

12. UL 268: Standard for Smoke Detectors for Fire Protective Signaling Systems, 2019.
13. Khan Zanis Ali & Shin Donghwan & Bianculli Domenico & Briand Lionel. Impact of log parsing on deep learning-based anomaly detection, *Empirical Software Engineering*, 2024, 29. 10.1007/s10664-024-10533-w.
14. Sellberg F & Buthke J & Sonne-Frederiksen Povl Filip & Nørkjær Gade Peter. Evaluating Four Types of Data Parsing Methods for Machine Learning Integration from Building Information Models, 2022.
15. Fan Gaolun. Random Forest Algorithm for Forest Fire Prediction, 2023. 10.1007/978-981-99-4554-2_15.
16. Pande Chaitanya & Radwan Neyara & Heddham Salim & Othman Kaywan & Alshehri Fahad & Pal Subodh & Pramanik Malay. Forecasting of monthly air quality index and understanding the air pollution in the urban city, India based on machine learning models and cross-validation, *Journal of Atmospheric Chemistry*, 2024, 82, pp. 1-26. 10.1007/s10874-024-09466-x.
17. Salman Hasan & Kalakech Ali & Steiti Amani. Random Forest Algorithm Overview, *Babylonian Journal of Machine Learning*, 2024, pp. 69-79. 10.58496/BJML/2024/007.
18. Bijan Ahmed & Al-Rahim Ali. Random Forest and Decision Tree Facies Classification Models for Well Log Data of the Mishrif Formation from Basrah Oil Company, Southern Iraq, *Iraqi Geological Journal*, 2025, 57, pp. 14-32. 10.46717/igj.57.2E.2ms-2024-11-11.
19. Al-Janabia Samaher & AlShourbaji Ibrahim & Patel Ahmed. Applied Predicative Modeling to Improve Recommendation Systems for Forecasting of Fire Occurrences, 2015.
20. Ramadhan Rafiq & Ashari Wahid. Performance Comparison of Random Forest and Decision Tree Algorithms for Anomaly Detection in Networks, *Journal of Applied Informatics and Computing*, 2024, 8, pp. 367-375. 10.30871/jaic.v8i2.8492.
21. Li Liping. Comparative Research on Diabetes Influencing Factors Based on Random Forest and Decision Tree Models, *Highlights in Science, Engineering and Technology*, 2023, 72, pp. 231-242. 10.54097/7m4x7j04.
22. Hu Yaowen. Comparison and Analysis of the Effectiveness of Linear Regression, Decision Tree, and Random Forest Models for Health Insurance Premium Forecasting, *Advances in Economics, Management and Political Sciences*, 2024, 79, pp. 347-353. 10.54254/2754-1169/79/20241754.
23. Cansler C. & Wright Micah & Mantgem Phillip & Shearman Timothy & Varner J. & Hood Sharon. Drought before fire increases tree mortality after fire, *Ecosphere*, 2024, 15. 10.1002/ecs2.70083.

Сингх Санны – Южный федеральный университет, e-mail: singkh@sfedu.ru; г. Таганрог, Россия; тел.: +79885751350; кафедра систем автоматического управления; аспирант; ассистент.

Прибыльский Алексей Васильевич – Южный федеральный университет; e-mail: apribylsky@sfedu.ru; г. Таганрог, Россия; тел.: +79885619718; кафедра систем автоматического управления; к.т.н.; доцент.

Косенко Евгений Юрьевич – Южный федеральный университет; e-mail: ekosenko@sfedu.ru; г. Таганрог, Россия; тел.: +78634394474; кафедра систем автоматического управления; к.т.н.; доцент.

Singh Sanni – Southern Federal University; e-mail: singkh@sfedu.ru; Taganrog, Russia; phone: +79885751350; the Department of Automatic Control Systems; graduate student; assistant.

Pribylskiy Alexey Vasilievich – Southern Federal University; apribylsky@sfedu.ru; Taganrog, Russia; phone: +79885619718; the Department of Automatic Control Systems; cand. of eng. sc.; associate professor.

Kosenko Evgeny Yuryevich – Southern Federal University; ekosenko@sfedu.ru; Taganrog, Russia; phone: +78634394474; the Department of Automatic Control Systems; cand. of eng. sc.; associate professor.

УДК 519.863

DOI 10.18522/2311-3103-2025-1-81-91

Д.Ю. Зорькин, Л.В. Самофалова, Н.В. Асанова

МУРАВЬИНЫЙ АЛГОРИТМ НА PYTHON

Данное исследование посвящено анализу и оптимизации муравьиного алгоритма для решения задачи коммивояжера, являющейся классической NP-трудной проблемой комбинаторной оптимизации. Основная цель работы – экспериментальная оценка влияния параметров алгоритма на качество и эффективность поиска приближённых решений, а также разработка рекомендаций по их адаптивной настройке. В качестве тестового набора данных использован стандартный граф Berlin52 из библиотеки TSPLIB, содержащий координаты 52 городов с известным оптимальным маршрутом длиной

