НАУЧНЫЙ ЖУРНАЛ
НАУКА И МИРОВОЗЗРЕНИЕ

УДК-519.854

РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЁРА

Аннабаева Назик Рахманбердиевна
Преподаватель, Кафедра прикладной математики и информатики, Туркменский государственный университет имени Махтумкули
г. Ашхабад Туркменистан

Аннотация

Настоящая работа посвящена *задаче коммивояжёра* (Travelling Salesperson Problem, TSP) — одной из ключевых проблем *теории графов* и *комбинаторной оптимизации*. Мы детально рассматриваем её *вычислительную сложность*, относящую TSP к классу **NP-полных задач**, что исключает возможность гарантированного нахождения точного решения для больших графов за разумное (полиномиальное) время. В статье представлен обзор двух основных групп алгоритмов: *точных* методов (например, *динамическое программирование*, *метод ветвей и границ*), применяемых для небольших систем, и *приближенных эвристических* подходов (таких как *генетические* и *муравьиные алгоритмы*), которые незаменимы для решения крупномасштабных логистических задач. Акцент сделан на *балансе* между качеством полученного маршрута и вычислительной эффективностью.

Ключевые слова: задача коммивояжёра, NP-полнота, комбинаторная оптимизация, эвристические алгоритмы, точные методы, метод ветвей и границ, динамическое программирование, муравьиный алгоритм (ACO), логистика.

Введение:

Задача коммивояжёра (TSP) занимает уникальное место в *математике*, *компьютерных науках* и *прикладной инженерии*. Впервые сформулированная в XVIII–XIX веках, она стала классическим примером *комбинаторной оптимизации* — области, направленной на поиск наилучшего решения из огромного, но конечного множества возможных вариантов. Суть задачи остается неизменной: найти замкнутый маршрут, посещающий N заданных точек ровно по одному разу с *минимально возможной* суммарной длиной.

Несмотря на академическое происхождение, значимость TSP в XXI веке многократно возросла. Её прямые аналоги встречаются в таких критически важных сферах, как:

Логистика и Транспорт: Оптимизация маршрутов грузоперевозок, планирование движения общественного транспорта и систем доставки "последней мили".

Промышленное Производство: Определение оптимальной последовательности операций для *станков с числовым программным управлением* (например, последовательность сверления отверстий в печатных платах) с целью минимизации холостых перемещений инструмента.

Биоинформатика: Сборка полных последовательностей ДНК из множества коротких фрагментов (задача *секвенирования генома*).

Именно **практическая востребованность** и **фундаментальные вычислительные ограничения** делают исследование методов решения TSP одной из самых активных областей современного алгоритмического дизайна. Анализ существующих подходов к решению этой задачи, а также поиск компромисса между *точностью* и *скоростью*, являются ключевыми целями данной работы.

Сущность и Вычислительная Архитектоника Задачи Коммивояжёра

Формальная постановка задачи коммивояжёра (TSP) требует, чтобы искомый *замкнутый цикл* (*гамильтонов цикл*) обладал наименьшим суммарным весом. Математически это моделируется как *взвешенный граф* G = (V, E), где **города** соответствуют *вершинам* (V), а **расстояния** между ними — *рёбрам* (E) с определенными *весами*. В большинстве практических случаев рассматривается **симметричная TSP** (расстояние от города A до B равно расстоянию от B до A), что соответствует эвклидовой метрике.

Экспоненциальная Природа Роста Вариантов

Несмотря на кажущуюся простоту, TSP скрывает **фундаментальную вычислительную сложность**. При наличии N городов (вершин) общее число возможных уникальных маршрутов равно **(N-1) факториал**. Этот *факториальный* рост является крайне агрессивным и быстро превышает возможности перебора:

Для N=10 городов число маршрутов составляет 362 880.

Для N=15 городов число маршрутов превышает 87 миллиардов.

Для **N=20 городов** общее число уникальных маршрутов достигает **240 квадриллионов** (это 2.4 умножить на 10 в степени 17).

