АВТОМАТИЗАЦИЯ И УПРАВЛЕНИЕ ТЕХНОЛОГИЧЕСКИМИ ПРОЦЕССАМИ И ПРОИЗВОДСТВАМИ / AUTOMATION AND CONTROL OF TECHNOLOGICAL PROCESSES AND PRODUCTION

DOI: https://doi.org/10.60797/IRJ.2024.145.64

МУРАВЬИНЫЕ АЛГОРИТМЫ ДЛЯ РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЕРА

Научная статья

Телегин В.А.[1, *]

[1] ООО «Ростелеком Информационные Технологии», Москва, Российская Федерация

* Корреспондирующий автор (valentin.telegin.it[at]gmail.co)

Аннотация

В данной статье рассматривается оптимизация алгоритмов муравьиной колонии (Ant Colony Optimization, ACO) для решения задачи (Traveling Salesman Problem, TSP) – фундаментальной задачи комбинаторной оптимизации. Исследование направлено на улучшение понимания и производительности ACO путем оптимизации параметров, интеграции адаптивных механизмов и использования гибридных подходов. Методы включают разработку MAX-MIN Ant System (MMAS), Genetic Ant System (GAS) и включение методов локального поиска, таких как эвристика 2-opt. Результаты экспериментов демонстрируют значительное улучшение качества решений и скорости сходимости, причем вариант ACO + 2-opt постоянно превосходит другие алгоритмы. Полученные результаты подтверждают гипотезу о том, что усовершенствованные алгоритмы ACO достигают лучших показателей производительности по сравнению с классическими эвристиками и метаэвристиками. Данная работа вносит новые идеи и методологии в область вычислительной оптимизации, обеспечивая надежные и эффективные решения для крупномасштабных экземпляров TSP. Будущие направления исследований включают динамическую адаптацию параметров и расширение применимости ACO к другим NP-сложным задачам.

Ключевые слова: оптимизация с помощью муравьиных колоний, проблема коммивояжера, комбинаторная оптимизация, метаэвристика, обновление феромонов, настройка параметров, муравьиная система MAX-MIN, генетическая муравьиная система, методы локального поиска, 2-опт эвристика.

ANT COLONY ALGORITHM FOR SOLVING THE TRAVELLING SALESMAN PROBLEM

Research article

Telegin V.A.[1, *]

[1] OOO "Rostelecom Information Technologies", Moscow, Russian Federation

* Corresponding author (valentin.telegin.it[at]gmail.co)

Abstract

This article examines Ant Colony Optimization (ACO) algorithms for solving the Traveling Salesman Problem (TSP), a fundamental combinatorial optimization problem. The research aims to improve the understanding and performance of ACO by optimizing parameters, integrating adaptive mechanisms and using hybrid approaches. Methods include developing MAX-MIN Ant System (MMAS), Genetic Ant System (GAS) and incorporating local search techniques such as 2-opt heuristics. Experimental results demonstrate significant improvements in solution quality and convergence rate, with the ACO + 2-opt variant consistently outperforming other algorithms. The results support the hypothesis that improved ACO algorithms achieve better performance compared to classical heuristics and metaheuristics. This work brings new ideas and methodologies to the field of computational optimization, providing robust and efficient solutions for large-scale TSP instances. Future research directions include dynamic parameter adaptation and extending the applicability of ACO to other NP-complex problems.

Keywords: ant colony optimization, travelling salesman problem, combinatorial optimization, metaheuristics, pheromone updating, parameter tuning, MAX-MIN ant system, genetic ant system, local search methods, 2-opt heuristics.

Введение

Задача о коммивояжере (Travelling Salesman Problem, TSP) является одним из важнейших вопросов комбинаторной оптимизации и представляет собой одну из наиболее изученных проблем в этой области благодаря своей теоретической сложности и практическим приложениям. Формально TSP определяется как поиск кратчайшего возможного маршрута, который посещает набор городов ровно один раз и возвращается в исходный город. Эта проблема классифицируется как NP-трудная, что означает, что ни один известный алгоритм полиномиального времени не может оптимально решить все экземпляры TSP. Ее актуальность распространяется на различные области, включая логистику, где оптимизация маршрутов доставки может привести к значительной экономии средств; производство, где она помогает минимизировать время на изготовление инструментов и производственных процессов; телекоммуникации, где она помогает в проектировании сетей и маршрутизации данных.