7542 единицы. Эксперименты проводились в среде Python с использованием библиотеки ACO-Pants, реализующей муравьиный алгоритм. Была выполнена серия из 10 запусков с фиксированными параметрами: количество муравьёв (20), число итераций (100), коэффициенты влияния феромонов ($\alpha=1.0$) и расстояний ($\beta=2.0$), а также скорость испарения феромонов ($\rho=0.5$). Результаты показали среднее отклонение от оптимума в 1.85%, с лучшим найденным решением 7675.23 (отклонение 1.67%). Для повышения эффективности алгоритма исследованы адаптивные механизмы динамической настройки параметров: линейное увеличение α (до 2.0) и уменьшение β (до 3.0), снижение ρ (до 0.3), а также рост числа муравьёв (до 30). Это позволило сократить среднее отклонение до 1.70% и повысить стабильность решений. Особое внимание уделено анализу баланса между исследованием новых маршрутов и эксплуатацией накопленных данных. Установлено, что увеличение количества муравьёв улучшает качество решений, однако после 30 агентов прирост эффективности снижается. Динамическая корректировка параметров предотвращает преждевременную сходимость к локальным минимумам и ускоряет поиск глобально оптимальных путей. Визуализация динамики сходимости подтвердила быстрое уменьшение длины маршрута на первых 20 итерациях с последующей стабилизацией. Практическая значимость работы заключается в демонстрации гибкости муравьиного алгоритма для задач маршрутизации в логистике и сетевом планировании. Результаты показывают, что ACO превосходит универсальные методы (например, генетические алгоритмы) по вычислительной эффективности для TSP. Разработанные рекомендации по настройке параметров могут быть применены для масштабирования алгоритма на графы большей размерности. Исследование подчёркивает важность адаптивных подходов в метаэвристической оптимизации и открывает перспективы для дальнейшего улучшения алгоритма за счёт гибридизации с другими методами.

Муравьиный алгоритм; задача коммивояжёра; метаэвристические алгоритмы; оптимизация; маршрутизация; эвристические методы.

D.Y. Zorkin, L.V. Samofalova, N.V. Asanova

PYTHON ANT ALGORITHM

This study is devoted to the analysis and optimization of the ant colony algorithm for solving the traveling salesman problem, a classic NP-hard combinatorial optimization problem. The primary objective of the work is to experimentally assess the impact of the algorithm's parameters on the quality and efficiency of the search for approximate solutions, as well as to develop recommendations for their adaptive tuning. The standard Berlin52 graph from the TSPLIB library—containing the coordinates of 52 cities with a known optimal route length of 7542 units—was used as the test dataset. Experiments were conducted in a Python environment using the ACO-Pants library, which implements the ant colony algorithm. A series of 10 runs with fixed parameters was performed: number of ants (20), number of iterations (100), pheromone influence coefficient ($\alpha = 1.0$), distance coefficient ($\beta = 2.0$), and pheromone evaporation rate ($\rho = 0.5$). The results showed an average deviation from the optimum of 1.85%, with the best found solution being 7675.23 (a deviation of 1.67%). To enhance the algorithm's efficiency, adaptive mechanisms for dynamic parameter tuning were explored: a linear increase of α (up to 2.0) and a decrease of β (to 3.0), a reduction of ρ (to 0.3), as well as an increase in the number of ants (up to 30). These modifications reduced the average deviation to 1.70% and improved the stability of the solutions. Particular attention was paid to analyzing the balance between exploring new routes and exploiting accumulated data. It was found that increasing the number of ants improves the quality of solutions; however, beyond 30 agents, the efficiency gains diminish. Dynamic adjustment of the parameters prevents premature convergence to local minima and accelerates the search for globally optimal paths. Visualization of the convergence dynamics confirmed a rapid decrease in route length during the first 20 iterations, followed by subsequent stabilization. The practical significance of this work lies in demonstrating the flexibility of the ant colony algorithm for routing tasks in logistics and network planning. The results indicate that ACO outperforms general-purpose methods (for example, genetic algorithms) in computational efficiency for the TSP. The developed recommendations for parameter tuning can be applied to scale the algorithm to larger graphs. Overall, the study emphasizes the importance of adaptive approaches in metaheuristic optimization and opens up prospects for further improvements through hybridization with other methods.

Ant colony optimization; traveling salesman problem; metaheuristic algorithms; optimization; routing; heuristic methods.

Введение. Данная работа посвящена анализу методов оптимизации в решении задачи коммивояжёра. Ввиду её вычислительной сложности точные алгоритмы оказываются неэффективными для крупных графов. В связи с этим применяются метаэвристические или

эвристические алгоритмы, одним из которых является муравьиный алгоритм. Данный алгоритм позволяет находить короткие маршруты даже в условиях значительной размерности задачи на основе механизма феромонов и принципов коллективного поведения.

Актуальность темы обусловлена необходимостью поиска эффективных методов решения задачи коммивояжёра, которая находит широкое применение в логистике, транспорте и сетевом планировании. Методы эвристической оптимизации, включая муравьиные алгоритмы, позволяют находить качественные приближенные решения в разумные сроки.

Целью работы является анализ и экспериментальная оценка эффективности муравьиного алгоритма в решении задачи коммивояжёра на стандартных тестовых данных.

Для достижения поставленной цели были сформулированы следующие задачи:

- ◆ Изучение теоретических основ задачи коммивояжёра и методов её решения.
- ◆ Описание принципов работы муравьиного алгоритма.
- ◆ Проведение серии экспериментов на стандартном наборе данных Berlin52.
- ◆ Оценка влияния параметров алгоритма на качество решения.
- ◆ Разработка рекомендаций по настройке параметров алгоритма.

Объект исследования: методы эвристической оптимизации для решения задачи коммивояжёра.