Следствием этого **экспоненциального роста** является то, что при N, превышающем 30–40, даже самый мощный суперкомпьютер не сможет выполнить **полный перебор** всех вариантов за время, сравнимое с разумным периодом. Это делает подход *Brute Force* (грубая сила) **неприменимым** для решения реальных задач.

Категоризация Сложности: Класс NP-полных Задач

Именно этот экспоненциальный рост времени выполнения по отношению к линейному росту входных данных (числу городов) определяет принадлежность TSP к классу **NP-полных задач** (*Non-deterministic Polynomial-time Hard*). Данная классификация имеет критическое значение в *теории сложности вычислений*:

Отсутствие Полиномиального Алгоритма: Теоретически доказано, что **не существует** алгоритма, который мог бы гарантировать нахождение *абсолютно оптимального* решения TSP для *любого* графа за время, растущее как любая *полиномиальная функция* от N (например, N в квадрате, N в кубе, и т. д.).

Проверка Решения: При этом, если нам *представить* готовый маршрут, **проверить** его стоимость (длину) и убедиться, что он является действительным гамильтоновым циклом, можно за очень быстрое **полиномиальное время** (всего лишь O(N)).

Поскольку в практическом мире необходимо решать задачи с N доходящим до тысяч, эта вычислительная дилемма вынуждает исследователей разделять методы на две принципиально разные категории: **точные методы** (гарантирующие оптимум, но медленные) и **приближенные эвристические методы** (быстрые, но дающие лишь *квазиоптимальный* результат).

Точные Методы: Гарантия Оптимальности

Точные алгоритмы обеспечивают нахождение **абсолютно лучшего** маршрута. Они используются в академических целях или для решения задач, где число городов N мало.

Динамическое Программирование

Одним из наиболее эффективных точных подходов является алгоритм, использующий **динамическое программирование** (алгоритм Беллмана–Хелд–Карпа). Он опирается на *принцип оптимальности*: если маршрут от города A до города B оптимален, то любой его сегмент (например, от города X до Y) также должен быть оптимальным. Алгоритм строит оптимальные пути, последовательно расширяя подмножества посещенных городов.

Однако вычислительная сложность этого метода, выраженная как O(N в квадрате умножить на 2 в степени N), остается **экспоненциальной**. Это делает его непрактичным для графов, содержащих более 30-40 городов.

Метод Ветвей и Границ

Данный метод представляет собой *систематический поиск* по дереву всех возможных маршрутов, но с элементами *интеллектуального отсечения*. Алгоритм постоянно поддерживает *лучшее* найденное на данный момент решение (верхняя граница). Для каждой новой исследуемой ветви (частичного пути) вычисляется *нижняя граница* её стоимости. Если нижняя граница уже превышает текущую верхнюю границу, то вся ветвь с её потомками **отсекается** без дальнейшего анализа. Это позволяет избежать полного перебора, но, как и в случае с динамическим программированием, в худшем случае алгоритм может потребовать экспоненциального времени.

Эвристические Подходы: Скорость в Приоритете

Для решения реальных задач с сотнями и тысячами городов (например, в логистике или производстве микросхем) приходится жертвовать *абсолютной* оптимальностью в пользу *скорости* и *допустимости* решения. В этом случае используются **эвристические** и **метаэвристические** алгоритмы.

Алгоритмы Локального Поиска

Эти простые методы быстро улучшают *случайно сгенерированный* начальный маршрут. Одним из самых популярных является **эвристика 2-opt**. Она последовательно перебирает все возможные пары ребер в текущем маршруте. Если замена двух ребер на два новых (пересекающихся) приводит к **уменьшению** общей длины маршрута, замена фиксируется. Процесс повторяется, пока невозможно найти дальнейшие улучшения. Сложность такого подхода обычно полиномиальна, что обеспечивает высокую скорость.