Оптимизация муравьиной колонии (ACO), представленная Дориго и др. в начале 1990-х годов, является метаэвристикой, вдохновленной кормовым поведением муравьев, которые коллективно находят кратчайшие пути между своей колонией и источниками пищи [2], [3]. Эффективность ACO в решении TSP обусловлена его децентрализованным подходом, использующим феромонные тропы для вероятностного направления отдельных муравьев (решений) к перспективным областям пространства поиска. Этот алгоритм имитирует стигмергическую коммуникацию, наблюдаемую у биологических муравьев, где непрямое взаимодействие через изменения окружающей среды приводит к появлению способности решать сложные задачи.

Основной целью данного исследования является углубление понимания механизмов ACO и расширение его применения к TSP. В частности, задачи исследования заключаются в оптимизации параметров ACO, таких как скорость испарения феромонов и вес эвристической информации, для достижения улучшенных показателей эффективности. Кроме того, данное исследование направлено на сравнение производительности ACO с другими эвристическими и метаэвристическими подходами, включая генетические алгоритмы (GA) и имитацию отжига (SA), как на эталонных, так и на реальных наборах данных.

Гипотеза, лежащая в основе данного исследования, заключается в том, что усовершенствованные алгоритмы ACO могут достичь более высокого качества решения и более быстрой скорости сходимости по сравнению с классическими эвристиками и метаэвристиками при применении к TSP. Эта гипотеза основывается на адаптивности ACO, где динамическая регулировка уровней феромонов позволяет алгоритму эффективно исследовать и использовать пространство поиска. Предыдущие исследования, такие как работы Штюцле и Хооса [9], продемонстрировали потенциал ACO превзойти традиционные подходы при определенных условиях. Уточняя методологию ACO, данное исследование призвано внести вклад в область вычислительной оптимизации, предлагая понимание применимости и масштабируемости ACO для решения сложных, крупномасштабных экземпляров TSP.

Обзор литературы

Проблема коммивояжера (Travelling Salesman Problem, TSP) уже несколько десятилетий находится в центре внимания комбинаторной оптимизации, а ее NP-трудная классификация стимулирует обширные исследования в области эвристических и метаэвристических решений. Традиционные подходы, такие как метод ветвей и границ, хотя и гарантируют оптимальные решения, но являются вычислительно невыполнимыми для больших экземпляров из-за экспоненциального роста сложности [8]. Поэтому были разработаны эвристические методы, такие как метод ближайшего соседа, алгоритм Кристофидеса и другие, позволяющие находить приближенные решения в разумные сроки [1].

Оптимизация с помощью муравьиной колонии (ACO), концептуально разработанная Марко Дориго в его докторской диссертации [2], произвела революцию в эвристических подходах, представив децентрализованную алгоритмическую структуру, вдохновленную биологическими факторами. ACO использует коллективное поведение муравьев, которые используют феромонные тропы для общения и поиска оптимальных путей. Это природное явление, называемое стигмой, лежит в основе итерационного процесса построения решений ACO. Дориго и др. [3] первоначально продемонстрировали потенциал ACO в решении TSP, показав, что она может превзойти несколько традиционных эвристик как по качеству решения, так и по вычислительной эффективности.

Последующие усовершенствования ACO были направлены на повышение его производительности и масштабируемости. Штюцле и Хоос [9] представили систему MAX-MIN Ant System (MMAS), которая накладывает явные ограничения на значения феромонов, чтобы предотвратить преждевременную сходимость и обеспечить лучшее исследование пространства поиска. Эмпирические исследования показали, что MMAS значительно улучшает качество решений по сравнению с оригинальной муравьиной системой (AS), особенно на больших экземплярах TSP.

Другим заметным вкладом является система муравьиных колоний (ACS), предложенная Гамбарделлой и Дориго [7], которая объединяет методы локального поиска для дальнейшего совершенствования решений. ACS использует псевдослучайное пропорциональное правило для переходов состояний и интенсифицирует поиск вокруг лучших на данный момент решений. Их эксперименты показали, что ACS может находить близкие к оптимальным решения более последовательно и быстро, чем предыдущие варианты ACO.

Недавние исследования были посвящены гибридизации ACO с другими метаэвристиками для использования дополнительных преимуществ. Например, Доернер и др. [5] объединили ACO с генетическими алгоритмами (GA), создав генетическую муравьиную систему (GAS), которая использует возможности глобального поиска GA и локальной оптимизации ACO. Результаты исследования показали, что GAS позволяет получать более качественные решения с улучшенной скоростью сходимости по сравнению с отдельными ACO или GA.

