

УДК 004.023

DOI: [10.26102/2310-6018/2025.51.4.060](https://doi.org/10.26102/2310-6018/2025.51.4.060)

Исследование генетического алгоритма решения задачи множественной маршрутизации транспортных средств с чередованием объектов двух типов

О.А. Медведева, С.Н. Медведев, А.Г. Трифонов 

Воронежский государственный университет, Воронеж, Российской Федерации

Резюме. В статье рассматривается задача маршрутизации нескольких транспортных средств, особенностью которой является требование чередования объектов двух видов в результирующих маршрутах. При этом объекты загрузки должны быть обязательно посещены ровно один раз, а центры разгрузки могут входить в маршруты транспортных средств неограниченное число раз. Необходимо минимизировать максимальную из длин маршрутов транспортных средств, то есть, другими словами, сократить разброс длин полученных маршрутов. Для решения поставленной задачи предлагается использовать генетический алгоритм. С этой целью способ построения хромосом, операторы отбора особей, скрещивания и мутации модернизируются с учетом специфики задачи, причем каждая хромосома включает маршруты всех транспортных средств. Основная часть статьи посвящена исследованию и настройке параметров предложенного алгоритма решения. Для разных размерностей решаемых задач оцениваются длины полученных маршрутов транспортных средств в зависимости от размера популяции, числа итераций алгоритма, а также значения коэффициента мутации. На основе проведенного исследования разработаны рекомендации по применению предложенного алгоритма в рамках поставленной задачи. Также проводится оценка работоспособности и результативности алгоритма в сравнении с точным методом решения. Показано, что предложенный алгоритм позволяет получать решения, близкие к оптимальному, за существенно меньшее время.

Ключевые слова: задача маршрутизации транспортных средств, генетический алгоритм, оптимизационная задача, задача с чередованием, эвристический алгоритм.

Для цитирования: Медведева О.А., Медведев С.Н., Трифонов А.Г. Исследование генетического алгоритма решения задачи множественной маршрутизации транспортных средств с чередованием объектов двух типов. *Моделирование, оптимизация и информационные технологии*. 2025;13(4). URL: <https://moitvivt.ru/tu/journal/pdf?id=2143> DOI: 10.26102/2310-6018/2025.51.4.060

The study of a genetic algorithm for solving the problem of multiple vehicle routing with alternating objects of two types

О.А. Medvedeva, S.N. Medvedev, A.G. Trifonov 

Voronezh State University, Voronezh, the Russian Federation

Abstract. This article considers a multi-vehicle routing problem requiring alternating two types of objects in the resulting routes. Loading facilities must be visited exactly once, while unloading centers can be included in vehicle routes an unlimited number of times. It is necessary to minimize the maximum length of vehicle routes, or, in other words, to reduce the variance in the lengths of the resulting routes. A genetic algorithm is proposed for solving this problem. To this end, the chromosome construction method, individual selection operators, crossovers, and mutations are optimized to address the specifics of the problem, with each chromosome including the routes of all vehicles. The main part of the article is devoted to the study and tuning of the parameters of the proposed solution algorithm. For different problem dimensions, the lengths of the resulting vehicle routes are estimated depending on the population size, the number of algorithm iterations, and the mutation coefficient. Based on the conducted

research, recommendations for applying the proposed algorithm to the given problem are developed. The algorithm's performance and effectiveness are also evaluated in comparison with an exact solution method. It is shown that the proposed algorithm allows obtaining solutions close to optimal in significantly less time.

Keywords: vehicle routing problem, genetic algorithm, optimization problem, interleaving problem, heuristic algorithm.

For citation: Medvedeva O.A., Medvedev S.N., Trifonov A.G. The study of a genetic algorithm for solving the problem of multiple vehicle routing with alternating objects of two types. *Modeling, Optimization and Information Technology*. 2025;13(4). (In Russ.). URL: <https://moitvivt.ru/ru/journal/pdf?id=2143> DOI: 10.26102/2310-6018/2025.51.4.060

Введение

Интерес к задачам маршрутизации обусловлен множеством модификаций, связанных с внесением дополнительных требований к формируемому маршруту и различными вариациями целевой функции. Из наиболее известных дополнительных требований можно выделить условия, отвечающие за загруженность машин. Так, в [1] рассматривается задача обслуживания потребителей с учетом полной загруженности машин, в [2] рассматривается доставка товара дронами, которым необходимо каждый раз возвращаться обратно на базу из-за ограничения по весу груза, в [3] рассматривается доставка товара суднами разной грузоподъемности.

