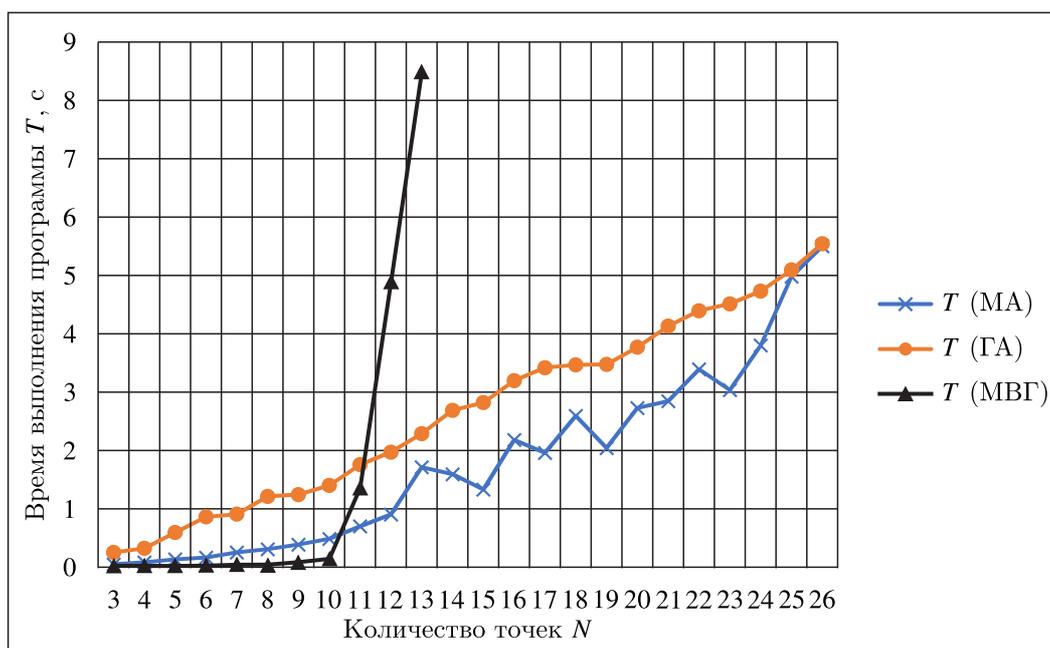
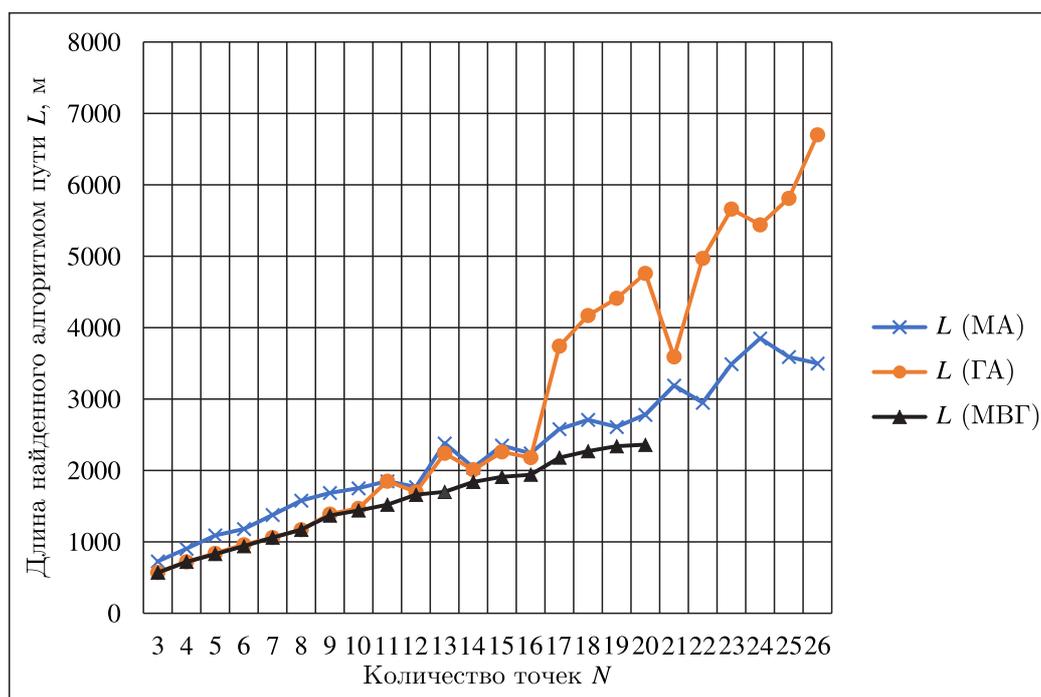
МВГ		ГА		МА	
Пиковый объем	Текущий объем	Пиковый объем	Текущий объем	Пиковый объем	Текущий объем
Малая выборка точек (10)					
334,302 КВ	149,660 КВ	334,749 КВ	153,752 КВ	333,747 КВ	191,578 КВ
Большая выборка точек (26)					
335,580 КВ	164,994 КВ	337,408 КВ	167,430 КВ	422,009 КВ	231,164 КВ