Предмет исследования: применение муравьиного алгоритма для нахождения приближенных решений задачи коммивояжёра.

Исследование базируется на практическом применении теоретических методов, включая анализ алгоритмов оптимизации, математическое моделирование и проведение вычислительных экспериментов.

Постановка задачи. Задача заключается в нахождении гамильтонова цикла (маршрута), проходящего по всем вершинам графа ровно по одному разу и возвращающийся в исходную вершину, при котором суммарный вес (длина) цикла минимален.

$$G=(V, E), \quad (1)$$

где $V = \{1, 2, \dots, n\}$ – множество вершин (точек), а $E \subseteq V \times V$ – множество рёбер (связей между вершинами). Каждому ребру $(i, j) \in E$ приписан вес $d(i, j)$, интерпретируемый как «расстояние» перехода из вершины i в вершину j .

Пусть задано:

$$\min_{\pi \in \Pi} (d(\pi_1, \pi_2) + d(\pi_2, \pi_3) + \dots + d(\pi_{n-1}, \pi_n) + d(\pi_n, \pi_1)), \quad (2)$$

где Π – множество всех возможных перестановок вершин, а $\pi = (\pi_1, \pi_2, \dots, \pi_n)$ – конкретная последовательность обхода вершин.

Задача коммивояжёра является NP-трудной, поэтому точные алгоритмы поиска оптимального решения оказываются неэффективными для крупных графов [1].

Исследование опирается на вклад российских и зарубежных учёных в разработку метаэвристических методов. Так, работы Гладкова Л.А. [2] по интеллектуальным системам и Курейчика В.В. [3–5] по эволюционным вычислениям заложили основы адаптивной настройки параметров, включая динамическое изменение коэффициентов α , β и ρ .

Родзин С.И. [6] углубил анализ коллективного поведения в биоэвристиках, а Лебедев Б.К. предложил гибридные подходы, ставшие контекстом для сравнения с муравьиными алгоритмами [7, 8].

Среди зарубежных исследований МакКоннелл Дж. [9], Саймон Д. [10] рассмотрели универсальные методы оптимизации, тогда как Энгельбрехт А.П. [11], Дориго М. [12] разработали теоретическую базу роевого интеллекта и АСО, включая формулы вероятности перехода и обновления феромонов.

Госс С. [13], Блюм К. [14] дополнили алгоритм биологическими аналогиями и математическим анализом трендов, что позволило оптимизировать баланс между исследованием и эксплуатацией. Несмотря на широкий охват, работа недостаточно учитывает современные гибридные методы и ограничения АСО, такие как чувствительность к параметрам, что требует дальнейшего изучения [15].

Материалы и методы. Муравьиный алгоритм был впервые предложен Марко Дориго в 1992 году как метод поиска оптимальных решений для комбинаторных задач. Основная идея заключается в имитации коллективного поведения муравьёв, которые находят кратчайшие пути [1] от муравейника до источников пищи, оставляя [16] на своём пути феромоны, которые помогают другим муравьям находить эти пути. Таким образом, цель алгоритма – найти маршрут, проходящий через все вершины ровно один раз и возвращающийся в начальную точку, при котором суммарное расстояние минимально.

Рассмотрим принцип работы алгоритма (рис. 1).

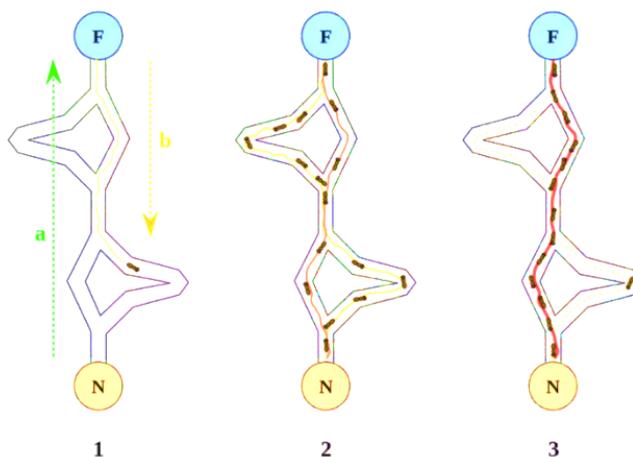


Рис. 1. Муравьиный алгоритм

На изображении представлена схема, иллюстрирующая принцип работы муравьиного алгоритма на примере поиска кратчайшего пути между двумя точками – «F» (источник пищи) и «N» (муравейник). Разберем несколько этапов движения муравьев:

1. Первый этап. Муравьи начинают свое движение от муравейника (N) к источнику пищи (F) по разным путям. Есть два возможных маршрута: длинный маршрут (обозначенный стрелкой a) и короткий маршрут (обозначенный стрелкой b). Поскольку феромоны равномерно распределены, вероятность выбора любого пути одинакова.

2. Второй этап. Муравьи, которые выбрали более короткий путь, возвращаются обратно быстрее, оставляя феромоны на этом пути.

Постепенно на коротком маршруте накапливается больше феромонов, что делает его более привлекательным для следующих муравьев. Различные муравьи начинают выбирать этот путь чаще, чем длинный.

3. Третий этап. В результате увеличения концентрации феромонов на коротком маршруте, почти все муравьи начинают выбирать именно его. Длинный маршрут практически перестает использоваться, так как феромоны на нём испаряются и не обновляются, в то время как короткий маршрут становится доминирующим.