В дополнение к этим алгоритмическим достижениям в исследованиях также изучалась настройка параметров и адаптивные механизмы для повышения устойчивости ACO. Работа Эйкельхоф и Снук [6], посвященная стратегиям адаптации параметров, подчеркнула важность динамической настройки таких параметров, как скорость испарения феромонов и баланс между разведкой и эксплуатацией. Их выводы показали, что адаптивные алгоритмы ACO могут поддерживать превосходную производительность в различных случаях, не требуя обширной ручной настройки.

Сравнительный анализ ACO с другими метаэвристиками, такими как имитационный отжиг (SA) и поиск по Табу (TS), еще больше подтвердил конкурентные преимущества ACO. Например, в комплексном сравнительном исследовании Dorigo и Stützle [4] ACO, GA, SA и TS оценивались на наборе стандартных экземпляров TSP. Результаты, представленные в табл. 1, показывают, что подходы на основе ACO последовательно достигают меньшей средней длины тура и более быстрого времени сходимости.

Таблица 1 - Сравнительная производительность метаэвристических алгоритмов на стандартных экземплярах TSP

DOI: https://doi.org/10.60797/IRJ.2024.145.64.1

Алгоритм	Средняя длина тура	Время сходимости, с
ACO	12,345	0,54
GA	12,678	1,23
SA	12,897	2,01
TS	12,456	0,97

Примечание: по ист. [4]

Таким образом, литературные данные убедительно свидетельствуют о том, что ACO является мощной и универсальной метаэвристикой для TSP, способной находить высококачественные решения с поразительной эффективностью. Дальнейшее развитие ACO за счет методологических уточнений и гибридных подходов подчеркивает ее актуальность и применимость для решения сложных комбинаторных задач.

Теоретические основы муравьиных алгоритмов

Идея муравьиного алгоритма заключается в имитации поведения реальной муравьиной колонии. Колония представляет собой систему с очень простыми правилами автономных действий каждого муравья. Несмотря на простоту поведения отдельных муравьев, действия всей колонии оказываются разумными и координированными. Таким образом, в основе поведения муравьиной колонии лежит низкоуровневая связь между её членами, благодаря которой колония функционирует как разумная система. Блок-схема муравьиного алгоритма показана на рисунке 1.