оба эвристических алгоритма выдавали решение, близкое к эталонному, а МА продолжил находить приближенные к оптимальному маршруты и до 20 точек.

Конкуренцию между ГА и МА по близости найденного маршрута к оптимальному можно проследить на рис. 7.

Средние значения пикового и текущего объема памяти, необходимого рассматриваемым алгоритмам для решения задачи с разными наборами точек, представлены в таблице 2. Текущий объем памяти определяется в конце работы программы, на этапе, когда она выдает результат.

Из таблицы 2 видно, что наиболее ресурсоемким в значении текущего объема памяти является МА: и для малых, и для больших наборов точек значения максимальны среди всех

Рис. 5. Временная сложность алгоритмов в зависимости от количества точек N Рис. 6. Зависимость длины найденного маршрута от количества точек N

рассматриваемых алгоритмов. Пиковый объем памяти для всех алгоритмов для малого набора точек практически равен. Объем памяти, необходимой МВГ и ГА и в моменты максимальной нагрузки, и в конце работы программы приблизительно одинаков.

Стоит отметить, что МВГ в сравнении для большой выборки точек (≥ 20) не участвовал, так как для него не проводились эксперименты с количеством точек больше 20 ввиду больших временных затрат на выполнение программы.

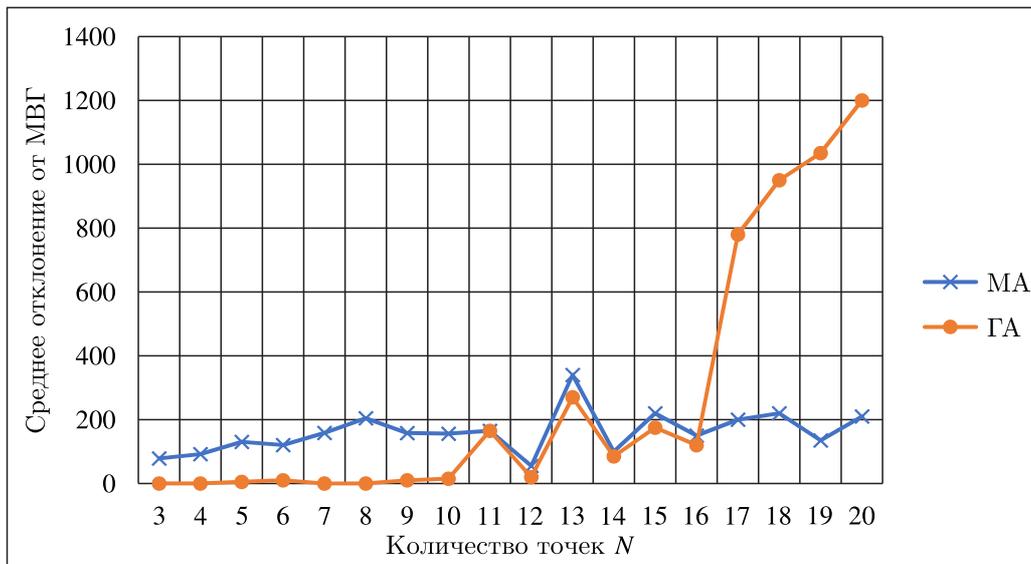


Рис. 7. Зависимость среднего отклонения длины пути, найденного эвристическими алгоритмами, от количества точек N

Исходя из полученных результатов, можно сделать следующие выводы.

1. Для задач малого размера (до 10 точек), требующих нахождения быстрого и оптимального решения, целесообразно применять МВГ. Он требует средних затрат памяти, прост в реализации и разработке.
2. Для задач большого размера (более 10 точек) наилучшим решением, в зависимости от ограниченного ресурса (длины маршрута, времени или объема памяти ЭВМ), оптимальным будет МА (при ограниченности времени и длины маршрута) или ГА (при ограниченности памяти ЭВМ).