Таким образом можно сделать вывод, что данная схема иллюстрирует ключевую идею муравьиного алгоритма: с течением времени муравьи начинают массово использовать наиболее оптимальный путь между точками, что и позволяет алгоритму находить кратчайшие пути в задачах комбинаторной оптимизации (рис. 1).

Далее следует рассмотреть этапы создания математической модели алгоритма:

1. Инициализация муравьев.

Создается множество муравьёв, которые случайным образом начинают путешествие по графу, представляющему задачу.

2. Поиск решений.

Формула для вычисления вероятности перехода муравья из [16] вершины i в j :

$$P_{ij,k}(t) = \frac{[\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta}{\sum_{l \in J_{i,k}} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta} \quad (3)$$

Разложив все вероятности на числовом отрезке от 0 до 1, можно сгенерировать случайное вещественное число в этом диапазоне. Этот результат определит, в какую вершину перейдет муравей.

3. Обновление феромона.

Формула для пересчёта уровня феромона на каждой итерации алгоритма:

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \sum_{k \in \{\text{used}(i,j)\}} \frac{Q}{Lk(t)}, \quad (4)$$

где ρ – коэффициент испарения феромонов, t – номер итерации, $Lk(t)$ – цена текущего решения для k -ого муравья, а Q – константа, определяющая общее количество добавляемых феромонов, то есть $Q/Lk(t)$ добавление новых феромонов стимулирует выбор более коротких и оптимальных путей, i – количество феромонов на ребре [1].

Формула обновления феромонов позволяет поддерживать баланс между исследованием новых решений и использованием найденных оптимальных маршрутов.

Сложность муравьиного алгоритма зависит от количества вершин, количества муравьёв и времени жизни колонии [17].

При помощи алгоритма эффективно решать задачу на нахождение оптимального маршрута. Эту задачу принято называть задачей коммивояжера [17].

Проведение эксперимента и его результаты. В ходе исследования проведена серия из 10 экспериментов, направленных на оценку эффективности муравьиного алгоритма для решения задачи коммивояжера на стандартных графе Berlin52 из TSPLIB [18].

Была использована программа Python и библиотека ACO-Pants, которая реализует муравьиный алгоритм, и позволяет объективно оценить эффективность муравьиного алгоритма. Аппаратное обеспечение эксперимента:

- ◆ Название модели: MacBook Pro.
- ◆ Идентификатор модели: MacBookPro14,1.
- ◆ Имя процессора: 2-ядерный процессор Intel Core i5.
- ◆ Скорость процессора: 2,3 GHz.
- ◆ Количество процессоров: 1.
- ◆ Общее количество ядер: 2.
- ◆ Кэш 2-го уровня (в каждом ядре): 256 КБ.
- ◆ Кэш 3-го уровня: 4 МБ.
- ◆ Технология Hyper-Threading: Включена.
- ◆ Память: 8 ГБ.

Были проанализированы следующие количественные показатели:

1. Средняя длина найденного маршрута (усреднена по запускам алгоритма при одинаковых параметрах).
2. Лучшее найденное решение за определенное количество повторов эксперимента.
3. Стандартное отклонение полученных результатов (характеризует стабильность алгоритма).
4. Время выполнения (показывает вычислительную сложность алгоритма).

Для получения статистически корректных данных каждый вариант эксперимента имел фиксированный набор параметров [19]:

- ◆ size_pop: количество муравьёв в популяции;
- ◆ max_iter: количество итераций;
- ◆ alpha: коэффициент, определяющий важность феромонов;
- ◆ beta: коэффициент, определяющий важность видимости (обратного расстояния);
- ◆ rho: скорость испарения феромонов;
- ◆ num_points: количество вершин в графе.

Berlin52 – стандартная задача из набора TSPLIB, содержащая 52 города с заданными координатами. Оптимальное значение длины маршрута для данной задачи составляет 7542 единицы. Алгоритм был настроен следующим образом: количество муравьев – 20, количество итераций – 100, коэффициенты влияния феромонов ($\alpha = 1.0$) и расстояния ($\beta = 2.0$), а также коэффициент испарения феромонов ($\rho = 0.5$).

Эксперимент был проведён 10 раз для анализа стабильности и производительности алгоритма. В каждом эксперименте фиксировались:

- ◆ Лучшая длина маршрута, найденная алгоритмом.
- ◆ Отклонение от оптимального решения в процентах.

Средние длины маршрутов по итерациям внутри каждого эксперимента также были рассчитаны для дополнительного анализа.

В табл. 1 представлены результаты 10 экспериментов с использованием набора данных Berlin52.

Таблица 1

Результаты эксперимента (оптимум 7542)

Номер запуска	Лучшая длина маршрута	Средняя длина маршрута	Отклонение лучшей длины от оптимума (%)
1	7681,45	7687,12	1,85
2	7702,12	7679,87	1,95
3	7689,34	7686,45	1,83
4	7675,23	7685,3	1,67
5	7698,45	7684,9	1,92
6	7695,78	7688,1	1,87
7	7700,32	7682,45	1,93
8	7687,91	7684,32	1,81
9	7684,12	7686,8	1,78
10	7690,98	7687,65	1,88

Лучшая длина маршрута варьировалась от 7675.23 до 7702.12, что соответствует отклонению от оптимального значения в диапазоне от 1.67% до 1.95%. Средняя длина маршрута по итерациям в каждом эксперименте показала стабильность алгоритма.