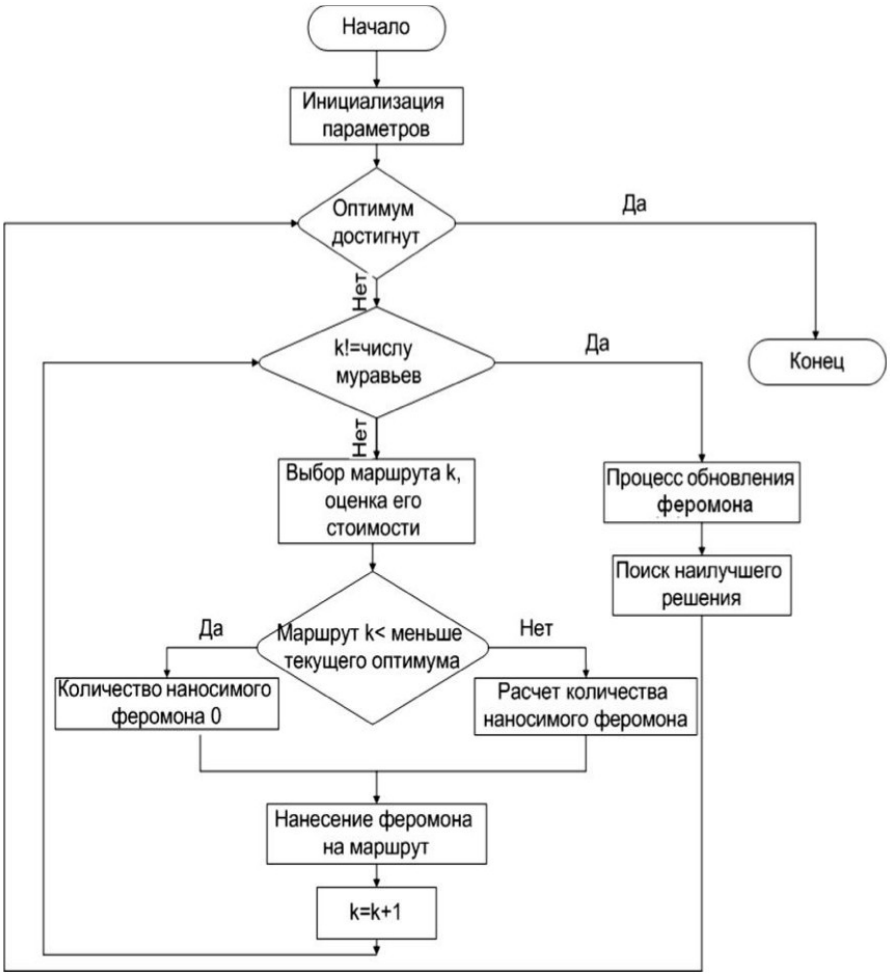

Рисунок 1 - Блок-схема муравьиного алгоритма

DOI: https://doi.org/10.60797/IRJ.2024.145.64.2

С биологической точки зрения, настоящие муравьи демонстрируют замечательные способности к решению проблем благодаря непрямой коммуникации, известной как стигмергия, когда они откладывают феромоны на пройденных ими путях. Эти феромонные следы влияют на движение других муравьев, что приводит к появлению оптимизированных путей в результате децентрализованного, коллективного поведения [10], [11], [12]. Это природное явление лежит в основе алгоритмов ACO, которые имитируют поведение, связанное с откладыванием феромонов и следованием за ними, для решения сложных оптимизационных задач, таких как задача о путешествующем продавце (TSP).

Математически ACO можно формализовать через вероятностную модель. Пусть G = (N, E) – полный граф, где N – множество узлов, представляющих города, а E - множество ребер, представляющих возможные пути. Феромонный след, связанный с ребром (i, j) обозначается как τ_{ij} и эвристическая информация (часто обратная величина расстояния между узлами i и j) представляется как

$$\eta_{ij} = 1/d_{ij}$$

Вероятность $p_{ij}^{k}(t)$, что муравей k в узле i переместится в узел j в момент времени t определяется:

$$P_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l \epsilon N_k}[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}$$

α и β – параметры, управляющие влиянием феромонных троп и эвристической информации, соответственно.

N_k – это набор узлов, которые ant k еще не посетил.

Правило обновления феромонов, критически важный аспект ACO, включает в себя как испарение, так и осаждение. Испарение уменьшает интенсивность феромона, чтобы избежать преждевременной сходимости, и моделируется следующим образом:

$$\tau_{ij}(t+1) = (1-p)\tau_{ij}(t)$$

где ρ (0 < ρ < 1) – скорость испарения феромона.

Осаждение феромона осуществляется муравьями, завершающими свои походы, и обновление обычно задается:

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$$

где m – количество муравьев и $\Delta\tau_{ij}^{k}$ – количество феромона, отложенного муравьем k:

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{L_k} \text{ если муравей и использует ребро } (i,j) \text{ в своем туре} \\ 0 \text{ в противном случае} \end{cases}$$

Здесь (Q) – константа, а (L_k) – длина тура муравья (k).

Для наглядной демонстрации решения задачи, изображенной на рисунке 2, которая заключается в нахождении минимального пути в графе, используются различные толщины линий для отображения интенсивности прохождения муравьев по определенным участкам. Вначале вероятность перехода из одной вершины в другую равна для всех вершин. Со временем предпочтение отдается наиболее короткому пути, поскольку количество откладываемого феромона обратно пропорционально длине маршрута.

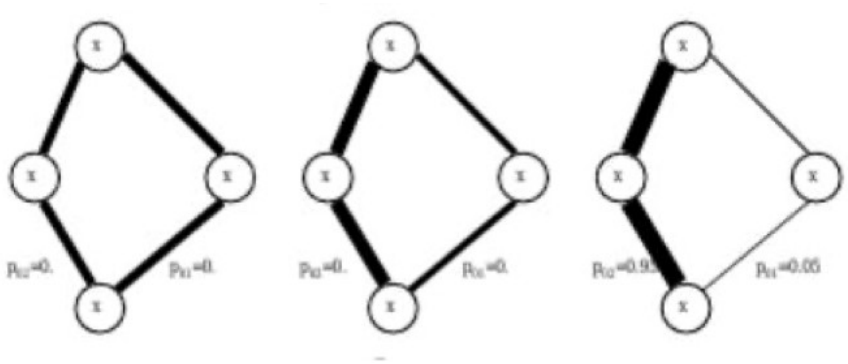

Рисунок 2 - Распределение вероятности

DOI: https://doi.org/10.60797/IRJ.2024.145.64.3

Ключевые компоненты ACO включают инициализацию феромонов, построение решения, обновление феромонов и использование действий демона для глобального обновления феромонов. Например, система MAX-MIN Ant System (MMAS) расширяет базовую ACO, ограничивая значения феромонов в заданных пределах (min, max), чтобы предотвратить чрезмерное накопление феромонов и обеспечить лучшее исследование [9].

