



Международный журнал информационных технологий и энергоэффективности

Сайт журнала: <http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004.021

АНАЛИЗ ЭФФЕКТИВНОСТИ АЛГОРИТМА МУРАВЬИНОЙ КОЛОНИИ В ЗАДАЧЕ МАРШРУТИЗАЦИИ ТРАНСПОРТНЫХ СРЕДСТВ С ОГРАНИЧЕННОЙ ВМЕСТИМОСТЬЮ

Гергедава Д.А.

ФГБОУ ВО «МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)», Москва, Россия, (125080, город Москва, Волоколамское ш., д. 4), e-mail: pusher123toxic@gmail.com

В условиях современной транспортной логистики, где эффективность и оптимизация играют ключевую роль, задача маршрутизации транспортных средств с ограниченной вместимостью представляет актуальную проблему. С увеличением объема грузов и строгих временных ограничений поиск оптимальных маршрутов становится сложной задачей.

В данном контексте алгоритмы, вдохновленные природными процессами, привлекают внимание исследователей. Алгоритм муравьиной колонии, основанный на коллективном поведении муравьев при поиске пищи, представляет перспективный подход к решению задач маршрутизации.

Статья направлена на анализ эффективности алгоритма муравьиной колонии в решении задачи маршрутизации транспортных средств с ограниченной вместимостью, а также анализ параметров, влияющих на эффективность алгоритма.

Ключевые слова: Популяционные алгоритмы, муравьиный алгоритм, автоматизация, задача маршрутизации транспорта, комбинаторная оптимизация.

ANALYSIS OF THE ANT COLONY ALGORITHM'S EFFICIENCY IN THE CAPACITY VEHICLE ROUTING PROBLEM

Gergedava D.A.

MOSCOW AVIATION INSTITUTE (NATIONAL RESEARCH UNIVERSITY), Moscow, Russia, (4, Volokolamskoye shosse, Moscow, 125080, Russian Federation), e-mail: pusher123toxic@gmail.com

In the context of modern transportation logistics, where efficiency and optimization play a crucial role, the problem of routing vehicles with limited capacity is a pertinent issue. With the increasing volume of cargo and strict time constraints, finding optimal routes becomes a challenging task.

In this context, algorithms inspired by natural processes attract researchers' attention. The ant colony algorithm, based on the collective behavior of ants in search of food, represents a promising approach to solving routing problems.

The article aims to analyze the efficiency of the ant colony algorithm in solving the problem of routing vehicles with limited capacity, as well as to analyze the parameters influencing the algorithm's effectiveness.

Keywords: Population algorithms, ant algorithm, automation, vehicle routing problem, combinatorial optimization.

Проблема оптимизации маршрутов транспортных средств представляет собой сложный класс задач комбинаторной оптимизации, возникающих в контексте логистических и снабженческих систем, где требуется эффективное перемещение товаров и услуг от

поставщиков до конечных потребителей. Эта проблема становится особенно актуальной в современной экономической среде, где динамичность рынка и быстрое развитие технологий создают необходимость в оптимизации процессов доставок [1].

Существует большое количество разновидностей задачи маршрутизации транспорта (VRP), отличающихся количеством и природой накладываемых ограничений [2]. Наиболее известные из них задача динамической маршрутизации транспортных средств (DVRP), иногда называемая задачей онлайн-маршрутизации, относительно недавно возникшая в связи с достижениями в области информационных технологий, которые позволяют получать и обрабатывать информацию в режиме реального времени, в DVRP некоторые заказы известны заранее, до начала рабочего дня, однако в течение дня системе приходится включать в изменяющийся график новые поступающие заказы, задача маршрутизации с ограниченными временными окнами (VRPTW), целью задачи VRPTW также является обслуживание клиентов с минимизацией общего пройденного расстояния, но при наличии дополнительного ограничения в виде заранее определенных временных интервалов, не нарушая ограничений по вместимости для каждого транспортного средства.

