МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №2
по дисциплине «Построение и анализ алгоритмов»
Тема: Кратчайшие пути в графах: коммивояжёр
Вариант: 2

Студент гр. 3388 _____ Павлов А.Р.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург
2025

Цель работы:

Изучить алгоритмы, реализующие решение задачи коммивояжера и реализовать Метод Ветвей и Границ, а также алгоритм ближайшего соседа.

Задание:

Решить задачу Коммивояжёра 2 различными способами. Алгоритм Литтла с модификацией: после приведения матрицы, к нижней оценке веса решения добавляется нижняя оценка суммарного веса остатка пути на основе МОД.

Приближённый алгоритм: АБС. Начинать АБС со стартовой вершины.

Реализация

Алгоритма Литтла:

Задача коммивояжера состоит в построении наиболее выгодного гамильтонова пути по графу из городов. Между городами-вершинами проложены дороги-ребра разной стоимости. Алгоритм Литтла решает задачу, работая с матрицей смежности графа городов.

В начале алгоритм выполняет редукцию матрицы — вычитает из каждой строки значение ее минимального элемента. Тоже самое он делает и со столбцами. Сложив вычтенные значения, алгоритм получает так называемое значение нижней границы на текущем шаге. Путь, стоимостью меньше этого значения построить невозможно. Редукция будет выполняться в дальнейшем на каждом шаге алгоритма, получая новые значения нижней границы.

Далее алгоритм строит бинарное дерево матриц. Принцип таков: одна ветвь дерева содержит все пути, в которых будет определенное ребро, другая ветвь — все пути, где это ребро остутствует. Алгоритм выбирает ребро с максимальным штрафом и создает двух потомков первоначальной матрицы по этому ребру. К нижней границе ветви дерева, где было запрещено ребро, необходимо прибавить значение штрафа запрещенного ребра. В дальнейшей работе, при определенных условиях, ветвь решений можно будет «отсечь» как заведомо невыгодную.

Чтобы запретить ребро, алгоритм в нужной ячейке матрицы смежности устанавливает бесконечность.

В правой ветви дерева алгоритм, наоборот, добавляет ребро в буфер решения. По определению гамильтоного пути, добавление ребра в решение автоматически запрещает множество путей, соответствующих переходу в «конец» ребра и выходу из «начала» ребра.

Это означает, что в матрице смежности алгоритм обращает в бесконечность все элементы строки и столбца, в которых расположено ребро, запре-

щая эти пути. После этого происходит редукция матрицы с обновлением нижней границы для этой ветви.

После этого происходит повтор: выбор ребра с максимальным штрафом и ветвление по нему.

Нижняя граница

Выше объясняется оценка нижней границы по сумме редукций. В модифицированном варианте алгоритма Литтла используется МОД — метод минимальных остовных деревьев.

Алгоритм Прима строит минимальное остовное дерево по текущей матрице смежности и стоимость прохождения по этому дереву может превосходить нижнюю границу, построенную по редукциям. Т.е такой подход может ускорить работу алгоритма, за счёт более строгой оценки нижних границ.

Отслеживание циклов

На какой-то итерации алгоритм может выбрать ребро, которое замкнёт гамильтонов цикл до прохода по всем вершинам. Чтобы избежать этой ошибки, и перед добавлением ребра происходит проверка цепочки ребер на корректность.

Таким образом алгоритм сохраняет все текущие матрицы и их нижние границы в очередь с приоритетом. В начале каждой итерации выбирается узел с минимальной нижней границей, чтобы продолжить ветвление с него.

В итоге цикл завершится, когда закончатся непосещённые вершины и будет получен первый гамильтонов цикл.

Далее алгоритм начинает искать лучший путь. Все ветви, нижние границы которых будут больше, чем стоимость прохода по текущему лучшему пути, отсекаются — они a priori не дадут лучшее решение.

Если найдется путь с меньшей стоимостью, то он станет новым лучшим.

Алгоритм завершит свою работу, когда будет найден маршрут, чья стоимость не будет превосходить все оставшиеся нижние границы, т.е когда опустошится очередь с приоритетом.

Жадный алгоритм

Дополнительно реализован второй способ решения задачи, использующий жадный алгоритм — Алгоритм Ближайшего Соседа.

Алгоритм стартует в произвольной вершине и на каждом шаге выбирает непосещенную вершину, путь к которой даст наименьшие затраты.

После посещения всех вершин алгоритм проверяет возможность возвращения в стартовую вершину.

Жадный алгоритм редко возвращает оптимальный путь, однако результат не будет отличаться от оптимального более чем в два раза. При этом алгоритму требуется намного меньше времени для работы, что делает его очень даже применимым на практике.

Оценка сложности алгоритма:

Временная сложность: $O\left(n^2 \cdot 2^n\right)$

Редукция матрицы: $O\left(n^2\right)$ для каждой строки и столбца (по n элементов).

Поиск ячейки с максимальным штрафом: $O\left(n^2\right)$ для проверки всех ячеек и вычисления штрафов.

Построение МОД: $O\left(n^2\right)$ для создания графа и $O(nlogn)$ для сортировки ребер (в худшем случае $O\left(n^2\right)$ из-за числа ребер).

Количество узлов: В худшем случае алгоритм исследует все возможные деревья, что дает $O\left(2^n\right)$ узлов.

Однако отсечение по границам значительно сокращает количество исследуемых узлов в среднем случае, делая алгоритм эффективнее полного перебора $O(n!)$.

Оценка сложности алгоритма:

Временная сложность: $O\left(n^2\right)$

$n - 1$ итераций по вершинам, n проверок при поиске ближайшего соседа на каждой итерации.

Вывод

В ходе лабораторной работы было написано решение задачи коммивояжера двумя способами. Алгоритм Литтла, основанный на методе ветвей и границ, находит оптимальный путь за приемлемое время по сравнению с полным перебором.

Жадный алгоритм и аналитически и экспериментально проигрывает первому алгоритму в точности, однако имеет значительное преимущество в скорости работы.