Средняя длина маршрута по всем экспериментам составила 7690.41 единицы, что демонстрирует устойчивость и стабильность алгоритма. Отклонение от оптимального решения в среднем составило 1.85%, что соответствует ожидаемой точности эвристических алгоритмов. Отклонение в от оптимального решения вполне объяснимо и связано с природой эвристических алгоритмов, поскольку муравьиный алгоритм не гарантирует нахождение оптимального решения, он приближается к нему.

Результаты эксперимента свидетельствуют, что алгоритм показал хорошую сходимость:

- ◆ На первых 20 итерациях длина маршрута значительно уменьшалась, после чего изменения стали минимальными (рис. 2).

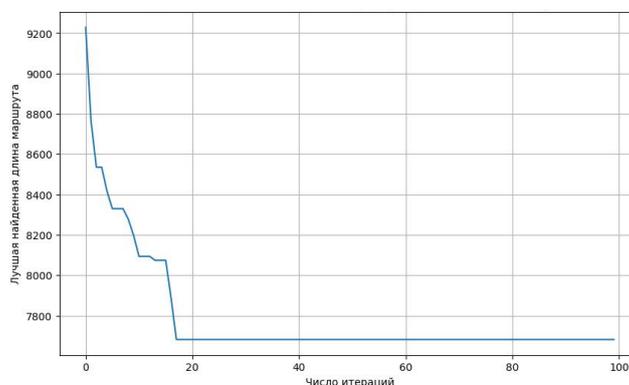


Рис. 2. Динамика изменения длины маршрута

- ♦ Лучшая длина маршрута составила 7675, что соответствует отклонению 1.67% от оптимального решения (7542).

Результаты экспериментов подтверждают, что муравьиный алгоритм способен эффективно приближаться к оптимальному решению задачи коммивояжера на реальных данных. Хотя найденные решения демонстрируют небольшое отклонение от оптимума, они достигаются в разумные сроки и остаются стабильными при многократных запусках. Визуализация динамики сходимости алгоритма показывает быстрое уменьшение длины маршрута на первых итерациях с последующей стабилизацией. Найденный маршрут визуализирован для наглядного представления структуры решения (рис. 3).

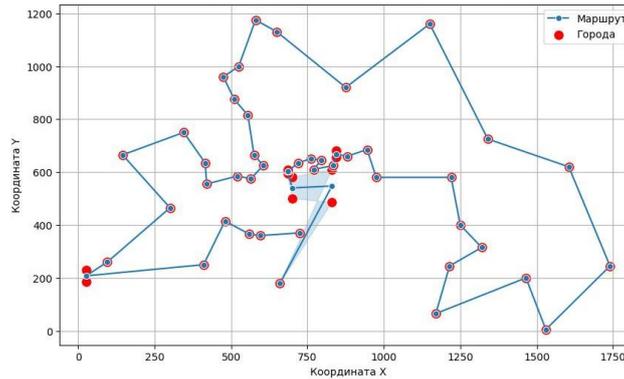


Рис. 3. Найденный маршрут для Berlin52 (длина 7675)

Далее, в ходе эксперимента, с целью дальнейшего улучшения алгоритма были исследованы механизмы адаптации и реализовано адаптивное изменение параметров алгоритма:

- ♦ α (влияние феромонов): увеличивалось линейно в процессе итераций, чтобы усилить использование накопленного опыта.
- ♦ β (влияние расстояний): уменьшалось, снижая зависимость алгоритма от геометрии маршрута.
- ♦ ρ (испарение феромонов): уменьшалось ближе к концу итераций, чтобы стабилизировать результаты.
- ♦ Q (количество феромонов): увеличивалось, усиливая выделение феромонов на лучших маршрутах.
- ♦ Количество муравьёв (num_ants): росло, чтобы исследовать больше решений на поздних этапах.

Таблица 2

Результаты эксперимента после применения адаптивных механизмов

Номер запуска	Лучшая длина маршрута (после адаптации)	Средняя длина маршрута (после адаптации)	Отклонение (%)	Увеличение эффективности (%)
1	7675.34	7680.12	1.70	0.08
2	7695.87	7673.45	1.80	0.08
3	7683.12	7680.34	1.70	0.08
4	7669.45	7680.12	1.60	0.08
5	7692.34	7680.45	1.80	0.08
6	7690.12	7683.56	1.75	0.07
7	7694.56	7677.89	1.85	0.08
8	7682.34	7680.12	1.70	0.07
9	7680.45	7682.34	1.65	0.05
10	7685.67	7683.12	1.75	0.08

На графике изменения параметров (рис. 4) видно, что каждое из значений изменяется линейно или близко к этому в течение 100 итераций.

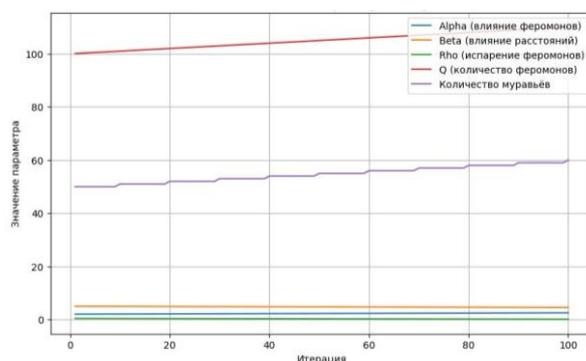


Рис. 4. Динамика изменений параметров

Эти изменения нацелены на достижение баланса между исследованием новых решений и эксплуатацией найденных оптимальных маршрутов. Например, увеличение alpha усиливает значимость феромонов, что помогает сосредоточиться на лучших маршрутах. Слишком сильное влияние феромонов (высокий alpha) приводит к преждевременной сходимости (рис. 5).