В этой статье мы сосредоточим внимание на задаче маршрутизации с ограниченной вместимостью (CVRP), являющейся усложнением классической VRP, но более простой чем VRPTW ввиду отсутствия временных окон. Любые из вариантов задачи маршрутизации представляют собой NP-сложную задачу комбинаторной оптимизации, поскольку так или иначе являются обобщением задачи коммивояжера [3]. Точные алгоритмы в 100% случаев позволяют получить оптимальное решение, однако крайне вычислительно затратны для NP-сложных задач, полный перебор имеет сложность $O(n!)$.

Ни один даже самый современный компьютер не способен в разумные сроки произвести такой объем вычислений, который привел бы к точному решению, уже даже для пары десятков вершин (Таблица 1).

Таблица 1 – Зависимость числа операций от размера задачи

n	$\frac{(n-1)!}{2}$	n	$\frac{(n-1)!}{2}$
3	1	8	2.520
4	3	9	20.160
5	12	10	181.440
6	60	20	6.082E+16
7	360	30	4.420E+30

Приведем математическую формулировку задачи маршрутизации с ограниченной грузоподъемностью (CVRP). Одним из основных ограничений является то, что суммарная масса груза на каждом маршруте грузовика не должна превышать его максимальной грузоподъемности. Кроме того, каждый клиент должен быть обслужен ровно один раз.

Пусть $G = (V, E)$ – граф, в котором $V = \{1 \dots n\}$ – набор вершин, задающих города (или клиентов) с центральным хабом, который без ограничения общности можно считать расположенным в вершине 1. А E – набор ребер, соответствующий дорогам, соединяющим города (клиентов). Вводится также матрица расстояний $C = (c_{ij})$, часто значения этой матрицы могут интерпретироваться в более общем смысле, как стоимость поездки из вершины i в j .

Минимизируем

$$\sum_{i \neq j} c_{ij} x_{ij} \rightarrow \min \quad (1)$$

Со следующими ограничениями

$$\left\{ \begin{array}{l} \sum_{e \in \delta(\{i\})} x_e = 2 \quad \forall i \in V \quad (2) \\ \sum_{e \in \delta(\{0\})} x_e = 2K \quad (3) \\ \sum_{e \in \delta(S)} x_e \geq 2k(S) \quad \forall S \subseteq V \quad (4) \\ x_e \leq 1 \quad \forall e \in E \quad \delta(\{0\}) \quad (5) \\ x_e \geq 0 \quad \forall e \in E \quad (6) \end{array} \right.$$

Здесь S – подмножество V , т.е. множество вершин, входящих в конкретный маршрут, $\delta(S)$ – множество ребер в маршруте с вершинами из S , K – количество доступных транспортных средств, $k(S)$ – это целая часть от деления суммы спросов вершин в множестве S на максимальную грузоподъемность грузовика C .

Ограничение (2) утверждает, что каждый клиент посещается каким-либо транспортным средством один раз, тогда как ограничение (3) утверждает, что K транспортных средств должны выезжать и въезжать на территорию склада. Ограничение (4) – это неравенство пропускной способности, которое требует, чтобы все подмножества обслуживались достаточным количеством транспортных средств. Ограничения (5), (6) требуют, чтобы каждое ребро, не примыкающее к депо, могло использоваться дважды, когда маршрут обслуживает только одного клиента.

Как было сказано ранее, для решения задачи будет использоваться алгоритм муравьиной колонии, дополнительно модифицированный таким образом, чтобы учитывать введенные ограничения, приведем математическую формулировку алгоритма.

Фактически алгоритм использует набор искусственных муравьев (особей), которые сотрудничают в решении задачи путем обмена информацией с помощью феромонов, нанесенных на ребра графа.

Вероятность перехода муравья из i –ой вершины в j –ую определяется следующей формулой

$$P_{ij} = \frac{\tau_{ij}^\alpha \eta_{ij}^\beta}{\sum_{m \in A} \tau_{im}^\alpha \eta_{im}^\beta} \quad (7)$$

τ_{ij} – количество феромона между городами i и j

η_{ij} – величина обратная расстоянию между городами, т.е. вероятность выбора города пропорциональна близости этого города и количеству феромона по дороге в него

A – все доступные для посещения города из текущей позиции муравья

Константы α и β определяют степень влияния близости следующей вершины и количества феромонов на пути на выбор муравья о переходе в новую вершину.