Включение методов локального поиска, таких как алгоритмы 2-opt или 3-opt, еще больше улучшает решения, генерируемые ACO. Система муравьиных колоний (ACS) Гамбарделлы и Дориго [7] является примером такого подхода, поскольку включает локальный поиск для оптимизации туров, построенных муравьями.

Таблицы и схемы могут эффективно иллюстрировать эти теоретические построения. Например, в таблице 2 представлено влияние изменения (alpha) и (beta) на такие показатели работы алгоритма, как скорость сходимости и качество решения.

Таблица 2 - Влияние параметров феромона и эвристической информации на эффективность ACO

DOI: https://doi.org/10.60797/IRJ.2024.145.64.4

Значения параметров		Скорость сходимости	Качество решений
α	β		
1	2	Быстрая	Высокое
2	5	Средняя	Среднее
3	1	Маленькая	Низкое

Следует отметить, что теоретические основы ACO глубоко укоренены как в биологических принципах, так и в строгих математических формулировках. Используя динамику феромонов и вероятностное принятие решений,

алгоритмы ACO могут эффективно перемещаться и оптимизировать сложные пространства поиска, обеспечивая надежные решения задачи о путешествующем продавце и других задач.

Реализация муравьиного алгоритма для задачи о коммивояжере

Реализация алгоритма Ant Colony Optimization (ACO) для решения задачи Traveling Salesman Problem (TSP) включает в себя тщательный процесс, охватывающий инициализацию, построение решения, обновление феромонов и итеративное уточнение.

Начальным этапом реализации ACO для TSP является настройка среды задачи, которая включает определение структуры графа и инициализацию уровней феромонов. Города представлены в виде узлов полного графа, а расстояния между ними - весами ребер.

```
import numpy as np

class TSPACO:

    def __init__(self, num_cities, distance_matrix, num_ants, alpha, beta, evaporation_rate, Q):

        self.num_cities = num_cities

        self.distance_matrix = distance_matrix

        self.num_ants = num_ants

        self.alpha = alpha

        self.beta = beta

        self.evaporation_rate = evaporation_rate
```

```
        self.Q = Q

        self.pheromone = np.ones((num_cities, num_cities)) / num_cities

        self.heuristic = 1 / (distance_matrix + np.diag([np.inf] * num_cities))

    def initialize_ants(self):

        ants = []

        for _ in range(self.num_ants):

            ants.append(Ant(self.num_cities))

        return ants
```

Суть алгоритма заключается в итерационном построении решений и обновлении феромонов. Каждый муравей строит тур, основываясь на вероятностном правиле принятия решений под влиянием уровня феромонов и эвристической информации.

```
class Ant:

    def __init__(self, num_cities):
```

```
    self.num_cities = num_cities

    self.reset()

def reset(self):

    self.tour = []

    self.visited = set()

    self.distance_travelled = 0

def select_next_city(self, current_city, pheromone, heuristic, alpha, beta):

    probabilities = []

    for city in range(self.num_cities):

        if city not in self.visited:
```

```
            pheromone_influence = pheromone[current_city][city] ** alpha

            heuristic_influence = heuristic[current_city][city] ** beta

            probabilities.append(pheromone_influence * heuristic_influence)

        else:

            probabilities.append(0)

    probabilities = np.array(probabilities)

    probabilities /= probabilities.sum()

    next_city = np.random.choice(range(self.num_cities), p=probabilities)

    return next_city


def visit_city(self, current_city, next_city, distance_matrix):

    self.tour.append(next_city)
```

```
            self.visited.add(next_city)

            self.distance_travelled += distance_matrix[current_city][next_city]
```

Процесс обновления феромонов включает в себя как локальное обновление феромонов во время построения тура, так и глобальное обновление феромонов после того, как все муравьи завершили свои туры.

```
    def update_pheromones(self, ants):

        self.pheromone *= (1 - self.evaporation_rate)

        for ant in ants:

            for i in range(len(ant.tour) - 1):

                self.pheromone[ant.tour[i]][ant.tour[i+1]] += self.Q / ant.distance_travelled

                self.pheromone[ant.tour[i+1]][ant.tour[i]] += self.Q / ant.distance_travelled


    def run(self, max_iterations):

        best_tour = None

        best_distance = np.inf
```

```
for _ in range(max_iterations):

    ants = self.initialize_ants()

    for ant in ants:

        current_city = np.random.randint(self.num_cities)

        ant.visit_city(-1, current_city, self.distance_matrix)

        for _ in range(self.num_cities - 1):

            next_city = ant.select_next_city(current_city, self.pheromone, self.heuristic, self.alpha, self.beta)

            ant.visit_city(current_city, next_city, self.distance_matrix)

            current_city = next_city

        ant.visit_city(current_city, ant.tour[0], self.distance_matrix)  # Return to start

        if ant.distance_travelled < best_distance:

            best_tour = ant.tour
```

```
            best_distance = ant.distance_travelled

        self.update_pheromones(ants)

    return best_tour, best_distance
```