Рассматриваемая нами задача нахождения близкого к оптимальному пути роботом-тележкой сочетает в себе ограничения сразу трех ресурсов: и времени, и длины маршрута, и объема памяти, отведенной для вычислений. Количество точек, которые необходимо будет объехать роботу-курьеру, постоянно меняется от 1 до 26. В данном случае рациональным решением будет использование ГА, имеющего средние (относительно других рассматриваемых алгоритмов) значения для решения задач любой сложности.

7. Выводы

Из проведенных расчетов и сравнительного анализа полученных результатов следует, что выбор алгоритма для построения оптимального маршрута зависит от количества заданных точек. Можно сделать следующие выводы.

1. Для малого количества заданных точек (в интервале 3–10) наиболее целесообразным способом поиска оптимального маршрута является точный алгоритм, реализуемый в данной работе через МВГ. Временные показатели работы программы, длины полученных маршрутов и объем используемой в процессе работы программы памяти являются наилучшими из представленных алгоритмов, что отражено в таблице 2. Стоит отметить, что длины маршрутов, полученные с помощью ГА, незначительно отклоняются от результатов МВГ (среднее отклонение $\Delta L_{\text{ср}} = 3,5$ м, или 0,86 %), но временные показатели работы программы многократно больше (в среднем время работы ГА в 18,7 раз больше, чем время работы МВГ). МА также

значительно уступает как в скорости работы (в среднем время работы МА в 4,6 раз больше, чем время работы МВГ), так и в длине построенных маршрутов (среднее отклонение $\Delta L_{\text{ср}} = 137,3$ м, или 27,47 %).

2. Для большого количества точек (в интервале 11–26), на основе проведенного эксперимента, наиболее целесообразным методом является МА. Для МВГ в данном интервале увеличение количества заданных точек сопряжено с увеличением времени работы программы, которое имеет экспоненциальный характер. Значительно отклоняются показатели длины полученного маршрута ГА по сравнению с МА (среднее отклонение $\Delta L_{\text{ср}} = 1100,625$ м, или 34,94 %), также время работы ГА дольше (в среднем время работы ГА в 1,56 раза больше, чем время работы МА).

Таким образом, выбор алгоритма для поиска оптимального маршрута необходимо осуществлять исходя из количества заданных точек. Для малого количества точек ($N \leq 10$) МВГ является наиболее предпочтительным. С увеличением числа точек более эффективные результаты дает эвристический подход, представленный в данной работе МА и ГА.

8. Заключение

На сегодняшний день в логистике наблюдается тенденция к использованию беспилотных технологий как одного из самых инновационных и перспективных направлений. Внимание к данному направлению обусловлено потенциальным снижением затрат и повышением прибыльности деятельности организаций. Часто такие технологии используются в сфере складской и транспортной логистики, когда беспилотному аппарату необходимо найти приближенный к оптимальному маршрут. Такую задачу можно свести к задаче коммивояжера, которая относится к NP-трудным задачам дискретной оптимизации, поэтому время ее решения с помощью точных методов растет экспоненциально с увеличением количества вершин графа. Для решения задачи коммивояжера на текущий момент существует множество методов.

В представленной статье предлагается решение задачи коммивояжера для вычисления оптимального маршрута робота тележки тремя различными алгоритмами. По результатам проведенного исследования каждый из выбранных для рассмотрения алгоритмов имеет свои достоинства и недостатки, поэтому могут быть применимы при различных условиях и ограничениях. Метод ветвей и границ, который близок к простому перебору, имеет ограничения по количеству точек маршрута, а именно не более 10, из-за экспоненциально возрастающего времени вычисления. Однако маршрут, вычисляемый в пределах 10 точек, наиболее близок к эталонному при сравнительно небольших затратах вычислительных ресурсов. Эвристические методы, представленные в статье муравьиным и генетическим алгоритмом, показывают хорошие результаты при большем наборе точек. При этом ГА менее ресурсоемкий, а МА более точен. Исходя из полученных результатов, комбинация этих методов при решении практических задач будет наиболее предпочтительна. На сегодняшний момент авторами ведется апробация других видов роевых алгоритмов (пчелиный алгоритм, алгоритм капель воды) и их комбинаций для решения задачи о рюкзаке, что является еще более актуальным для беспилотных роботов-тележек с ограниченной емкостью, осуществляющих доставку товаров.

