



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА «ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ
ТЕХНОЛОГИИ» (ИУ7)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.04 Программная инженерия

О Т Ч Е Т

по лабораторной работе № 6

Название: Решение задачи коммивояжёра методом полного перебора и муравьиным алгоритмом

Дисциплина: Анализ алгоритмов

Студент

ИУ7-52Б

(Группа)

(Подпись, дата)

Е.В. Брянская

(И.О. Фамилия)

Преподаватель

Л.Л. Волкова

(Подпись, дата)

(И.О. Фамилия)

Москва, 2020

Оглавление

Введение	3
1 Аналитическая часть	4
1.1 Цель и задачи	4
1.2 Задача о коммивояжёре	4
1.3 Алгоритм полного перебора	4
1.4 Муравьиный алгоритм	5
Вывод	7
2 Конструкторская часть	8
3 Технологическая часть	9
4 Исследовательская часть	10
Заключение	11
Список литературы	12

Введение

В этой лабораторной работе будут рассматриваться два алгоритма, решающие задачу коммивояжёра.

Задача о коммивояжёре (travelling-salesman problem) является одной из знаменитых задач теории комбинаторики, была поставлена в 1934 году и заключается в поиске самого выгодного (минимального по стоимости) маршрута, проходящего строго по одному разу по всем приведённым городам с последующим возвратом в исходный город. Таким образом, выбор подходящего маршрута осуществляется среди гамильтоновых циклов.

Гамильтонов цикл - это такой цикл (замкнутый путь), который проходит через каждую вершину ровно по одному разу.

Задача коммивояжёра имеет ряд практических применений, к примеру, она использовалась для составления маршрутов лиц, занимающихся выемкой монет из таксофонов. В этом случае, в качестве пунктов, которые нужно посетить, выступали места установки таксофонов, а стоимость – время в пути между двумя точками.

Также она используется в задаче о сверлильном станке. Сверлильный станок изготавливает металлические листы с определённым количеством отверстий, координаты которых заранее известны. Нужно найти кратчайший путь через все отверстия, то есть наименьшее время, затрачиваемое на изготовление одной детали.

Для решения этой задачи есть несколько алгоритмов, в этой лабораторной работе будут рассмотрены: алгоритм полного перебора и муравьиный алгоритм.

1. Аналитическая часть

В этом разделе будут поставлены цель и основные задачи лабораторной работы, которые будут решаться по мере её выполнения.

1.1. Цель и задачи

Цель данной работы: провести сравнительный анализ метода полного перебора и эвристического метода на базе муравьиного алгоритма.

Для достижения поставленной цели необходимо решить следующий ряд **задач**:

- 1) дать описание базовой задачи;
- 2) описать алгоритмы;
- 3) реализовать все рассмотренные алгоритмы;
- 4) провести параметризацию муравьиного алгоритма для выбранного класса задач, то есть определить такие комбинации параметров или их диапазонов, при которых метод даёт наилучшие результаты.

1.2. Задача о коммивояжёре

В задаче о коммивояжёре, которая тесно связана с задачей о гамильтоновом цикле, коммивояжёр должен посетить n городов. Можно сказать, что коммивояжёру нужно совершить тур, или гамильтонов цикл, посетив каждый город ровно по одному разу и, завершив путешествие в том же городе, из которого он выехал. С каждым переездом из города i в город j связана некоторая стоимость пути $c(i, j)$, выраженная целым неотрицательным числом, и коммивояжёру нужно совершить тур таким образом, чтобы общая стоимость (т.е. сумма стоимостей всех переездов) была минимальной. [1]

Для решения этой задачи предлагается два следующих алгоритма.

1.3. Алгоритм полного перебора

Этот алгоритм заключается в полном переборе всех возможных комбинаций точек (городов). На вход подаётся число городов N и матрица стоимостей C . Так как количество городов равно N , то последовательно будут рассматриваться все перестановки из $N - 1$ положительных целых чисел. Будет анализироваться каждый из этих возможных туров, и будет выбран тот, у которого наименьшая стоимость. [2]

Этот алгоритм достаточно точный, но продолжительность таких вычислений может занять много времени.

1.4. Муравьиный алгоритм

В его основе лежит моделирование колонии муравьёв, которая существует t дней. Колония представляет собой систему, в которой строго определены правила автономного поведения особей. Особенностью является то, что члены колонии могут быстро находить кратчайший путь и обмениваться этой информацией. В начале каждого t -ого дня осуществляется проход очередной партии муравьёв по городам.

В природе происходит не прямой обмен информацией, при котором одна особь изменяет некоторую область окружающей среды, а другие используют эту информацию, когда попадают в эту область. Каждый муравей оставляет особое вещество – феромон, по которому ориентируются следующие за ним особи. Чем больше муравьёв проходит по какому-либо участку, тем выше на нём концентрация этого вещества, тем самым насекомые помечают наиболее выгодные пути.