Распределение феромона для муравья с индексом k по граням (дорогам), описывается следующими формулами

$$\Delta\tau_{ij,k}(t) = \begin{cases} \frac{Q}{L_k(t)}, & \text{если } (i, j) \in T_k(t), \\ 0, & \text{если } (i, j) \notin T_k(t), \end{cases} \quad (8)$$

$$\tau_{ij}(t+1) = (1-p) \cdot \tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij,k}(t), \quad (9)$$

Добавка феромона, которую делает муравей k между i -ым и j -ым городом на итерации t равна отношению некоторой константы Q к длине маршрута $L_k(t)$, пройденного муравьем k , при условии, что эти два города являются частью пройденного муравьем маршрута $(i, j) \in T_k(t)$.

Количество феромона между городами i и j на новой итерации $\tau_{ij}(t+1)$ равно сумме количества феромона между городами на предыдущей итерации $\tau_{ij}(t)$, умноженной на коэффициент испарения $(1-p)$ и суммы всех новых порций феромона, которые были отложены всеми муравьями на данном участке $\sum_{k=1}^m \Delta\tau_{ij,k}(t)$.

Временная сложность этого алгоритма зависит от времени жизни колонии t (число итераций), количества вершин графа n и числа муравьев m , и определяется как $O(tn^2m)$.

Алгоритм был реализован на языке программирования Python 3.9 с использованием аппаратного обеспечения на базе четырехъядерного процессора Intel Core I3-12100F с частотой 3.5 ГГц и оперативной памятью Kingston KF426C16BB2A объемом 16 Гб 2666 МГц.

Основные методы, определенные в классе:

- **solutionOfOneAnt(vertices, edges, capacityLimit, demand, pheromones, depot)** - метод строит маршрут для каждого транспортного средства из депо для одной особи из колонии, поочередно выбирая следующий город с учетом вероятности, зависящей от феромонов и стоимости перемещения, маршрут завершается, когда транспортное средство достигает своей максимальной вместимости;
- **updatePheromones(pheromones, solutions)** – этот метод обновляет матрицу феромонов, сначала происходит испарение феромонов в соответствии с коэффициентом испарения для всех элементов матрицы. Затем, для каждого маршрута в каждом решении, обновляются феромоны на соответствующих ребрах графа. Количество отложенных феромонов обратно пропорционально длине маршрута. Таким образом, более оптимальные маршруты вносят больший вклад в увеличение феромонов;

- **rateTour(solution)** – в этом методе для каждого маршрута в решении вычисляется общая стоимость, учитывая стоимость перемещения между городами, возвращает список маршрутов и соответствующих им затрат.

Данные тестовых экземпляров считываются из текстового документа, содержащем евклидовы координаты вершин на плоскости, ограничение вместимости, количество доступных транспортных средств и величину спроса для каждой вершины, для экспериментальных вычислений используются общедоступные экземпляры из CVRPLIB. Алгоритм инициализируется со следующими параметрами (Таблица 2).

Таблица 2 – Начальные параметры алгоритма

Параметр	Значение
α	2.4
β	2.4
ρ	0.1
Q	5
Количество итераций	1000

Для проведения вычислительных экспериментов используется группа экземпляров Set E (Christofides and Eilon, 1969) (Рисунок 1), выбор обоснован широкой вариативностью размерности задач, количества используемых транспортных средств и грузоподъемности, что позволит сделать более комплексные выводы о производительности алгоритма.

Instance	n	K	Q	UB	Opt
E-n13-k4	12	4	6000	247	yes
E-n22-k4	21	4	6000	375	yes
E-n23-k3	22	3	4500	569	yes
E-n30-k3	29	3	4500	534	yes
E-n31-k7	30	7	140	379	yes
E-n33-k4	32	4	8000	835	yes
E-n51-k5	50	5	160	521	yes
E-n76-k7	75	7	220	682	yes
E-n76-k8	75	8	180	735	yes
E-n76-k10	75	10	140	830	yes
E-n76-k14	75	14	100	1021	yes
E-n101-k8	100	8	200	815	yes
E-n101-k14	100	14	112	1067	yes

Рисунок 1 – Экземпляры набора Set E

Далее приводятся результаты вычислительных экспериментов в двух разных сценариях, в первом из них количество особей муравьев в популяции полагается равным количеству вершин в графе, во втором фиксируется значение 100 особей для всех экземпляров, что значительно больше количества вершин большей части из них и совпадает с ним для E-n101-k8 и E-n101-k14, нотация X-nxx-kxx означает следующее: X-буквенное наименование набора

данных из CVRPLIB, n – количество вершин, k – количество транспортных средств, обслуживающих клиентов.