Список литературы (References)

- Костюк Ю. Л.* Задача коммивояжера: улучшенная нижняя граница в методе ветвей и границ // Прикладная дискретная математика. — 2013. — № 4 (22). — С. 73–82.
Kostyuk Yu. L. Zadacha kommivoyazhera: uluchshennaya nizhnyaya granitsa v metode vetvei i granits [The traveling salesman's problem: an improved lower bound in the method of branches and boundaries] // Prikladnaya diskretnaya matematika [Applied Discrete Mathematics]. — 2013. — No. 4 (22). — P. 73–82 (in Russian).

- Костюк Ю. Л.* Эффективная реализация алгоритма решения задачи коммивояжера методом ветвей и границ // Прикладная дискретная математика. — 2013. — № 2 (20). — С. 78–80.
Kostyuk Yu. L. Effektivnaya realizatsiya algoritma resheniya zadachi kommivoyazhera metodom vetvei i granits [Effective implementation of the algorithm for solving the traveling salesman problem by the method of branches and boundaries] // Prikladnaya diskretnaya matematika [Applied Discrete Mathematics]. — 2013. — No. 2 (20). — P. 78–80 (in Russian).
- Кремер О. Б., Подвальный С. Л.* Программная реализация решения оптимизационных задач методом генетического алгоритма // Вестник Воронежского государственного технического университета. — 2012. — С. 21–24.
Kremer O. B., Podval'nyi S. L. Programmnaya realizatsiya resheniya optimizatsionnykh zadach metodom geneticheskogo algoritma [Software implementation of optimization problem solving by genetic algorithm method] // Vestnik Voronezhskogo gosudarstvennogo tekhnicheskogo universiteta [Bulletin of the Voronezh State Technical University]. — 2012. — P. 21–24 (in Russian).
- Курейчик В. М., Кажаров А. А.* Использование роевого интеллекта в решении NP-трудных задач // Известия Южного федерального университета. Технические науки. — 2011. — С. 30–36.
Kureichik V. M., Kazharov A. A. Ispol'zovanie roevogo intellekta v reshenii NP-trudnykh zadach [The use of swarm intelligence in solving NP-hard problems] // Izvestiya Yuzhnogo federal'nogo universiteta. Tekhnicheskie nauki [Izvestia of the Southern Federal University. Technical sciences]. — 2011. — P. 21–24 (in Russian).
- Подшивалов С. Ф., Подшивалова К. С., Левицкая Л. В., Дунаев С. В.* Совершенствование алгоритма метода ветвей и границ в задаче коммивояжера для автомобильного транспортного графа // Модели, системы, сети в экономике, технике, природе и обществе. — 2017. — № 1. — С. 169–179.
Podshivalov S. F., Podshivalova K. S., Levitskaya L. V., Dunaev S. V. Sovershenstvovanie algoritma metoda vetvei i granits v zadache kommivoyazhera dlya avtomobil'nogo transportnogo grafa [Improving the algorithm of the method of branches and boundaries in the traveling salesman problem for the automobile transport graph] // Modeli, sistemy, seti v ekonomike, tekhnike, prirode i obshchestve [Models, systems, networks in economics, technology, nature and society]. — 2017. — No. 1. — P. 169–179 (in Russian).
- Робот-курьер Яндекс.Ровер начал работать в Сколково: [Электронный ресурс] // Яндекс, 2020. — URL: https://yandex.ru/company/services_news/2020/2020-04-29-1 (дата обращения: 29.10.2021).
Robot-kur'er Yandeks.Rover nachal rabotat' v Skolkovo [Yandex courier robot.Rover started working in Skolkovo] [Electronic resource] // Yandex, 2020. — Available at: https://yandex.ru/company/services_news/2020/2020-04-29-1 (accessed: 29.10.2021).
- Роботы-курьеры помогут создать безбарьерную среду. [Электронный ресурс] // +1Платформа, 2021. — URL: <https://platform.plus-one.ru/news/2021/09/01/2359-roboty-kuriery-pomogut-sozdat-bezbariernuyu-sredu> (дата обращения: 29.10.2021).
Roboty-kur'ery pomogut sozdat' bezbar'ernuyu sredu [Courier robots will help to create a barrier-free environment] [Electronic resource] // +1Platforma, 2021. — Available at: <https://platform.plus-one.ru/news/2021/09/01/2359-roboty-kuriery-pomogut-sozdat-bezbariernuyu-sredu> (accessed: 29.10.2021).
- Семенов С. С.* Программа для решения задачи коммивояжера с помощью муравьиного алгоритма / С. С. Семенов, Д. Ф. Ткачев, А. В. Педан, Е. А. Алисевиц, А. В. Попов, О. С. Воронцов, Д. В. Киселев, И. С. Климов // Хроники объединенного фонда электронных ресурсов «Наука и образование». — 2016. — № 6. — С. 31.
Semenov S. S. Programma dlya resheniya zadachi kommivoyazhera s pomoshch'yu murav'inogo algoritma [Program for solving the traveling salesman's problem with the help of an ant algorithm] / S. S. Semenov, D. F. Tkachev, A. V. Pedan, E. A. Alisevich, A. V. Popov, O. S. Vorontsov, D. V. Kiselev, I. S. Klimov // Khroniki obedinennogo fonda elektronnykh resursov «Nauka i obrazovanie» [Chronicles of the United Fund of Electronic Resources Science and Education]. — 2016. — No. 6. — P. 21–24 (in Russian).
- Снабжение: роботы и дроны на страже логистических цепочек. [Электронный ресурс] // РБК, 2020. — URL: <http://digital-russia.rbc.ru/articles/snabzhenie-roboty-i-drony-na-strazhe-logisticheskikh-tsepochek/> (дата обращения: 29.10.2021).
Snabzhenie: roboty i drony na strazhe logisticheskikh tsepochek [Supply: robots and drones on guard of logistics chains] [Electronic resource] // RBK, 2020. — Available at: <http://digital-russia.rbc.ru/articles/snabzhenie-roboty-i-drony-na-strazhe-logisticheskikh-tsepochek/> (accessed 29.10.2021).
- Чмырь Д. А.* Оптимизация технологических процессов генетическими алгоритмами. — М.: Изд-во AP LAMBERT Academic Publishing, 2018. — 52 с.
Chmyr' D. A. Optimizatsiya tekhnologicheskikh protsessov geneticheskimi algoritmami [Optimization of technological processes by genetic algorithms]. — Moscow: AP LAMBERT Academic Publishing, 2018. — P. 21–24 (in Russian).

