

## Обзор решений задачи коммивояжёра

Ключевые слова: задача коммивояжёра, муравьиный алгоритм, динамическое программирование

### Аннотация

Задача коммивояжёра является NP-полной, в связи с чем возникают проблемы с её решением. Существует множество способов найти как кратчайший путь, так и приближение к нему. В данной статье сравнивались различные алгоритмы для решения этой задачи.

### Введение

Задача коммивояжёра является одной из задач в логистике[1][2] и имеет множество алгоритмов решения. Какие-то из них после завершения своей работы находят наикратчайший маршрут (например алгоритм ветвей и границ[3]). Другие же имеют меньшую вычислительную сложность. Такое разнообразие решений позволяет выбирать метод, подходящий под конкретную задачу, учитывая все его преимущества и слабые места

Объектами данного исследования являются методы решения задачи коммивояжёра. Предметы исследования - точность и ресурсозатратность алгоритмов решения задачи коммивояжёра. Целью исследования является сравнение методов решения задачи коммивояжёра. Были поставлены следующие задачи:

1. Обзор методов
2. Анализ точности методов
3. Анализ эффективности по времени
4. Анализ эффективности по памяти
5. Сравнение.

### Обзор предметной области

#### Принцип отбора аналогов

Существует множество решений задачи, основанных на разных подходах. Поэтому с помощью "Google Академия" был произведён поиск исследований алгоритмов разных подходов и взяты несколько методов из этих групп. Запросы были следующие:

- \* "точные методы решения задачи коммивояжёра"
- \* "анализ задачи коммивояжёра"
- \* "муравьиный алгоритм".

### *Метод полного перебора*

Алгоритм заключается в полном переборе всех возможных маршрутов и нахождения кратчайшего.[4]

### *Динамическое программирование*

Точный алгоритм, идея которого состоит в нахождении для каждой вершины Б другой вершины А, такой что сумма длины пути из начальной точки до вершины А и длины пути А-Б минимальна в сравнении с другими вершинами.[3]

### *Метод ветвей и границ*

Точный метод, основанный на полном переборе, в котором не рассматриваются варианты, заведомо длиннее полученного ранее результата.[3]

### *Муравьиный алгоритм*

Этот алгоритм основан на идее, что муравей помечает путь от муравейника до своей цели феромоном, который со временем испаряется. Как следствие, чем короче путь, тем больше феромона на нём останется после возвращения муравья и тем выше вероятность, что муравей снова пойдёт по нему, вновь оставляя феромон.[4]

### *"Жадный" алгоритм*

"Жадный" алгоритм начинает выполнение с начальной вершины А и эта вершина помечается как "посещённая". Далее выбирается ближайшая к А вершина Б. Следом, вершина Б становится текущей и "посещённой". Потом из этой вершины также выбирается ближайшая к ней вершина В, которая ещё не отмечена как "посещённая". Таким образом "посещаются" все вершины [4]

## **Критерии сравнения аналогов**

### *Вычислительная сложность*

Данный критерий оценивает зависимость объема работы, которая выполняется алгоритмом, от размера входных данных (в данном случае - от числа рассматриваемых сигналов).

### *Точность*

Оценка точности алгоритма позволяет понять, какой результат будет получен

### *Затраты времени*

Этот критерий позволяет оценить скорость работы алгоритма.

### *Затраты памяти*

Оценка затрат памяти позволяет оценить сколько алгоритму понадобится свободного места.

### Таблица сравнения аналогов

Сравнение аналогов по описанным выше критериям приведено в таблице 1.

**Таблица 1 – Сравнение алгоритмов и методов по ключевым критериям**

Метод	Точность	Затраты времени	Затраты памяти
Полный перебор	гарантировано будет найден кратчайший маршрут	$O(n!)$	$O(n)$ , т.к. хранится только текущий маршрут
Динамическое программирование	гарантировано будет найден кратчайший маршрут	$O(n^2 2^n)[3]$	$O(n^2 2^n)[3]$
Метод ветвей и границ	гарантировано будет найден кратчайший маршрут	не сложнее, чем $O(n!)$ , поскольку полный перебор является частным случаем	$O(n)$
"Жадный" алгоритм	кратчайший маршрут будет найден только при условии, что он является "жадным" маршрутом	$O(n^2)$	$O(n)$
Муравьиный алгоритм	кратчайший маршрут будет найден с ненулевой вероятностью	$O(n^2)[5]$	$O(n)$

### Выводы по итогам сравнения

Методы для гарантированного нахождения кратчайшего маршрута требуют либо много времени, либо много памяти, при этом алгоритмы для нахождения приближения не могут гарантировать нахождение точного решения.

### Выбор метода решения

Если в задаче требуется найти строго кратчайший путь, то нужно выбирать один из точных методов, при этом алгоритм ветвей и границ имеет меньшую времязатратность по сравнению с полным перебором и требует меньше памяти по сравнению с динамическим программированием.

Если в задаче достаточно найти приближение к точному решению, то муравьиный алгоритм имеет преимущество.

## **Заключение**

В ходе работы были рассмотрены 5 методов решения задачи коммивояжёра. Для них были определены критерии сравнения, которые позволяют оценить достоинства и недостатки алгоритмов. Направлением дальнейших исследований будет следующее:

- 1) Проанализировать модификации муравьиного алгоритма
- 2) Проанализировать модификации алгоритма ветвей и границ .

## **Список использованных источников**

1. Ковалева Е. А. АНАЛИЗ ВЫБОРА ОПТИМАЛЬНОГО МАРШРУТА ДЛЯ ПЕРЕДВИЖЕНИЯ ГРУЗОВОГО ТРАНСПОРТА В МЕГАПОЛИСЕ // НАУЧНАЯ СЕССИЯ ГУАП. – 2016. – С. 91-95.
2. Гребенников М. В., Лысенко А. С., Рыжкин П. П. ЗАДАЧА КОММИВОЯЖЁРА В ЛОГИСТИКЕ // Международная научно-техническая конференция молодых ученых БГТУ им. ВГ Шухова, посвященная 170-летию со дня рождения. – 2023. – С. 49.
3. Меламед И. И., Сергеев С. И., Сигал И. Х. Задача коммивояжера. Точные методы // Автоматика и телемеханика. – 1989. – №. 10. – С. 3-29.
4. Семенов С. С. и др. Анализ трудоемкости различных алгоритмических подходов для решения задачи коммивояжера // Системы управления, связи и безопасности. – 2017. – №. 1. – С. 116-131
5. Штовба С. Д. Муравьиные алгоритмы // Exponenta Pro. Математика в приложениях. – 2003. – Т. 4. – №. 4. – С. 70-75.