Таблица 3 – Количество особей совпадает с количеством муравьев

Экземпляр	Муравьиный алгоритм	Верхняя граница	Погрешность %
E-n22-k4	383	375	2.14
E-n23-k3	579	569	2.03
E-n30-k3	522	534	-2.17
E-n33-k4	882	835	5.70
E-n51-k5	635	521	21.90
E-n76-k7	836	682	20.42
E-n76-k14	1174	1021	14.98
E-n101-k8	964	815	18.28
E-n101-k14	1272	1067	19.21

Таблица 4 – Количество особей равно 100

Экземпляр	Муравьиный алгоритм	Верхняя граница	Погрешность %
E-n22-k4	378	375	0.80
E-n23-k3	574	569	0.87
E-n30-k3	519	534	-2.89
E-n33-k4	862	835	3.23
E-n51-k5	619	521	18.80
E-n76-k7	824	682	20.82
E-n76-k14	1158	1021	13.41
E-n101-k8	958	815	17.54
E-n101-k14	1244	1067	16.58

Итоговое решение гораздо в меньшей степени выглядит как формализация естественной логики, маршруты, построенные АСО, в целом кажутся более беспорядочными, хотя разрыв остается удовлетворительным, на Рисунке 2. приведено сравнение решения полученного муравьиным алгоритмом с оптимальным решением.

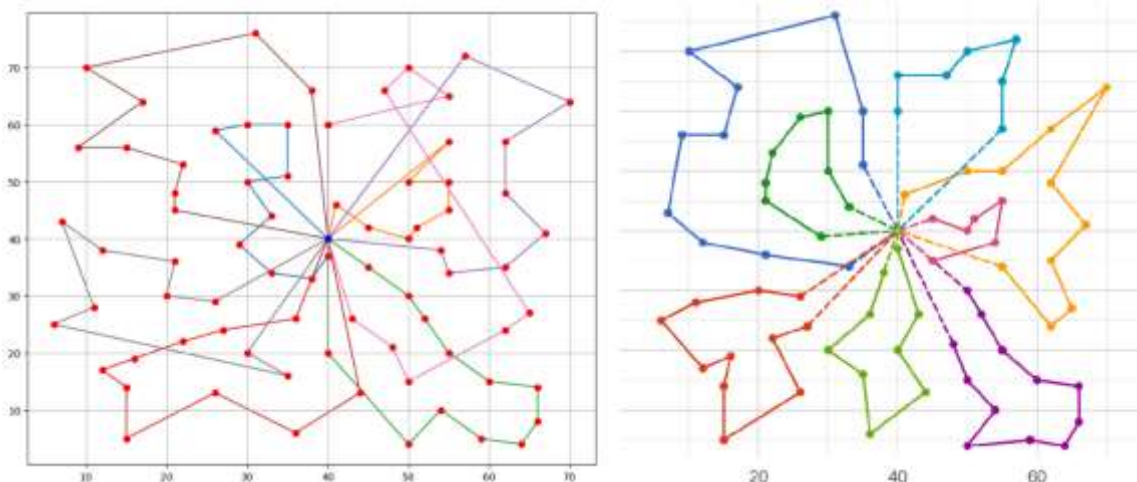


Рисунок 2 – Решения АСО и оптимальное решение для экземпляра E-n76-k10

Можно также заметить, что для экземпляра E-n30-k3 АСО достигает значительного улучшения, однако это объясняется увеличением количества используемых транспортных средств на одно, в связи с чем мы видим отличающиеся по структуре решения на Рисунке 3.