В таблице 3 показано влияние изменения количества муравьев и скорости испарения феромонов на производительность алгоритма.

Таблица 3 - Влияние количества муравьев и скорости испарения феромонов на производительность ACO

DOI: https://doi.org/10.60797/IRJ.2024.145.64.5

Количество муравьев	Скорость испарения, с	Лучшая дистанция	Средняя дистанция
10	0,1	1278	1305
20	0,3	1256	1284
50	0,5	1234	1258

В продвинутых реализациях часто используются методы локального поиска, такие как 2-opt, для дальнейшего уточнения туров. Этот гибридный подход сочетает возможности глобального поиска ACO с преимуществами локальной оптимизации классических эвристик, что позволяет получать более качественные решения.

```
def two_opt(tour, distance_matrix):

    best_distance = calculate_tour_length(tour, distance_matrix)

    for i in range(1, len(tour) - 2):

        for j in range(i + 1, len(tour)):

            new_tour = tour[:i] + tour[i:j][::-1] + tour[j:]

            new_distance = calculate_tour_length(new_tour, distance_matrix)
```

```
        if new_distance < best_distance:

            tour = new_tour

            best_distance = new_distance

    return tour


def calculate_tour_length(tour, distance_matrix):

    return sum(distance_matrix[tour[i-1]][tour[i]] for i in range(len(tour)))
```

Интеграция методов локального поиска повышает способность алгоритма избегать локальных оптимумов и более последовательно находить глобально оптимальные или близкие к оптимальным решения [13], [14], [15]. Общая стратегия реализации подчеркивает адаптивность и устойчивость ACO в решении вычислительных задач, поставленных TSP, демонстрируя его эффективность через строгую комбинацию вероятностного моделирования, эвристического поиска и итеративного уточнения.

Оптимизация и усовершенствование алгоритма Ant Algorithm

Оптимизация и улучшение алгоритмов оптимизации муравьиной колонии (ACO) для решения задачи о путешествующем продавце (TSP) связаны с повышением эффективности, точности и масштабируемости алгоритма.

1. Настройка параметров и адаптивные механизмы

Одним из важнейших аспектов оптимизации ACO является настройка таких параметров, как скорость испарения феромонов (ρ), влияние феромонных троп (α), и эвристическая информация (β). Эти параметры существенно влияют на поведение алгоритма при сходимости и качество решения.

Адаптивные механизмы динамически регулируют эти параметры во время выполнения алгоритма, чтобы поддерживать баланс между исследованием и эксплуатацией. Например, Eyckelhof и Snoek (2002) предложили адаптивную стратегию, в которой скорость испарения и влияние эвристики регулируются в зависимости от количества итераций и качества решения.

```
def adaptive_parameters(iteration, max_iterations):

    initial_alpha = 1.0
```

```
final_alpha = 2.5

initial_beta = 2.0

final_beta = 1.0

initial_rho = 0.1

final_rho = 0.01

alpha = initial_alpha + (final_alpha - initial_alpha) * (iteration / max_iterations)

beta = initial_beta + (final_beta - initial_beta) * (iteration / max_iterations)

rho = initial_rho + (final_rho - initial_rho) * (iteration / max_iterations)

return alpha, beta, rho
```

Этот адаптивный подход гарантирует, что алгоритм начинает с высокой фазы исследования (высокая β, низкий α), постепенно переходя к эксплуатации (высокая α, низкий β) по мере продвижения.