Также есть отдельные классы задач маршрутизации, такие как проблема маршрутизации транспортных средств с ограничениями по мощности (CVRP) и проблема маршрутизации транспортных средств с временными окнами (VRPTW). В [4] описана постановка и математическая модель задачи с ограничениями по мощности транспортных средств, в [5] данная задача решается с помощью поиска с запретами, а в [6] использован двухфазный алгоритм поиска по окрестностям, который последовательно решает сначала задачу маршрутизации, а потом кластеризации объектов. В [7] описана задача маршрутизации транспортных средств с временными окнами, для решения которой используется алгоритм муравьиных колоний. В [8] также с помощью алгоритма муравьиных колоний решается задача маршрутизации транспортных средств с ограничением по времени

В большинстве своем многочисленные модификации задачи маршрутизации транспортных средств являются NP-трудными [9]. Это означает, что при увеличении размерности задачи временная сложность точных методов решения растет экспоненциально. В связи с этим актуальной становится проблема разработки эвристических алгоритмов решения. Самыми известными из них являются генетические алгоритмы [1, 10] и алгоритмы муравьиных колоний [11].

Эвристические алгоритмы, как и точные, также подвержены модификациям. Встречаются подходы, основанные на предварительной кластеризации объектов [9], где каждый объект принадлежит одному или нескольким кластерам, и необходимо построить такой маршрут, чтобы каждый кластер был посещен только один раз. В [12] решена задача маршрутизации с ограничением пропускной способности транспортного средства с помощью алгоритма кластеризации. В [13] использован алгоритм усеченных ветвей. Дерево поиска в этой процедуре содержит столько же уровней, сколько маршрутов транспортных средств, и каждый уровень содержит набор допустимых и недоминируемых маршрутов транспортных средств.

В [11] авторами рассмотрена задача маршрутизации транспортных средств с требованием чередования объектов двух типов в результирующих маршрутах. Для решения задач такого типа в [14] предлагается точный алгоритм, основанный на методе

ветвей и границ. В [15] используются жадные алгоритмы, на основе которых строится их вероятностная модификация.

В данной статье будет рассмотрена задача множественной маршрутизации транспортных средств с чередованием неподвижных объектов двух типов, для решения которой будет использоваться генетический алгоритм.

Материалы и методы

В статье рассматривается следующая постановка задачи множественной маршрутизации транспортных средств:

Рассматривается совокупность вершин трех типов: вершины первого типа (объекты загрузки), которые посещаются только один раз и имеют одинаковый вес в 1 условную единицу; вершины второго типа, которые являются центрами разгрузки и посещаются неограниченное число раз; а также вершина депо, в которой начинают и заканчивают свои маршруты все транспортные средства (агенты). Грузоподъемность агентов равна единице, что подразумевает необходимость чередования в их маршрутах вершин загрузки и центров разгрузки. Агенты должны в совокупности обеспечить доставку грузов из всех объектов загрузки в центры разгрузки. При этом требуется свести к минимуму разброс времен выполнения маршрутов для всех агентов.

Содержательная постановка задачи может быть представлена как сбор стогов сена с поля в грузовые машины сельскохозяйственной техникой [15].

На Рисунке 1 представлен пример допустимых маршрутов для двух агентов. Объекты загрузки представлены кругами, центры разгрузки – квадратами.

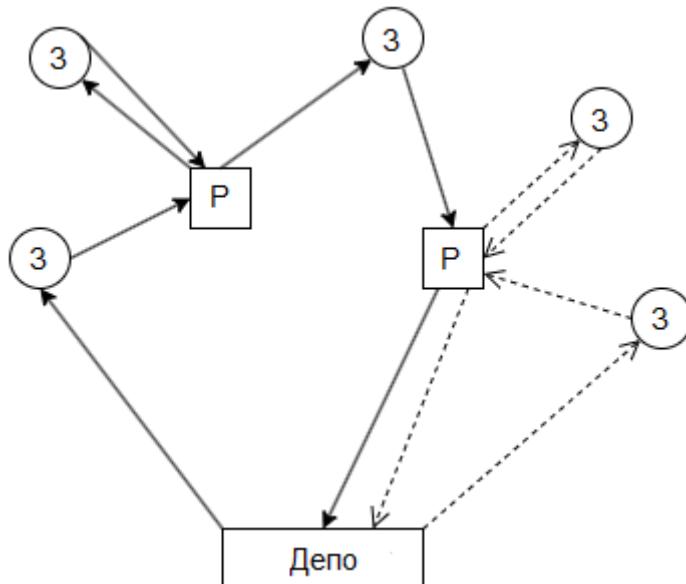


Рисунок 1 – Пример допустимых маршрутов агентов

Figure 1 – Example of acceptable agent routes

Математическая модель задачи подробно рассмотрена авторами в [14] и имеет следующий вид:

$$L(X) = \max_{t=1,p} \left\{ \sum_{i=0}^n \sum_{j=1}^{m+1} \sum_{k=0}^n c_{ijk} x_{ijk}^t \right\} \rightarrow \min, \quad (1)$$

$$\sum_{t=1}^p \sum_{i=0}^n \sum_{k=1}^n x_{ijk}^t = 1, \quad j = \overline{1, m}, \quad (2)$$

$$\sum_{t=1}^p \sum_{i=0}^n \sum_{j=1}^{m+1} x_{ijk}^t = \sum_{t=1}^p \sum_{i=0}^n \sum_{j=1}^{m+1} x_{kji}^t, \quad k = \overline{0, n}, \quad (3)$$

$$\sum_{t=1}^p \sum_{j=1}^{m+1} \sum_{k=0}^n x_{0jk}^t = p, \quad (4)$$

$$\sum_{t=1}^p \sum_{i=0}^n \sum_{j=1}^{m+1} x_{ij0}^t = p, \quad (5)$$

$$\sum_{t=1}^p \sum_{i=0}^n \sum_{k=1}^n x_{i(m+1)k}^t = 0, \quad (6)$$

$$x_{ijk}^t \in \{0,1\}, i = \overline{1, n}, j = \overline{1, m}, k = \overline{1, n}, t = \overline{1, p}. \quad (7)$$

Здесь m – количество объектов загрузки, n – количество центров разгрузки, p – число агентов. Для поддержания требования чередования объектов в маршрутах депо представляется парой фиктивных объектов с индексами $i = 0$ и $j = m + 1$ соответственно. Матрица $\{c_{ijk}\}_{(n+1) \times (m+1) \times (n+1)}$ отвечает за временные затраты на перемещение агента между i -м центром разгрузки, j -м объектом загрузки и k -м центром разгрузки. Кроме того, вводятся переменные $x_{ijk}^t \in \{0,1\}$, $i = \overline{1, n}$, $j = \overline{1, m}$, $k = \overline{1, n}$, $t = \overline{1, p}$, для которых $x_{ijk}^t = 1$, если в маршрут t -го агента входит цепочка, состоящая из i -го центра разгрузки, j -го объекта загрузки и k -го центра разгрузки.

Отдельно следует сказать о виде целевой функции. Она подразумевает минимизацию по всем агентам максимального времени выполнения построенного маршрута, то есть, другими словами, реализацию минимаксного критерия по всем маршрутам транспортных средств.

Для решения описанной оптимизационной задачи предлагается использовать эволюционный алгоритм решения – генетический алгоритм. Остановимся подробнее на основных этапах построения алгоритма применительно к рассмотренной задаче.

В классической задаче маршрутизации хромосома представляет собой маршрут транспортного средства. Генами выступают вершины, через которые проезжают транспортные средства. Популяция представляет собой набор таких маршрутов [12]. Однако в связи с тем, что в рассматриваемой в статье задаче подразумевается наличие нескольких агентов и чередование в их маршрутах объектов двух типов, хромосома принимает несколько иной вид. Гены, стоящие на четных местах, соответствуют объектам загрузки. Они должны быть уникальными в рамках всей хромосомы. Гены на нечетных местах отвечают за центры разгрузки и могут повторяться неограниченное число раз. Для разделения маршрутов разных транспортных средств используются гены, соответствующие депо. Причем для сохранения четно-нечетного порядка следования генов депо реализуется двумя генами загрузки и разгрузки соответственно. Иллюстративный пример допустимого вида хромосомы представлен на Рисунке 2, где для объектов загрузки принято обозначение 3, для разгрузки – Р, пара генов депо обозначена как Дз и Др, индекс i отвечает за номер агента.

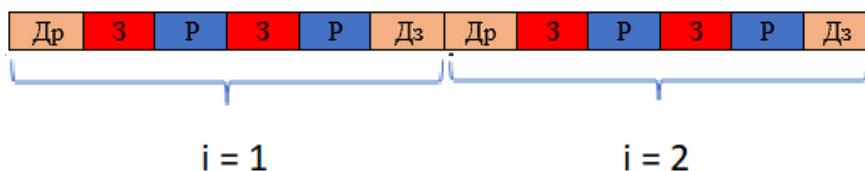


Рисунок 2 – Иллюстративный пример формирования хромосомы
 Figure 2 – An illustrative example of chromosome formation

При формировании начальной популяции предполагается, что в маршрут каждого транспортного средства должно попасть схожее число посещаемых объектов. В связи с

этим в маршрут каждого агента помещается $\lceil \frac{m}{p} \rceil$ случайно выбранных пар, состоящих из центров разгрузки и неповторяющихся объектов загрузки.

Псевдокод жадного алгоритма формирования хромосомы представлен ниже.