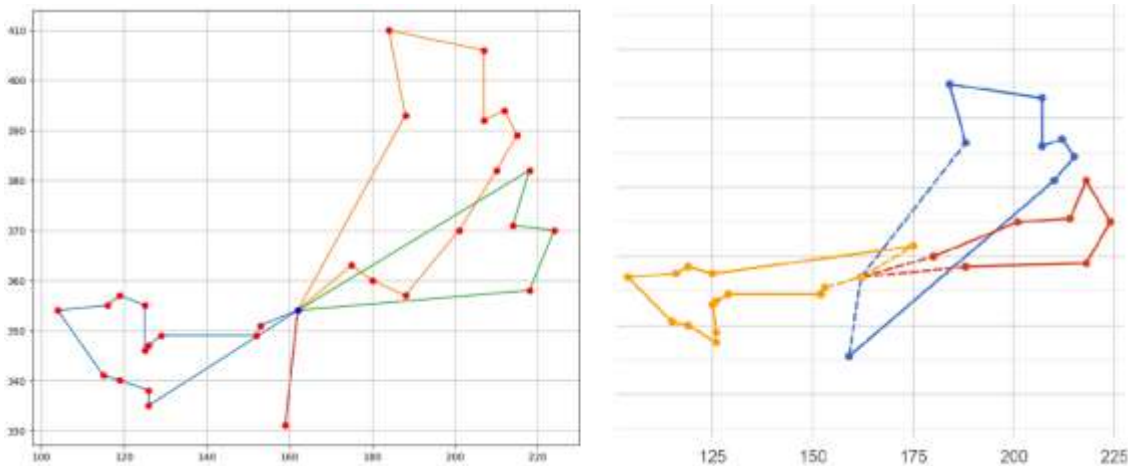


Рисунок 3 – Решения АСО и оптимальное решение для экземпляра E-n30-k3

При большем количестве муравьев алгоритм демонстрирует гораздо лучшие показатели сходимости, что можно наблюдать на графиках сходимости от числа итераций для некоторых экземпляров (Рисунок 4.)

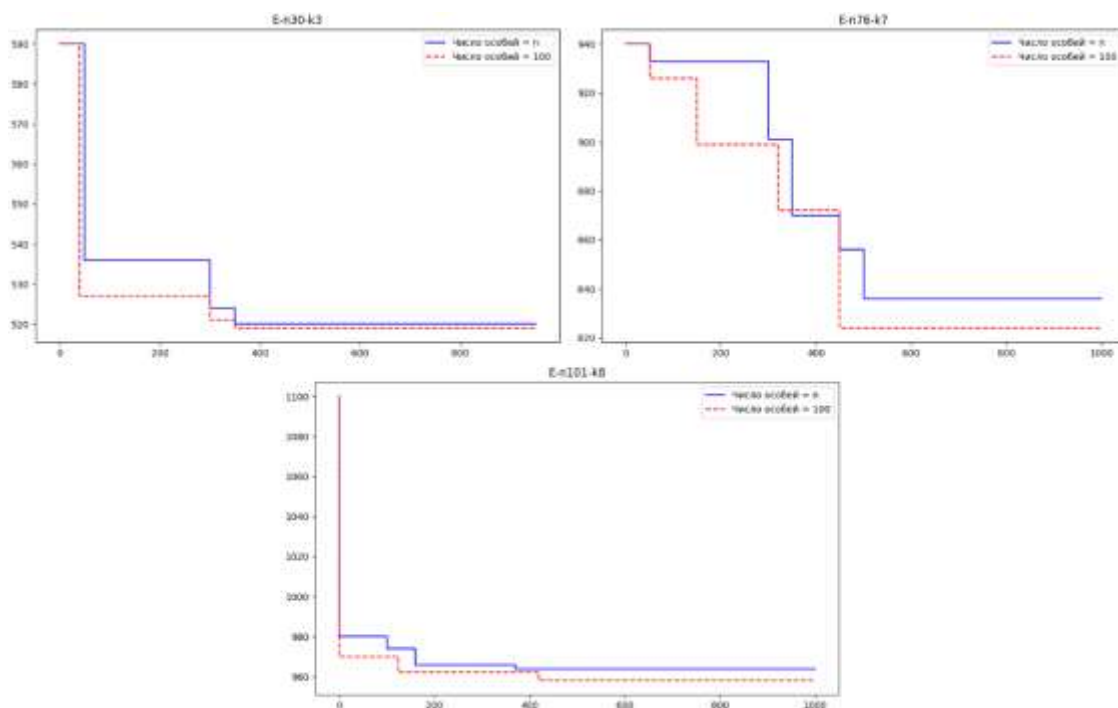


Рисунок 4 – Сходимость от числа итераций

Параметр количества муравьев сильно влияет на уровень сходимости и результирующее решение. Однако следует отметить, что увеличение количества муравьев влияет на время поиска решения (Рисунок 5.)

