МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тверской государственный технический университет»

(ТвГТУ)

Кафедра “Программного обеспечения”

**Курсовая работа**

по дисциплине “Теория вычислительных процессов”

Тема: «Задача коммивояжера»

Выполнил: студент группы

ПИН 17.06

Завгороднев Е. Ю

Проверил:

Калабин А.Л

Тверь 2021

Оглавление

[Цель работы 3](#_Toc68678658)

[Постановка задач 3](#_Toc68678659)

[Описание точного алгоритма 3](#_Toc68678660)

[Описание приближенного алгоритма 4](#_Toc68678661)

[Зависимость времени решения от размерности входных данных для точного решения 4](#_Toc68678662)

[Зависимость времени решения от размерности входных данных для приближенного решения 5](#_Toc68678663)

[Зависимость точности решения от размерности входных данных для приближенного решения 5](#_Toc68678664)

[Выводы по проделанной работе 7](#_Toc68678665)

[Литература 7](#_Toc68678666)

# Цель работы

Сравнение точного и приближенного алгоритма решения задачи коммивояжера.

# Постановка задач



Задача коммивояжёра (англ. Travelling salesman problem, TSP) (коммивояжёр — разъездной сбытовой посредник) — одна из самых известных задач комбинаторной оптимизации, заключающаяся в отыскании самого выгодного маршрута, проходящего через указанные города хотя бы по одному разу с последующим возвратом в исходный город. В условиях задачи указываются критерий выгодности маршрута (кратчайший, самый дешёвый, совокупный критерий и т. п.) и соответствующие матрицы расстояний, стоимости и т. п)

1. Решение задачи точным алгоритмом
2. Решение задачи приближенным алгоритмом: метод минимального остовного дерева
3. Построить зависимости времени решения от размерности входных данных.
4. Сравнить эту зависимость с зависимостью для точного решения.
5. Построить зависимость точности решения от размерности входных данных

# Описание точного алгоритма

1. Рекурсивным способом считаем расстояния всех путей между городами
2. Рекурсивная функция создает перестановки n чисел. Она решает чуть более общую задачу: переставляет всеми возможными способами элементы массива, начиная с индекса lf и вправо от него. Для этого, поочерёдно ставит на место элемента ar[lf] все элементы ar[i] c i ≥ lf (переставляя их местами) и рекурсивно повторяет действия для массива меньшего размера, начиная с индекса lf+1.
3. Выбираем самый оптимальный.

# Описание приближенного алгоритма

Алгоритм Прима — алгоритм построения минимального остовного дерева взвешенного связного неориентированного графа.

На вход алгоритма подаётся связный неориентированный граф. Для каждого ребра задаётся его стоимость.

Сначала берётся произвольная вершина и находится ребро, инцидентное данной вершине и обладающее наименьшей стоимостью. Найденное ребро и соединяемые им две вершины образуют дерево. Затем, рассматриваются рёбра графа, один конец которых — уже принадлежащая дереву вершина, а другой — нет; из этих рёбер выбирается ребро наименьшей стоимости. Выбираемое на каждом шаге ребро присоединяется к дереву. Рост дерева происходит до тех пор, пока не будут исчерпаны все вершины исходного графа.

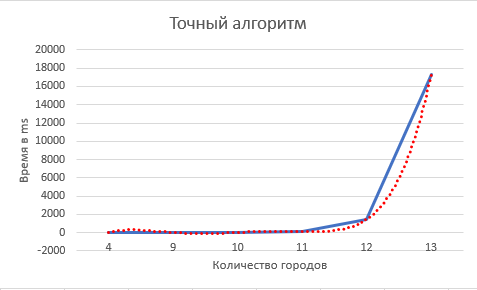
Результатом работы алгоритма является остовное дерево минимальной стоимости.

Задача коммивояжера может быть сформулирована в терминах теории графов как задача о нахождении минимального остовного дерева в графе, вершины которого представляют города, рёбра — это пары городов, между которыми можно проложить прямую дорогу, а вес ребра равен стоимости строительства соответствующей дороги.

# Зависимость времени решения от размерности входных данных для точного решения

Тестирование производилось на 4, 9, 10, 11, 12, 13, 20 городах.

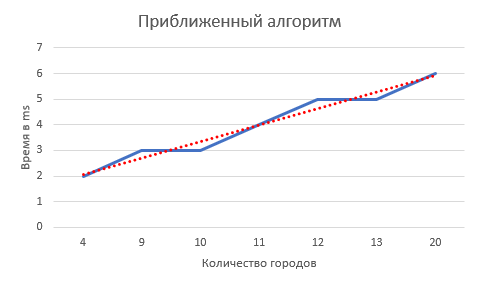
Результат:



Время очень быстро растет с количеством входных данных.

С количеством городов > 15 трудно выполнять задачу на обычных ПК.

# Зависимость времени решения от размерности входных данных для приближенного решения

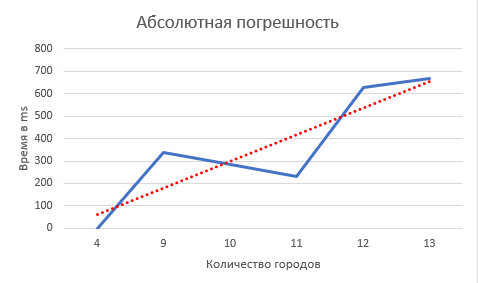


В приближенном алгоритме время изменяется равномерно. ПК может вычислить задачу с количеством городов больше 15. Однако значительный выигрыш по времени компенсируется точностью решения.

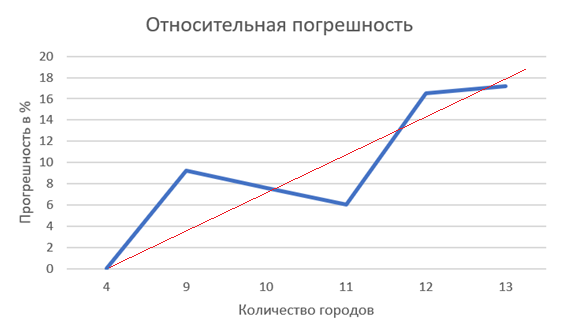
# Зависимость точности решения от размерности входных данных для приближенного решения

Абсолютная погрешность

|  |  |
| --- | --- |
| Количество городов | Абсолютная погрешность |
| 4 | 0 |
| 9 | 339 |
| 10 | 284 |
| 11 | 233 |
| 12 | 630 |
| 13 | 669 |
| 20 | - |



|  |  |
| --- | --- |
| Количество городов | Относительная погрешность |
| 4 | 0 |
| 9 | 9,24 % |
| 10 | 7,63 % |
| 11 | 6,07 % |
| 12 | 16,53 % |
| 13 | 17,16 % |
| 20 | - |



Как можно увидеть, погрешность приближенного алгоритма возрастает с увеличением входных данных.

# Выводы по проделанной работе

Точный алгоритм отличается быстрым ростом времени выполнения при изменении входных данных, но дает правильное решение в любых случаях.

Приближенный алгоритм иногда может давать неточное решение, но это компенсируется скоростью выполнения.

# Литература

1 Мозговой М.В. Классика программирования: алгоритмы, языки, автоматы, компиляторы. Практический подход. — СПб.: Наука и Техника, 2006. — 320