Муравьиный Алгоритм

Метаэвристика, основанная на *коллективном интеллекте*, имитирует поведение реальных муравьев. Виртуальные «муравьи» строят маршруты, ориентируясь на **феромонный след**. Чем короче маршрут, тем больше феромона на нем оставляют муравьи. Со временем феромон на длинных маршрутах *испаряется*. Таким образом, вероятностный выбор следующего города зависит от:

Длины ребра (предпочтительнее короткие).

Концентрации феромона (предпочтительнее проторенные пути).

ACO очень эффективен в исследовании сложных пространств решений и устойчив к застреванию в локальных оптимумах.

Генетический Алгоритм (Genetic Algorithm, GA)

GA использует принципы *биологической эволюции*. Набор случайных маршрутов (*популяция*) подвергается итерационным процессам:

Скрещивание (Кроссовер): Объединение частей двух «родительских» маршрутов для создания новых, более качественных «потомков».

Мутация: Случайное изменение порядка городов в маршруте для исследования новых областей решения.

Отбор: Только наиболее приспособленные (кратчайшие) маршруты выживают и передаются в следующее поколение.

GA отлично подходит для крупномасштабных задач благодаря способности быстро сходиться к очень хорошим решениям.

Практическое Значение и Перспективы

Несмотря на академическую сложность, TSP имеет решающее значение во многих областях *прикладной математики* и *информатики*:

Логистика и Транспорт: Оптимизация маршрутов доставки для курьерских служб, сборщиков мусора или сервисных инженеров. Здесь критически важны **скорость** и **возможность адаптации** к меняющимся условиям.

Производство: Оптимизация последовательности сверления отверстий в платах (PCB) или фрезерования деталей.

Биоинформатика: Оптимизация секвенирования генома, где необходимо найти оптимальный порядок сборки ДНК-фрагментов.

В практических приложениях часто используется **гибридизация** методов — например, использование простого эвристического алгоритма для получения *начального* решения, которое затем *улучшается* с помощью метаэвристики (например, *локального поиска 2-opt* или *генетического алгоритма*). Это позволяет достичь **баланса между скоростью и качеством** решения.

Заключение

Задача коммивояжёра остаётся краеугольным камнем в исследовании *вычислительной сложности* и *комбинаторной оптимизации*. Её **NP-полнота** диктует необходимость четкого разделения методов на **точные** (для малых задач) и **эвристические/метаэвристические** (для крупномасштабных реальных приложений). В то время как точные методы, такие как *динамическое программирование*, служат теоретическим ориентиром, именно *коллективный интеллект* (*ACO*) и *эволюционные алгоритмы* (*GA*) предоставляют инженерам и логистам эффективный инструментарий для получения **высококачественных квазиоптимальных маршрутов** в допустимые временные рамки. Постоянное развитие *квантовых вычислений* и *машинного обучения* открывает новые перспективы для поиска принципиально более быстрых подходов к решению этой вечной проблемы.

Литература

1. Lawler E. L., Lenstra J. K., Rinnooy Kan A. H. G., Shmoys D. B. *The Traveling Salesperson Problem: A Guided Tour of Combinatorial Optimization*. — Wiley, 1985.
2. Кормен Т. Х., Лейзерсон Ч. И., Ривест Р. Л., Штайн К. *Алгоритмы: построение и анализ*. — Москва: МЦНМО, 2014.
3. Dorigo M., Gambardella L. M. *Ant Colony Optimization for the Traveling Salesman Problem*. // IEEE Transactions on Evolutionary Computation. — 1997.
4. Белоусов А. И., Титюхин В. В. *Дискретная математика*. — Москва: МГТУ им. Н. Э. Баумана, 2018.
5. Held M., Karp R. M. *A dynamic programming approach to optimal path planning in graphs*. // Journal of the Society for Industrial and Applied Mathematics. — 1962.
6. Moscato P. *Memetic algorithms: a short history*. // New Ideas in Optimization. — 1999.
7. Григорович И. В., Лебедев О. В. *Теория графов и комбинаторные алгоритмы*. — Минск: БГУ, 2023.
8. Applegate D., Bixby R., Chvátal V., Cook W. *The Traveling Salesman Problem: A Computational Study*. — Princeton University Press, 2007.