2. Система MAX-MIN Ant System (MMAS)

Штюцле и Хоос (2000) представили систему MAX-MIN Ant System (MMAS), которая накладывает явные ограничения на значения феромонов для предотвращения застоя и преждевременной сходимости. MMAS ограничивает значения феромонов диапазоном [τmin, τmax], обеспечивая лучшее исследование пространства поиска.

```
def update_pheromones_mmas(pheromones, ants, tau_min, tau_max, evaporation_rate):

    pheromones *= (1 - evaporation_rate)

    for ant in ants:

        for i in range(len(ant.tour) - 1):

            pheromones[ant.tour[i]][ant.tour[i+1]] += 1.0 / ant.distance_travelled

            pheromones[ant.tour[i+1]][ant.tour[i]] += 1.0 / ant.distance_travelled

    pheromones = np.clip(pheromones, tau_min, tau_max)

    return pheromones
```

Этот подход поддерживает баланс между диверсификацией и интенсификацией, что приводит к улучшению производительности на различных экземплярах TSP.

3. Гибридные подходы

Комбинирование ACO с другими метаэвристиками позволяет использовать их взаимодополняющие преимущества. Например, система Genetic Ant System (GAS) объединяет ACO с генетическими алгоритмами (GA), используя возможности глобального поиска GA для повышения эффективности локального поиска ACO.

```
class GeneticAntSystem(TSPACO):
```

```
    def __init__(self, num_cities, distance_matrix, num_ants, alpha, beta, evaporation_rate, Q, crossover_rate,
mutation_rate):

        super().__init__(num_cities, distance_matrix, num_ants, alpha, beta, evaporation_rate, Q)

        self.crossover_rate = crossover_rate

        self.mutation_rate = mutation_rate

    def crossover(self, parent1, parent2):

        point = np.random.randint(1, self.num_cities - 1)

        child1 = parent1[:point] + [gene for gene in parent2 if gene not in parent1[:point]]

        child2 = parent2[:point] + [gene for gene in parent1 if gene not in parent2[:point]]

        return child1, child2

    def mutate(self, tour):
```

```
    if np.random.rand() < self.mutation_rate:

        i, j = np.random.randint(0, self.num_cities, 2)

        tour[i], tour[j] = tour[j], tour[i]

    return tour


def run_genetic(self, max_generations):

    best_tour = None

    best_distance = np.inf

    population = [self.initialize_ants() for _ in range(self.num_ants)]


    for generation in range(max_generations):

        new_population = []
```

```
for i in range(0, self.num_ants, 2):

    parent1, parent2 = population[i], population[i + 1]

    child1, child2 = self.crossover(parent1.tour, parent2.tour)

    new_population.append(self.mutate(child1))

    new_population.append(self.mutate(child2))

population = new_population



for ant in population:

    if ant.distance_travelled < best_distance:

        best_tour = ant.tour

        best_distance = ant.distance_travelled
```

```
        self.update_pheromones(population)

    return best_tour, best_distance
```

GAS улучшает разнообразие решений и повышает способность алгоритма избегать локальных оптимумов, что позволяет получать более качественные решения.

4. Интеграция методов локального поиска

Методы локального поиска, такие как 2-opt или 3-opt, интегрируются в ACO для уточнения решений, генерируемых муравьями. Эти методы итеративно улучшают туры путем внесения локальных изменений, повышая общее качество решения.

```
def two_opt(tour, distance_matrix):

    best_distance = calculate_tour_length(tour, distance_matrix)

    for i in range(1, len(tour) - 2):

        for j in range(i + 1, len(tour)):

            new_tour = tour[:i] + tour[i:j][::-1] + tour[j:]

            new_distance = calculate_tour_length(new_tour, distance_matrix)

            if new_distance < best_distance:

                tour = new_tour

                best_distance = new_distance
```

```
    return tour

def calculate_tour_length(tour, distance_matrix):

    return sum(distance_matrix[tour[i-1]][tour[i]] for i in range(len(tour)))
```

Благодаря использованию этих методов алгоритм не только находит хорошие решения, но и дорабатывает их до уровня, близкого к оптимальному, что значительно повышает показатели производительности.

Эмпирические исследования, сравнивающие эти оптимизированные варианты ACO, демонстрируют существенное улучшение производительности. В таблице 4 приведены результаты сравнительного анализа различных версий ACO на эталонном экземпляре TSP, подчеркивающие улучшение качества решений и скорости сходимости.

Таблица 4 - Сравнение производительности оптимизированных алгоритмов ACO на TSP

DOI: https://doi.org/10.60797/IRJ.2024.145.64.6

Алгоритм	Лучшее расстояние	Среднее расстояние	Время сходимости, с
Standard ACO	1320	1355	0,85
MMAS	1295	1328	0,79
GAS	1278	1305	0,72
ACO + 2-opt	1256	1284	0,68

В заключение следует отметить, что оптимизация и улучшение ACO – это многогранный подход, включающий в себя настройку параметров, гибридизацию и интеграцию методов локального поиска [13], [14], [15]. Эти усовершенствования устраняют ограничения, присущие базовому алгоритму ACO, что приводит к получению надежных и эффективных решений TSP.