- Agatz N., Bouman P., Schmidt M.* Optimization Approaches for the Traveling Salesman Problem with Drone // *Transportation Science*. — 2018. — Vol. 52, No. 4. — P. 965–981.
- Branke J., Guntsch M.* Solving the probabilistic TSP with ant colony optimization // *Journal of Mathematical Modelling and Algorithms*. — 2004. — P. 403–425.
- Dorigo M., Member S.* Ant colony system: a cooperative learning approach to the traveling salesman problem // *IEEE transactions on evolutionary computation*. — 1997. — Vol. 1, No. 1. — P. 53–66.
- Greco F.* Traveling Salesman Problem. — InTech, 2008. — 212 p.
- Guntsch M., Middendorf M., Schmeck H.* An ant colony optimization approach to dynamic TSP // *Proceedings of the 3rd annual conference on Research in information technology*. — 2001. — P. 860–867.
- Gutin G., Punnen A.* The Traveling Salesman Problem and Its Variations. — London: Springer-Verlag, 2006. — 40 p.
- Last Link: Quantifying the Cost. [Electronic resource] // Cushman & Wakefield, 2019. — Available at: https://www.cushmanwakefield.com.ua/sites/default/files/2020-02/CUS42002904%20Last%20Link%20Report_v13.pdf (accessed: 29.10.2021).
- Manyika J.* Jobs lost, jobs gained: What the future of work will mean for jobs, skills, and wages / J. Manyika, S. Lund, M. Chui, J. Bughin, J. Woetzel, P. Batra, R. Ko, S. Sanghvi. [Electronic resource] // McKinsey & Company, 2017. — Available at: <https://www.mckinsey.com/featured-insights/future-of-work/jobs-lost-jobs-gained-what-the-future-of-work-will-mean-for-jobs-skills-and-wages> (accessed: 29.10.2021).
- Nilsson C.* Heuristics for the traveling salesman problem. — Linköping, Sweden: Linköping University, 2013.
- Ouyang J., Yan G. R.* Multi-group ant colony system algorithm for TSP // *Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai*. — 2004. — P. 26–29.
- Razali N., Geraghty J.* Genetic algorithm performance with different selection strategies in solving TSP // *Proceedings of the World Congress on Engineering*. — London, UK. — 2011. — Vol. II. — 6 p.
- Robots Threaten Bigger Slice of Jobs in US, Other Rich Nations. [Electronic resource] // *Wired*, 2017. — Available at: <https://www.wired.com/story/robots-threaten-bigger-slice-of-jobs-in-us-other-rich-nations/> (accessed: 29.10.2021).
- Uğur A., Aydın D.* An interactive simulation and analysis software for solving TSP using Ant Colony Optimization algorithms // *Advances in Engineering software*. — 2009. — P. 341–349.