В алгоритме происходит моделирование такого поведения на некотором графе, ребра которого представляют собой возможные пути перемещения, и, как результат, наиболее обогащённый феромонами путь по рёбрам этого графа и будет решением поставленной задачи.

В основе муравьиного алгоритма лежат следующие принципы.

- В силу того, что каждый город должен быть посещён только один раз, у каждого муравья в колонии хранится информация об уже посещённых пунктах (далее этот список будет обозначаться как J);
- Муравьи обладают «зрением», то есть желанием посетить тот или иной город j , и эта величина рассчитывается по формуле 1.1.

$$\eta_{ij} = \frac{1}{D_{ij}}, \quad (1.1)$$

где D_{ij} - расстояние между городами i и j .

- Как говорилось выше, немаловажную роль в выборе следующего участка пути, играет феромон, который оставили другие муравьи, его количество обозначается как

τ_{ij} . И вероятность того, что дальнейший маршрут k -ого муравья будет построен по текущему ребру определяется по формуле 1.2.

$$\begin{cases} P_{ij,k}(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in J_{i,k}} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta}, j \in J_{i,k}; \\ P_{ij,k}(t) = 0, j \notin J_{i,k}, \end{cases} \quad (1.2)$$

где α, β – коэффициенты стадности и жадности, то есть, параметры, которые задают веса феромона, при $\alpha = 0$ алгоритм вырождается до жадного алгоритма (то есть будет выбран ближайший город).

В случае, если муравью удалось построить маршрут, удовлетворяющий всем требованиям, на задействованных рёбрах увеличивается количество феромона. Насколько изменится величина этого вещества на конкретном участке, рассчитывается следующим образом (формула 1.3).

$$\Delta\tau_{ij,k}(t) = \begin{cases} \frac{Q}{L_k(t)}, (i, j) \in T_k(t); \\ 0, (i, j) \notin T_k(t), \end{cases} \quad (1.3)$$

где Q – параметр, имеющий значение порядка длины оптимального пути, L_k – длина маршрута, $T_k(t)$ – маршрут, пройденный k -ым муравьём к моменту времени t .

Помимо всего прочего, алгоритм учитывает, подобно природной экосистеме, испарение феромона с наступлением ночи t -ого дня, причём с такой скоростью, чтобы не забывались хорошие решения и не возникало преждевременной сходимости. Рассчёты производятся по формуле 1.4.

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t); \quad \Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij,k}(t), \quad (1.4)$$

где $\rho \in [0, 1]$ – коэффициент испарения, m – количество муравьёв в колонии.

Для большей модификации алгоритма используются «элитные» муравьи, которые усиливают рёбра наилучшего маршрута, найденного с начала работы алгоритма. [3]

В результате работы алгоритма находится кратчайший маршрут среди всех тех, которые были найдены. Но важно обратить внимание на то, что не всегда этот маршрут совпадёт с решением, которое можно получить с помощью алгоритма полного перебора, ввиду определённых причин. Можно получить возможно не совсем точное, но очень близкое к идеальному значение.

Вывод

Были поставлены цель и задачи текущей лабораторной работы, также дано описание рассматриваемых алгоритмов.

2. Конструкторская часть

3. Технологическая часть

4. Исследовательская часть

Заключение

Список литературы

1. Задача коммивояжёра [Электронный ресурс]. Режим доступа: <http://www.avprog.narod.ru/student/kommi.htm> свободный (дата обращения: 30.11.2020)
2. Кормен, Томас Х. и др Алгоритмы: построение и анализ, 3-е изд. : Пер. с англ. - М. : ООО "И.Д. Вильямс 2018. - 1328 с. : ил. - Парал. тит. англ. - ISBN 978-5-8459-2016-4 (рус.).
3. Ульянов М.В. Ресурсно-эффективные компьютерные алгоритмы. Разработка и анализ - Учебное пособие. М.: НАУКА, ФИЗМАТЛИТ, 2007. - 376 с. ISBN 978-5-9221-0950-5
4. Клейнберг Дж., Тардос Е. Алгоритмы: разработка и применение. Классика Computer Science /Пер. с англ. Е. Матвеева. - СПб.: Питер, 2016. - 800 с.: ил. - (Серия "Классика computer science"). ISBN 978-5-496-01545-5
5. Документация по C++ [Электронный ресурс]. Режим доступа: <https://docs.microsoft.com/ru-ru/cpp/cpp/?view=msvc-160>, свободный (дата обращения: 22.11.2020)
6. Документация по Visual Studio 2019 [Электронный ресурс]. Режим доступа: <https://docs.microsoft.com/ru-ru/visualstudio/windows/?view=vs-2019>, свободный (дата обращения: 21.11.2020)
7. QueryPerformanceCounter function [Электронный ресурс]. Режим доступа: <https://docs.microsoft.com/en-us/windows/win32/api/profileapi/nf-profileapi-queryperformancecounter>, свободный (дата обращения: 22.11.2020).