Генерация_популяции (p, C, размер_популяции)

Для каждой особи:

Для каждого ТС:

Хромосома.Добавить(Депо)

$i = 0$

Пока $i < m/t$:

Хромосома.Добавить(random(min1, min2)) #где min1, min2 $\in M$, $\neg \in$ Хромосома

#Добавление одного из двух ближайших целевых объектов

Хромосома.Добавить(random(min1, min2)) #где min1, min2 $\in N$

#Добавление одного из двух ближайших центров

$i = i + 1$

Хромосома.Добавить(Депо)

Популяция.Добавить(Хромосома)

Процедура селекции особей в предлагаемом алгоритме реализуется методом ранжирования. В качестве значения функции приспособленности предлагается взять величину, обратно пропорциональную наибольшему времени выполнения из полученных маршрутов агентов

$$f_s = \frac{1}{L_s}, \quad (8)$$

где L_s – значение целевой функции (1) для s -ой особи в популяции.

Для поддержания многообразия особей в популяции в генетическом алгоритме предусмотрены две процедуры: скрещивания и мутации.

В рассматриваемом алгоритме процедура скрещивания реализуется с помощью равномерного кроссовера. Однако специфика задачи требует внесения корректировок в стандартную процедуру.

Возможно несколько ситуаций. Если на текущей позиции у обоих родителей находятся либо объекты загрузки, либо центры разгрузки, то из них случайным образом выбирается объект для соответствующего гена потомка. Если у одного из родителей рассматриваемый ген отвечает депо, то в силу парности такого объекта, необходимо рассмотреть два рядом стоящие гена второго родителя и случайный выбор проводить между выбранными парами генов. При следующем обнаружении объекта депо у родителей соответствующая пара генов потомка будет заполняться генами родителя, выбранного ранее. Данная процедура необходима для сохранения количества маршрутов в хромосоме.

На Рисунке 3 продемонстрирован принцип работы процедуры скрещивания.

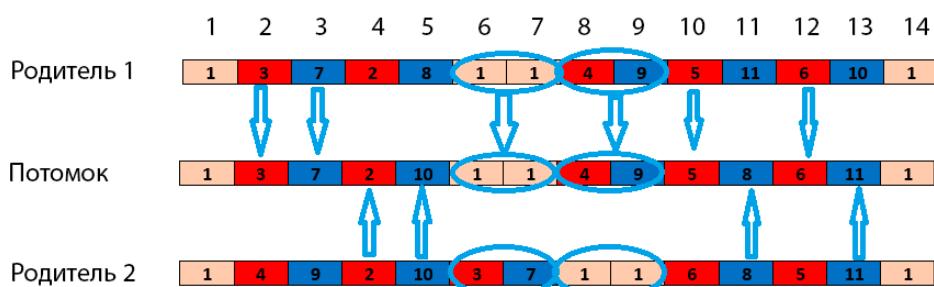


Рисунок 3 – Иллюстративный пример процедуры скрещивания
 Figure 3 – An illustrative example of the crossing procedure

В приведенном примере гены со 2 по 5 и с 10 по 13 отвечают первой из рассмотренных ситуаций. Для них объекты в потомка выбираются случайным образом из соответствующих генов родителей. В генах с номерами 6 и 7 у первого родителя находятся объекты депо, поэтому выбор проводится из пар генов родителей. В рассмотренном примере случайный выбор пал на первого из родителей. Следующее появление объектов депо фиксируется у второго родителя в генах с номерами 8 и 9. При этом соответствующие гены потомка заполняются объектами из выбранного ранее первого родителя.

Псевдокод предложенного алгоритма скрещивания представлен ниже.

Скрещивание(Родитель1, Родитель2)

Потомок = []

Потомок.Добавить(Депо)

Для $i = 1$ до Длина(Хромосома)-1:

Если Потомок[i] is NULL:

Если Родитель1[i] = Депо или Родитель2 = Депо: #если встретилось депо

$x = \text{random}(1,2)$ #случайный выбор родителя

Потомок.Добавить(Родитель{ x }[i])

Потомок.Добавить(Родитель{ x }[$i+1$])

#Добавление пары генов: целевого объекта и центра или два депо

Для $j = i+2$ до Длина(Хромосома)-1:

Если Родитель1[i] = Депо или Родитель2 = Депо:

#Поиск следующего депо

Потомок[j] = Родитель{ x }[i]

Потомок[$j+1$] = Родитель{ x }[$i+1$]

#Добавление на месте следующего депо генов того же родителя

Break

$i = i+1$

Иначе:

Ген =

$$\begin{cases} \text{random}(\text{Род1}[i], \text{Род2}[i]), \text{ если } (\text{Род}[i] \in N) \text{ или } (\text{Род}[i] \in M \text{ и } \text{Род}[i] \neg\in \text{Chromosome}) \\ \text{Род1}[i], \text{ если } \text{Род}[i] \in M \text{ и } \text{Род2}[i] \in \text{Chromosome} \\ \text{Род2}[i], \text{ если } \text{Род}[i] \in M \text{ и } \text{Род1}[i] \in \text{Chromosome} \\ \text{random}(M), \text{ если } \text{Род}[i] \in M \text{ и } \text{Род}[i] \in \text{Chromosome} \text{ и } \text{random}(M) \neg\in \text{Chromosome} \end{cases}$$

Добавить(Ген)

Потомок.Добавить(Депо)

Алгоритм, описанный в фигурных скобках, необходим для сохранения требования чередования и исключения повторения объектов загрузки в хромосоме. Если указатель стоит на гене, отвечающим за центр разгрузки, то выбор родителя происходит случайным образом. Если указатель находится на гене, соответствующем объекту загрузки, то возможны три ситуации. Первый вариант подразумевает, что ни один из рассматриваемых объектов не числится в формируемой хромосоме потомка. Тогда объект выбирается случайно из соответствующих генов обоих родителей. Второй вариант предполагает, что текущий объект одного из родителей в хромосоме потомка уже встречался, а другого нет. В этом случае выбирается принудительно тот объект, который еще не встречался в хромосоме потомка. Третий вариант возникает, когда гены обоих родителей уже присутствуют в хромосоме. Тогда данный ген потомка формируется случайным образом из объектов загрузки, которые еще не присутствуют в хромосоме.

Для обеспечения разнообразия хромосом в популяции и избегания локальных экстремумов в генетическом алгоритме предусмотрена процедура мутации. В рамках рассматриваемой задачи предлагается три равновероятных варианта оператора мутации.

Первый оператор направлен на изменение последовательности объектов в текущей хромосоме. При этом выбираются два гена одного вида (объекты загрузки или центры выгрузки) и их позиции меняются местами. Пример такого преобразования хромосомы представлен на Рисунке 4.

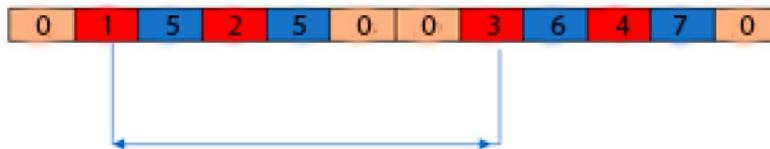


Рисунок 4 – Иллюстрационный пример изменения последовательности объектов
Figure 4 – An illustrative example of changing the sequence of objects

Второй вариант оператора мутации ориентирован на изменение протяженностей маршрутов транспортных средств. С этой целью предлагается менять местоположение пары объектов, отвечающих депо. Случайным образом выбирается пара подряд идущих генов загрузки и разгрузки и меняется местами с соответствующими объектами депо (Рисунок 5).

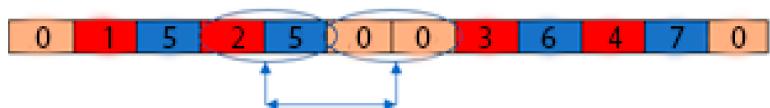


Рисунок 5 – Иллюстрационный пример изменения местоположения депо
Figure 5 – An illustrative example of a change in depot location

По условию задачи все объекты загрузки должны встретиться в каждой хромосоме ровно по одному разу. Однако на центры разгрузки требование обязательного появления в маршрутах не накладывается. В связи с этим возможна ситуация, когда некоторый центр разгрузки будет не задействован в хромосомах. При этом многообразие потомков существенно сократится. Для избегания такого вырождения популяции предлагается третий вариант оператора мутации, который заключается в случайной замене гена, отвечающего центру разгрузки, на другой объект того же типа (Рисунок 6).

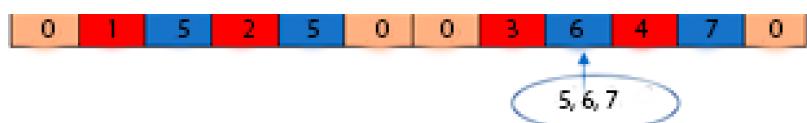


Рисунок 6 – Иллюстрационный пример изменения центра разгрузки
Figure 6 – An illustration of the third mutation operator in action

Результаты

Предложенный в статье алгоритм решения программно реализован на языке Python. Проведена серия вычислительных экспериментов с целью анализа быстродействия и эффективности разработанного алгоритма, а также оценки влияния параметров на полученный результат.

Входными данными для экспериментов являются матрица расстояний между объектами загрузки, центрами разгрузки и депо для задач разной размерности, а также количество транспортных средств. Матрица расстояний вычисляется через координаты точек, случайным образом распределенных на плоскости размера 100×100 условных единиц.