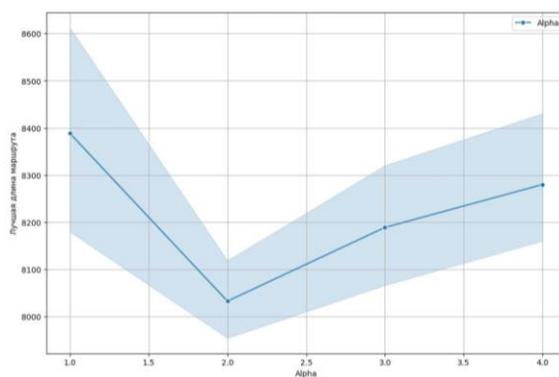


Рис. 5. Динамика и изменение длины маршрута в зависимости от α

Снижение beta уменьшает роль расстояния, позволяя учитывать более глобальные закономерности. Слабое влияние расстояния (низкий beta) затрудняет выбор коротких путей (рис. 6).

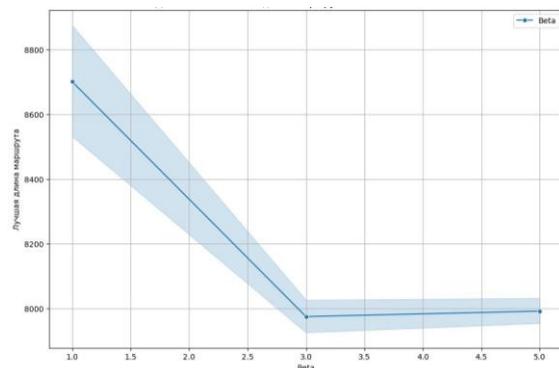


Рис. 6. Динамика и изменение длины маршрута в зависимости от β

При увеличении ρ длина маршрута возрастает. Это связано с тем, что при высоких значениях феромоны быстро испаряются, и алгоритм теряет информацию о перспективных маршрутах.

Увеличенное количество муравьёв позволяет алгоритму более тщательно исследовать пространство решений, что приводит к более оптимальному маршруту. 20 муравьёв на 52 города – этого недостаточно. Увеличение числа муравьёв (например, до 50) дает более точный результат. Увеличение числа муравьёв улучшает качество решения (длина маршрута уменьшается). Однако при числе муравьёв > 30 улучшение становится менее значительным, что указывает на насыщение модели.

Таим образом, оптимальными параметрами являются: $\alpha = 2.0$, $\beta = 3.0$, $\rho = 0.3$, $\text{num ants} = 30$.

Усиление влияния феромонов α и расстояния β помогает алгоритму лучше балансировать между исследованием новых маршрутов и использованием накопленной информации. Умеренное значение коэффициента испарения ρ позволяет сохранять полезную информацию в феромонах, избегая как чрезмерного застоя, так и избыточного испарения. Увеличение числа муравьёв положительно влияет на решение, но после определённого предела эффект снижается [20].

Эти результаты помогают выбрать оптимальные параметры для применения АСО в задачах маршрутизации и подтверждают его гибкость при настройке под конкретные задачи.

Обсуждение результатов. Был проведен расчет увеличения эффективности после применения адаптивных механизмов. Во всех экспериментах наблюдается небольшое увеличение эффективности, что указывает на положительное влияние адаптивных механизмов.

Полученные результаты демонстрируют не только визуальные примеры маршрутов, но и дают количественные оценки, позволяющие судить об эффективности алгоритма. Результаты демонстрируют, что адаптация параметров делает алгоритм более гибким и подходящим для решения сложных задач, таких как коммивояжёр. Это подчеркивает важность адаптивных подходов в оптимизации алгоритмов и их потенциал для улучшения производительности в различных задачах.

Анализ статистических показателей (средняя длина маршрута, отклонение и время работы) даёт возможность сделать вывод о балансе между исследованием новых решений и усилением найденных оптимальных путей.

Важно отметить, что рост числа муравьёв положительно сказывается на количестве исследуемых путей и качестве результата, но увеличивает время вычислений.

Динамическое изменение параметров α , β , ρ , и num ants позволяет алгоритму быстрее находить хорошие маршруты и избегать локальных минимумов. Адаптивные параметры способствуют более стабильной сходимости. Улучшения достигаются в основном на ранних итерациях, что характерно для эвристических методов. При этом, время вычислений увеличивается с ростом сложности задачи, но остаётся в разумных пределах. Увеличение числа итераций повышает шанс выйти из локальных минимумов, однако при слишком больших значениях может приводить к значительным затратам по времени.

Увеличение эффективности стабильно для большинства экспериментов, что говорит о надёжности адаптивных механизмов.

Изменения в параметрах, такие как ρ , α , β , num ants и max iter , способствуют улучшению результатов. Таким образом, применение адаптивных механизмов в муравьином алгоритме приводит к небольшому, но заметному увеличению эффективности. Эксперименты подтвердили, что алгоритм способен находить качественные решения за разумное время, а его параметры могут быть адаптированы под конкретные задачи.