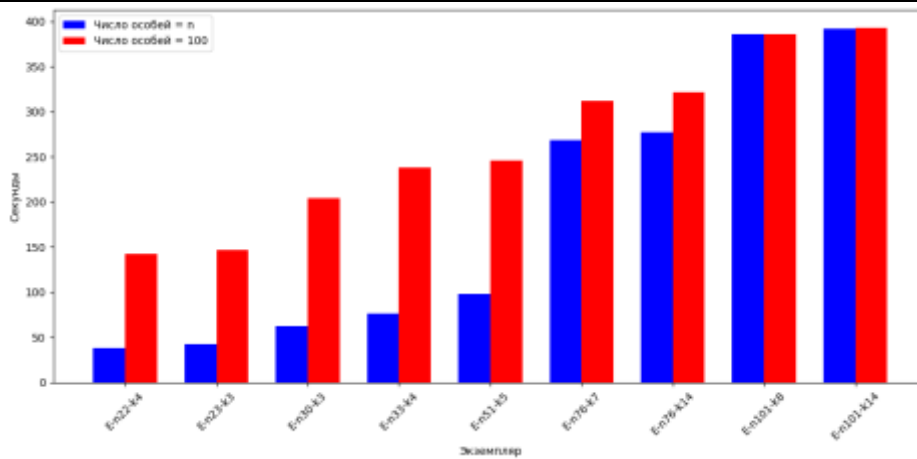


Рисунок 5 – Сравнение времени сходимости

Незначительное влияние значение параметра числа особей оказывает также на использование памяти, расход которой увеличивается примерно экспоненциально с увеличением размерности задачи (Рисунок 6.)

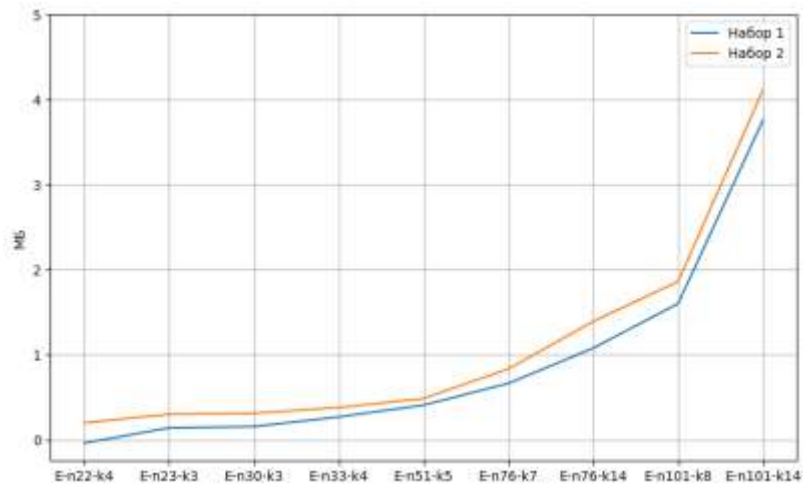


Рисунок 6 – Расход памяти при разном количестве особей

Далее анализирует степень влияния различных настраиваемых параметров на сходимость алгоритма, в изначальной конфигурации попеременно изменяется значение каждого из параметров в большую и меньшую сторону (Таблицы 5, 6.).