Для каждой размерности рассматривалось 500 сгенерированных таким образом матриц расстояний и вычислялись средние значения длин полученных маршрутов.

В первом эксперименте исследуется влияние размера популяции на результирующие маршруты при разном количестве итераций алгоритма. Оценивается максимальное значение из длин полученных маршрутов агентов. Результаты эксперимента приведены в Таблице 1.

Таблица 1 – Влияния размера популяции и количества итераций алгоритма на длины результирующий маршрутов

Table 1 – The influence of population size and the number of algorithm iterations on the lengths of the resulting routes

Кол-во объектов загрузки	Кол-во центров разгрузки	Кол-во агентов	Размер популяции	Кол-во итераций			
				1000	2500	5000	10000
10	5	2	10	821	796	796	796
10	5	2	15	796	796	796	796
20	5	2	10	1433	1256	1231	1206
20	5	2	20	1242	1198	1124	1101
20	5	5	10	876	794	731	708
20	5	5	20	763	728	702	694
20	5	5	30	721	703	687	679
50	10	5	20	2132	2036	1964	1899
50	10	5	40	1985	1896	1845	1760
50	10	5	60	1856	1734	1684	1684
50	10	10	20	1145	1021	976	926
50	10	10	40	1048	947	845	807
50	10	10	60	894	843	802	802
100	10	5	20	3582	3216	3142	3056
100	10	5	50	3154	2994	2855	2784
100	10	5	100	2846	2467	2265	2156
100	20	10	20	1954	1743	1689	1618
100	20	10	50	1739	1573	1532	1437
100	20	10	100	1513	1474	1368	1323

Из данных, представленных в Таблице 1, можно сделать вывод, что выбирать размер популяции стоит исходя из размерности задачи. Рекомендуется выбирать количество особей в популяции близкое к общему количеству точек, присутствующему в условиях задачи. При относительно небольших размерностях задачи (до 50 точек), достигнуть оптимального решения можно в пределах 5000 итераций. При больших размерностях задачи (50 и более точек) 5000 итераций не хватит для нахождения решения, близкого к оптимальному, в связи с чем необходимо увеличивать количество итераций до 10000 и более.

Во втором вычислительном эксперименте сравниваются результаты работы алгоритма при разных значениях коэффициента мутации для выявления влияния этого параметра на результат работы алгоритма. Размер популяции взят равным размерности задачи. В задачах участвуют два мобильных объекта. Результаты эксперимента приведены в Таблице 2.

Таблица 2 – Влияние коэффициента мутации на длины результирующих маршрутов
Table 2 – The influence of the mutation coefficient on the lengths of the resulting routes

Кол-во объектов загрузки	Кол-во центров разгрузки	Коэффициент мутации	1000 итераций	2500 итераций	5000 итераций
15	5	0	789	789	789
15	5	0.05	764	745	745
15	5	0.30	745	745	745
15	5	0.70	752	745	745
50	10	0	6724	3676	3126
50	10	0.05	3462	2476	2176
50	10	0.30	1718	1718	1718
50	10	0.70	2134	1718	1718
100	20	0	86424	64712	64712
100	20	0.05	30726	27142	22411
100	20	0.30	17642	14731	11862
100	20	0.70	26154	18743	12798

Из эксперимента можно сделать вывод, что в рамках поставленной задачи лучше всего использовать коэффициент мутаций, равный 0,30. Отсутствие мутации или слишком маленькие значения коэффициента, порядка 0,05, приводят к застреванию в локальном минимуме, из-за чего на поиск решения требуется слишком большое количество итераций. Большое значение коэффициента порядка 0,70 вносит большую долю случайности в алгоритм, из-за чего на поиск решения также требуется большое количество итераций.

В третьем вычислительном эксперименте сравниваются результаты работы предложенного алгоритма с результатами точного метода ветвей и границ [14]. В эксперименте оценивается суммарная длина всех построенных маршрутов мобильных объектов. Критерием останова является неизменность значения целевой функции за последние 200 итераций. Результаты эксперимента приведены в Таблице 3.

Таблица 3 – Длины результирующих маршрутов при использовании генетического алгоритма и метода ветвей и границ

Table 3 – Lengths of resulting routes when using the genetic algorithm and the branch and bound method

Кол-во объектов загрузки	Кол-во центров разгрузки	Кол-во агентов	Средняя длина пути	Метод ветвей и границ	Время работы генетического алгоритма, мс	Время работы точного метода, мс
10	1	1	714	714	<1	<1
10	2	2	539	527	<1	59
10	5	5	359	336	1	6274
10	10	10	328	295	1,1	>30000
30	1	1	2334	2276	1,2	1032
30	2	2	1626	1548	1,5	10657
30	5	5	925	877	12	>20000
30	10	10	704	611	12,5	>50000
50	1	1	3449	3374	21	>15000

Из эксперимента можно сделать вывод, что при использовании предложенного алгоритма среднее отклонение полученных значений от оптимального решения методом

ветвей и границ равно 7,2 %. При этом время работы алгоритма в разы меньше времени, затраченного методом ветвей и границ на поиск оптимального решения.