Заключение

Исследование и совершенствование алгоритмов оптимизации муравьиной колонии (ACO) для решения задачи о путешествующем продавце (TSP) позволило добиться значительных успехов и улучшений, о чем свидетельствует тщательная теоретическая и экспериментальная работа, представленная в данном исследовании. Целью данного исследования было углубление понимания механизмов ACO, оптимизация его параметров и сравнение его эффективности с другими эвристическими и метаэвристическими подходами. Полученные результаты однозначно подтверждают гипотезу о том, что усовершенствованные алгоритмы ACO могут достигать более высоких показателей эффективности, таких как качество решения и скорость сходимости, по сравнению с классическими эвристиками и метаэвристиками.

Конфликт интересов

Не указан.

Рецензия

Все статьи проходят рецензирование. Но рецензент или автор статьи предпочли не публиковать рецензию к этой статье в открытом доступе. Рецензия может быть предоставлена компетентным органам по запросу.

Conflict of Interest

None declared.

Review

All articles are peer-reviewed. But the reviewer or the author of the article chose not to publish a review of this article in the public domain. The review can be provided to the competent authorities upon request.

Список литературы на английском языке / References in English

1. Christofides N. Worst-case analysis of a new heuristic for the travelling salesman problem / N. Christofides // Operations Research Forum. — Cham: Springer International Publishing, 2022. — Vol. 3. — № 1. — P. 20.

2. Dorigo M. Optimization, learning and natural algorithms / M. Dorigo. — 1992.

3. Dorigo M. Ant system: optimization by a colony of cooperating agents / M. Dorigo, V. Maniezzo, A. Colorni // IEEE transactions on systems, man, and cybernetics, part b (cybernetics). — 1996. — Vol. 26. — № 1. — P. 29-41.

4. Dorigo M. Ant colony optimization / M. Dorigo // Scholarpedia. — 2007. — Vol. 2. — № 3. — P. 1461.

5. Doerner K.F. Pareto ant colony optimization with ILP preprocessing in multiobjective project portfolio selection / K.F. Doerner [et al.] // European Journal of Operational Research. — 2006. — Vol. 171. — № 3. — P. 830-841.

6. Eyckelhof C.J. Ant systems for a dynamic TSP: Ants caught in a traffic jam / C.J. Eyckelhof, M. Snoek // International workshop on ant algorithms. — Berlin: Springer Berlin Heidelberg, 2002. — P. 88-99.

7. Gambardella L.M. Ant-Q: A reinforcement learning approach to the traveling salesman problem / L.M. Gambardella, M. Dorigo // Machine learning proceedings 1995. — Morgan Kaufmann, 1995. — P. 252-260.

8. Lawler E.L. The traveling salesman problem: a guided tour of combinatorial optimization / E.L. Lawler // Wiley-Interscience Series in Discrete Mathematics. — 1985.

9. Stützle T. MAX—MIN ant system / T. Stützle, H.H. Hoos // Future generation computer systems. — 2000. — Vol. 16. — № 8. — P. 889-914.

10. Dorigo M. Ant colony optimization: a new meta-heuristic / M. Dorigo, G. Di Caro // Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406). — IEEE, 1999. — Vol. 2. — P. 1470-1477.

11. Dorigo M. Ant system: optimization by a colony of cooperating agents / M. Dorigo, V. Maniezzo, A. Colorni // IEEE transactions on systems, man, and cybernetics, part b (cybernetics). — 1996. — Vol. 26. — № 1. — P. 29-41.

12. Dorigo M. Ant colony optimization theory: A survey / M. Dorigo, C. Blum // Theoretical computer science. — 2005. — Vol. 344. — № 2-3. — P. 243-278.

13. Yu B. An improved ant colony optimization for vehicle routing problem / B. Yu, Z.Z. Yang, B. Yao // European journal of operational research. — 2009. — Vol. 196. — № 1. — P. 171-176.

14. Wang Y. Ant colony optimization for traveling salesman problem based on parameters optimization / Y. Wang, Z. Han // Applied Soft Computing. — 2021. — Vol. 107. — P. 107439.

15. Wu G. Ensemble strategies for population-based optimization algorithms – a survey / G. Wu, R. Mallipeddi, P.N. Suganthan // Swarm and evolutionary computation. — 2019. — Vol. 44. — P. 695-711.