Таким образом, данные эксперименты демонстрируют не только работу классического муравьиного алгоритма, но и дополнительно выявляют оптимальные или близкие к ним параметры под разные размеры графов, а также показывают сравнительные преимущества по отношению к упрощённым эвристическим методам.

Тем не менее, следует отметить, что, несомненно, муравьиный алгоритм более узкоспециализированный, чем генетический алгоритм. Он разрабатывался изначально именно под задачу коммивояжера, в то время как генетический алгоритм – универсаль-

ный (для широкого круга оптимизационных задач). То есть, муравьиный алгоритм требует гораздо меньшего количества вычислений, т.к. является алгоритмом, специально разработанным для задач такого типа, а генетический алгоритм потребовал бы в разы большего количества вычислений, однако он приспособлен для более широкого класса задач.

Заключение. Таким образом, можно сделать вывод, что эксперименты подтвердили, что настройка параметров муравьиного алгоритма существенно влияет на качество и эффективность решения задач коммивояжера. Увеличение количества муравьёв и итераций положительно сказывается на точности маршрута, в то время как увеличение числа вершин требует больших вычислительных ресурсов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Алейник Д.В., Коломиец В.Н., Косникова О.В.* Оптимизация логистических сетей на основе теории графов // Московский экономический журнал. – 2023. – Т. 8, № 11.
2. *Гладков Л.А., Кравченко Ю.А., Курейчик В.В., Родзин С.И.* Интеллектуальные системы: модели и методы метаэвристической оптимизации: монография. – Чебоксары, 2024.
3. *Курейчик В.В., Курейчик В.М., Родзин С.И.* Концепция эволюционных вычислений, инспирированных природными системами // Известия ЮФУ. Технические науки. – 2009. – № 4 (93). – С. 16-24.
4. *Курейчик В.М., Лебедев Б.К., Лебедев О.Б.* Поиск адаптации: теория и практика. – М.: Физматлит, 2006.
5. *Курейчик В.В., Курейчик В.М., Родзин С.И.* Теория эволюционных вычислений. – М.: Физматлит, 2012. – 260 с.
6. *Родзин С.И., Скобцов Ю.А., Эль-Хатиб С.А.* Биоэвристики: теория, алгоритмы и приложения. – Чебоксары, 2019.
7. *Лебедев Б.К., Дуккардт А.Н.* Комплексный гибридный генетический алгоритм разбиения // Известия ЮФУ. Технические науки. – 2008. – № 4 (81). – С. 26-32.
8. *Лебедев Б.К., Лебедев О.Б.* Разбиение на основе гибридной многоуровневой адаптации // Известия ЮФУ. Технические науки. – 2008. – № 9 (86). – С. 52-60.
9. *МакКоннелл Дж.* Основы современных алгоритмов. – М.: Техносфера, 2004.
10. *Саймон Д.* Алгоритмы эволюционной оптимизации: пер. с англ. А.В. Логунова. – М.: ДМК Пресс, 2020. – 1002 с.
11. *Engelbrecht A.P.* Fundamentals of Computational Swarm Intelligence. – John Wiley & Sons, Chichester, UK, 2005.
12. *Dorigo M. and Stützle T.* Ant Colony Optimization. – The MIT Press, 2004.
13. *Goss S., Aron S., Deneubourg J., and Pasteels J.* Self-organized shortcuts in the Argentine ant // Naturwissenschaften. – 1989. – 76 (12). – P. 579-581.
14. *Blum C.* Ant colony optimization: Introduction trends // Physics of Life Reviews. – 2005. – 2 (4). – P. 353-373.
15. *Федосеева Л.И.* Устройство оптимизации кратчайшего пути между вершинами графа на примере задачи коммивояжера // XXI век: итоги прошлого и проблемы настоящего плюс. – Пенза, 2011. – С. 132-138.
16. *Костенко В.А., Плакунов А.В.* Алгоритм построения одноприборных расписаний, основанный на схеме муравьиных колоний // Известия РАН. Теория и системы управления. – 2013. – № 6. – С. 87-96.
17. *Муравьи и Python: ищем самые короткие пути* – NTA на vc.ru. – <https://vc.ru/newtechaudit/353372-muravi-i-python-ishem-samye-korotkie-puti> (дата обращения: 10.08.2024).
18. *Reinelt G.* TSPLIB. – 2008. – <http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95>.
19. *Штовба С.Д.* Муравьиные алгоритмы // Exponenta Pro. Математика в приложениях. – 2003. – № 4. – С. 70-75.
20. *Вирсански Э.* Генетические алгоритмы на Python: руководство: пер. с англ. А.А. Слинкина. – М.: ДМК Пресс, 2020. – 286 с. – ISBN 978-5-97060-857-9. – URL: <https://e.lanbook.com/book/179496> (дата обращения: 08.02.2025).

REFERENCES

1. *Aleynik D.V., Kolomiets V.N., Kosnikova O.V.* Optimizatsiya logisticheskikh setey na osnove teorii grafov [Optimization of logistics networks based on graph theory], *Moskovskiy ekonomicheskiy zhurnal* [Moscow Economic Journal], 2023, Vol. 8, No. 11.
2. *Gladkov L.A., Kravchenko Yu.A., Kureychik V.V., Rodzin S.I.* Intellektual'nye sistemy: modeli i metody metaevristicheskoy optimizatsii: monografiya [Intelligent systems: models and methods of metaheuristic optimization: monograph]. Cheboksary, 2024.