В четвертом вычислительном эксперименте сравниваются максимальная и минимальная длины путей транспортных средств. Маршруты пронумерованы от 1 до 5 по убыванию полученных значений. Результаты эксперимента приведены в Таблице 4.

Таблица 4 – Длины маршрутов агентов при разных размерностях задачи
Table 4 – Lengths of routes of agents for different problem dimensions

Неподвижные объекты × Агенты	1-й маршрут	2-й маршрут	3-й маршрут	4-й маршрут	5-й маршрут	Разница (%)
15 × 2	745	717	–	–	–	3,9
15 × 3	513	496	487	–	–	5,3
15 × 5	312	299	295	287	282	10,6
50 × 2	1718	1629	–	–	–	5,5
50 × 3	1217	1164	1137	–	–	7,0
50 × 5	704	672	658	649	649	8,4
100 × 2	11862	11719	–	–	–	1,2
100 × 3	7424	7274	7213	–	–	2,9
100 × 5	5124	4998	4904	4904	4889	4,9

Из эксперимента можно сделать вывод, что при увеличении количества агентов, участвующих в задаче, разница между максимальной и минимальной длиной маршрута в процентном соотношении увеличивается, зависимость от размерности задачи отсутствует.

Заключение

В статье рассмотрена задача маршрутизации нескольких агентов с требованием чередования в их маршрутах объектов двух типов. Целью является минимизация разброса времен выполнения полученных маршрутов. Для поставленной задачи предложен генетический алгоритм решения, этапы которого модифицированы с учетом специфики поставленной задачи. Разработаны рекомендации по настройке параметров алгоритма в зависимости от размерности решаемой задачи. Также проведен сравнительный анализ разработанного алгоритма с точным методом ветвей и границ. Показано, что предложенный подход к решению задачи маршрутизации позволяет получить решение, близкое к оптимальному, за несопоставимо меньшее время.

СПИСОК ИСТОЧНИКОВ / REFERENCES

1. El Bouyahiyiou K. The Selective Full Truckload Multi-Depot Vehicle Routing Problem with Time Windows: Formulation and a Genetic Algorithm. *International Journal of Supply and Operations Management*. 2022;9(3):299–320. <https://doi.org/10.22034/ijsom.2022.109076.2168>
2. Mbiadou Saleu R.G., Deroussi L., Feillet D., Grangeon N., Quilliot A. The Parallel Drone Scheduling Problem with Multiple Drones and Vehicles. *European Journal of Operational Research*. 2022;300(2):571–589. <https://doi.org/10.1016/j.ejor.2021.08.014>
3. Головцов Д.Л. Задача маршрутизации судов с различной грузоподъёмностью морского транспортного комплекса арктической зоны России. *Вестник государственного университета морского и речного флота им. адмирала С.О. Макарова*. 2015;(6):85–92.