Таблица 5 – Влияние значений параметров на точность решения

Экземпляр	Муравьиный алгоритм								Верхняя граница
	$\alpha = 2$	$\alpha = 5$	$\beta = 2$	$\beta = 5$	$Q = 3$	$Q = 9$	$\tau_0 = 0.1$	$\tau_0 = 1$	
E-n22-k4	383	378	411	378	378	380	378	383	375
E-n23-k3	580	578	659	578	578	580	578	576	569
E-n30-k3	522	519	572	519	519	520	519	518	534
E-n33-k4	882	879	956	879	879	880	879	872	835
E-n51-k5	635	620	841	620	620	622	620	616	521
E-n76-k7	821	811	1172	811	811	812	620	616	682

Таблица 6 – Влияние значений параметров на точность решения

Экземпляр	Потребление памяти (МБ)		Время выполнения (сек)		Потребление памяти (МБ)		Время выполнения (сек)		Верхняя граница
	$Q = 3$	$Q = 9$	$Q = 3$	$Q = 9$	$\tau_0 = 0.1$	$\tau_0 = 1$	$\tau_0 = 0.1$	$\tau_0 = 1$	
E-n22-k4	0.12	0.1	29.82	29.94	0.1	0.14	28.78	31.5	375
E-n23-k3	0.12	0.13	32.51	32.98	0.13	0.18	31.44	32.79	569
E-n30-k3	0.23	0.24	73.47	73.84	0.22	0.29	62.78	67.14	534
E-n33-k4	0.25	0.25	80.07	80.62	0.24	0.33	80.07	85.62	835
E-n51-k5	0.57	0.57	254.28	254.37	0.57	0.68	247.62	289.28	521
E-n76-k7	3.51	3.53	853.5	855.19	3.6	4.2	849.4	897.34	682

Заключение

Полученные результаты экспериментальных вычислений дают возможность сделать вывод, что муравьиный алгоритм (ACO) представляет собой мощный инструмент для решения задач комбинаторной оптимизации, в частности, задачи маршрутизации с ограниченной грузоподъемностью (CVRP). Параметры Q и ρ – степень испарения, не оказывают существенного влияния на потребление памяти или время работы алгоритма, однако влияют на сходимость алгоритма и точность полученного решения, разница по потребляемой памяти варьируется в пределах 0,01 – 0,02 Мб, значительное влияние на точность оказывают параметры α и β , как и число итераций и число муравьев, однако системности не наблюдается, алгоритм требует точной настройки под конкретную задачу с учетом структуры имеющихся данных.

Список литературы

1. Мартин Кристофер. Логистика и управление цепочками поставок: создание сетей, создающих добавленную стоимость. Пирсон, 2016.

2. Саид Латераль, Карим Абу эль Мехди б, Мохаммед Саид Риффи. Проблема маршрутизации транспортных средств за последнее десятилетие: варианты, таксономия и метаэвристика, 2023.
3. Дахия С. & Сангван С. (2018). Обзор литературы по проблеме коммивояжера. Международный исследовательский журнал 5(16), С.1152-1155. Проверено 18 марта 2022 г.
4. Као Ю., Чен М. Х. и Хуан Ю. Т. (2012). Гибридный алгоритм, основанный на АСО и PSO для задач маршрутизации транспортных средств с емкостью. Математические проблемы в инженерии, 2012.
5. Нин Дж., Чжан К., Чжан С. & Чжан Б. (2018). Алгоритм оптимизации колонии муравьев, основанный на обновлении информации по наилучшему пути. Информационные науки, С.433-434.

References

1. Martin Christopher. Logistics and Supply Chain Management: Creating Value-Adding Networks. Pearson, 2016.
 2. Said Elatara, Karim Abouelmehdib, Mohammed Essaid Riffi. The vehicle routing problem in the last decade: variants, taxonomy and metaheuristics, 2023.
 3. Dahiya C. & Sangwan S. (2018). Literature Review on Travelling Salesman Problem. International Journal of Research 5(16), pp.1152-1155. Retrieved March 18, 2022
 4. Kao, Y., Chen, M. H., & Huang, Y. T. (2012). A Hybrid Algorithm Based on ACO and PSO for Capacitated Vehicle Routing Problems. Mathematical Problems in Engineering, 2012.
 5. Ning, J., Zhang, Q., Zhang, C., & Zhang, B. (2018). A Best-Path-Updating Information-Guided Ant Colony Optimization Algorithm. Information Sciences, pp.433–434.
-