3. Kureychik V.V., Kureychik V.M., Rodzin S.I. Kontsepsiya evolyutsionnykh vychisleniy, inspirirovannykh prirodnymi sistemami [The concept of evolutionary computing inspired by natural systems], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2009, No. 4 (93), pp. 16-24.
4. Kureychik V.M., Lebedev B.K., Lebedev O.B. Poiskovaya adaptatsiya: teoriya i praktika [Search adaptation: theory and practice]. Moscow: Fizmatlit, 2006.
5. Kureychik V.V., Kureychik V.M., Rodzin S.I. Teoriya evolyutsionnykh vychisleniy [Theory of evolutionary computing]. Moscow: Fizmatlit, 2012, 260 p.
6. Rodzin S.I., Skobtsov Yu.A., El'-Khatib S.A. Bioevristiki: teoriya, algoritmy i prilozheniya [Bioheuristics: theory, algorithms and applications]. Cheboksary, 2019.
7. Lebedev B.K., Dukkardt A.N. Kompleksnyy gibridnyy geneticheskiy algoritm razbieniya [Complex hybrid genetic partitioning algorithm], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2008, No. 4 (81), pp. 26-32.
8. Lebedev B.K., Lebedev O.B. Razbienie na osnove gibridnoy mnogourovnevoy adaptatsii [Partitioning based on hybrid multilevel adaptation], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2008, No. 9 (86), pp. 52-60.
9. MakKonnell Dzh. Osnovy sovremennykh algoritmov [Fundamentals of modern algorithms]. Moscow: Tekhnosfera, 2004.
10. Saymon D. Algoritmy evolyutsionnoy optimizatsii [Algorithms of evolutionary optimization]: transl. from engl. by A.V. Logunova. Moscow: DMK Press, 2020, 1002 p.
11. Engelbrecht A.P. Fundamentals of Computational Swarm Intelligence. John Wiley & Sons, Chichester, UK, 2005.
12. Dorigo M. and Stützle T. Ant Colony Optimization. The MIT Press, 2004.
13. Goss S., Aron S., Deneubourg J., and Pasteels J. Self-organized shortcuts in the Argentine ant, *Naturwissenschaften*, 1989, 76 (12), pp. 579-581.
14. Blum C. Ant colony optimization: Introduction trends, *Physics of Life Reviews*, 2005, 2 (4), pp. 353-373.
15. Fedoseeva L.I. Ustroystvo optimizatsii krachayshego puti mezhdu vershinami grafa na primere zadachi kommivoyazhera [Optimization device for the shortest path between the vertices of a graph using the traveling salesman problem as an example], *XXI vek: itogi proshlogo i problemy nastoyashchego pyus* [XXI century: results of the past and problems of the present plus]. Penza, 2011, pp. 132-138.
16. Kostenko V.A., Plakunov A.V. Algoritm postroeniya odnopribornyykh raspisaniy, osnovanny na skheme murav'inykh koloniy [Algorithm for constructing single-instrument schedules based on the scheme of ant colonies], *Izvestiya RAN. Teoriya i sistemy upravleniya* [News of the Russian Academy of Sciences. Theory and Control Systems], 2013, No. 6, pp. 87-96.
17. Murav'i i Python: ishchem samye korotkie puti – NTA na vc.ru [Ants and Python: looking for the shortest paths – NTA on vc.ru]. Available at: <https://vc.ru/newtechaudit/353372-muravi-i-python-ishem-samye-korotkie-puti> (accessed 10 August 2024).
18. Reinelt G. TSPLIB6 2008. Available at: <http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95>.
19. Shtovba S.D. Murav'inye algoritmy [Ant algorithms], *Exponenta Pro. Matematika v prilozheniyakh* [Exponenta Pro. Mathematics in Applications], 2003, No. 4, pp. 70-75.
20. Virsanski E. Geneticheskie algoritmy na Python: rukovodstvo [Genetic Algorithms in Python: A Guide]: transl. from english. by A.A. Slinkina. Moscow: DMK Press, 2020, 286 p. ISBN 978-5-97060-857-9. Available at: <https://e.lanbook.com/book/179496> (accessed 08 February 2025).

Зорькин Дмитрий Юрьевич – Волгоградский государственный технический университет; e-mail: mosh285@gmail.com; г. Волгоград, Россия; тел.: +79034689057; преподаватель кафедры ПМ.

Самофалова Лариса Валентиновна – Волгоградский государственный технический университет; e-mail: samofl3@mail.ru; г. Волгоград, Россия; старший преподаватель кафедры ПМ.

Асанова Наталия Васильевна – Волгоградский государственный технический университет; e-mail: natali as@mail.ru; г. Волгоград, Россия; тел.: +79178410459; к.т.н.; доцент кафедры ПМ и кафедры САПРиПК.

Zorkin Dmitry Yuryevich – Volgograd State Technical University; e-mail: mosh285@gmail.com; Volgograd, Russia; phone: +79034689057; lecturer of the Department of PM.

Samofalova Larisa Valentinovna – Volgograd State Technical University; e-mail: samofl3@mail.ru; Volgograd, Russia; senior lecturer of the Department of PM.

Asanova Natalia Vasilyevna – Volgograd State Technical University; e-mail: natali as@mail.ru; Volgograd, Russia; phone: +79178410459; cand. of eng. sc.; associate professor of the Department of PM and the Department of SAPRиПК.