- Golovcov D.L. Vessels Routing Problem with Heterogeneous Fleetin Marine Transportation Complex Arctic Zone of Russia. *Vestnik gosudarstvennogo universiteta morskogo i technologicheskogo flota im. admirala S.O. Makarova*. 2015;(6):85–92. (In Russ.).
4. Lin Sh.-W., Yu V.F., Chou Sh.-Y. Solving the Truck and Trailer Routing Problem Based on a Simulated Annealing Heuristic. *Computers & Operations Research*. 2009;36(5):1683–1692. <https://doi.org/10.1016/j.cor.2008.04.005>
5. Semet F., Taillard E. Solving Real-Life Vehicle Routing Problems Efficiently Using Tabu Search. *Annals of Operations Research*. 1993;41(4):469–488. <https://doi.org/10.1007/BF02023006>
6. Villegas J.G., Prins Ch., Prodhon C., Medaglia A.L., Velasco N. A GRASP with Evolutionary Path Relinking for the Truck and Trailer Routing Problem. *Computers & Operations Research*. 2011;38(9):1319–1334. <https://doi.org/10.1016/j.cor.2010.11.011>
7. Гвоздев Л.Р., Медведева Т.А. Решение задачи маршрутизации транспортных средств с временными окнами с помощью алгоритма муравьиных колоний. *Молодой исследователь Дона*. 2022;(3):58–61.
Gvozdev L.R., Medvedeva T.A. Solving the Problem of Vehicles Routing with Time Windows Using the Ant Colony Algorithm. *Young Don Researcher*. 2022;(3):58–61. (In Russ.).
8. Долгова О.Э., Пересветов В.В. Муравьиный алгоритм с ослаблением ограничений по временным окнам в решении задачи маршрутизации транспорта. *Вычислительные технологии*. 2018;23(5):49–62. <https://doi.org/10.25743/ICT.2018.23.5.005>
Dolgova O.E., Peresvetov V.V. An Ant Colony Optimization Algorithm with Time Relaxed Window Constraints for Solving the Vehicle Routing Problem. *Computational Technologies*. 2018;23(5):49–62. (In Russ.). <https://doi.org/10.25743/ICT.2018.23.5.005>
9. Lenstra J.K., Rinnooy Kan A.H.G. Complexity of Vehicle Routing and Scheduling Problems. *Networks*. 1981;11(2):221–227. <https://doi.org/10.1002/net.3230110211>
10. Posada A., Rivera J.C., Palacio J.D. Selective Vehicle Routing Problem: A Hybrid Genetic Algorithm Approach. In: *Artificial Evolution: 14th International Conference, Evolution Artificielle, EA 2019, 29–30 October 2019, Mulhouse, France*. Cham: Springer; 2020. P. 148–161. https://doi.org/10.1007/978-3-030-45715-0_12
11. Medvedev S.N., Medvedeva O.A., Zueva Y.R., Chernyshova G.D. Formulation and Algorithmization of the Interleaved Vehicle Routing Problem. In: *Journal of Physics: Conference Series: Volume 1203: International Conference "Applied Mathematics, Computational Science and Mechanics: Current Problems", AMCSM 2018, 17–19 December 2018, Voronezh, Russia*. Institute of Physics Publishing; 2019. <https://doi.org/10.1088/1742-6596/1203/1/012053>
12. Ghaziri H. Supervision in the Self-Organizing Feature Map: Application to the Vehicle Routing Problem. In: *Meta-Heuristics: Theory and Applications*. New York: Springer; 1996. P. 651–660. https://doi.org/10.1007/978-1-4613-1361-8_39
13. Mingozi A., Bianco L., Ricciardelli S. Dynamic Programming Strategies for the Traveling Salesman Problem with Time Window and Precedence Constraints. *Operations Research*. 1997;45(3):365–377. <https://doi.org/10.1287/opre.45.3.365>
14. Медведев С.Н. Математическая модель и алгоритм решения задачи маршрутизации транспортных средств с несколькими центрами с чередованием и единым местом сбора. *Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии*. 2021;(1):21–32. <https://doi.org/10.17308/sait.2021.1/3368>

- Medvedev S.N. A Mathematical Model and an Algorithm for Solving a Multi-Depot Vehicle Routing Problem with a Single End Point. *Proceedings of Voronezh State University. Series: Systems Analysis and Information Technologies.* 2021;(1):21–32. (In Russ.). <https://doi.org/10.17308/sait.2021.1/3368>
15. Medvedev S.N. Greedy and Adaptive Algorithms for Multi-Depot Vehicle Routing with Object Alternation. *Automation and Remote Control.* 2023;84(3):305–325. <https://doi.org/10.1134/S0005117923030086>

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

Медведев Сергей Николаевич, кандидат физико-математических наук, доцент, декан факультета прикладной математики, информатики и механики Воронежского государственного университета, Воронеж, Российская Федерация.

e-mail: s_n_medvedev@mail.ru

Трифонов Антон Георгиевич, аспирант кафедры вычислительной математики и прикладных информационных технологий факультета прикладной математики, информатики и механики Воронежского государственного университета, Воронеж, Российская Федерация.

e-mail: anton.trifonov@mail.ru

Статья поступила в редакцию 27.11.2025; одобрена после рецензирования 16.12.2025; принята к публикации 22.12.2025.

The article was submitted 27.11.2025; approved after reviewing 16.12.2025; accepted for publication 22.12.2025.

Olga A. Medvedeva, Candidate of Physico-mathematical Sciences, Docent, Associate Professor at the Department of Computational Mathematics and Applied Information Technologies, Faculty of Applied Mathematics, Computer Science and Mechanics, Voronezh State University, Voronezh, the Russian Federation.

Sergey N. Medvedev, Candidate of Physico-mathematical Sciences, Docent, Dean of Faculty of Applied Mathematics, Computer Science and Mechanics, Voronezh State University, Voronezh, the Russian Federation.

Anton G. Trifonov, Postgraduate at the Department of Computational Mathematics and Applied Information Technologies, Faculty of Applied Mathematics, Computer Science and Mechanics, Voronezh State University, Voronezh, the Russian Federation.