ДИСЦИПЛИНА	**Системы синхронизированного планирования ресурсов предприятия**
	(полное наименование дисциплины без сокращений)
ИНСТИТУТ	**информационных технологий**
КАФЕДРА	**корпоративных информационных систем**
	полное наименование кафедры
ВИД УЧЕБНОГО МАТЕРИАЛА	**Методические указания к практической работе**
	(в соответствии с пп.1-11)
ПРЕПОДАВАТЕЛЬ	**Демидова Лилия Анатольевна**
	(фамилия, имя, отчество)
СЕМЕСТР	**2 семестр (весенний), 2023 – 2024 учебный год**
	(семестр обучения, учебный год)

Задача коммивояжера

Постановка задачи

Классическая постановка задачи о коммивояжере (TSP, travelling salesman problem) выглядит следующим образом:

Имеется N городов, которые должен обойти коммивояжер с минимальными затратами. При этом на его маршрут накладывается два ограничения:

– маршрут должен быть замкнутым, то есть коммивояжер должен вернуться в тот город, из которого он начал движение;

– в каждом из городов коммивояжер должен побывать точно один раз, то есть надо обязательно обойти все города, не побывав ни в одном городе дважды.

Для расчета затрат существует матрица условий, содержащая затраты на переход из каждого города в каждый, при этом считается, что можно перейти из любого города в любой, кроме того же самого (в матрице как бы вычеркивается диагональ).

Целью решения является нахождения маршрута, удовлетворяющего всем условиям и при этом имеющего минимальную сумму затрат.

Методы решения

Для решения «Задачи коммивояжера» (ЗК) существует несколько методов решения. Они отличаются, прежде всего, скоростью решения и точностью.

«Жадный» алгоритм

Жадный алгоритм – алгоритм нахождения наикратчайшего расстояния путем выбора самого короткого, еще не выбранного ребра при условии, что оно не образует цикл цикла с уже выбранными ребрами. Жадным этот алгоритм называется потому, что на последних шагах приходится жестоко расплачиваться за его жадность. Алгоритм быстрый, но не точный.

Деревянный алгоритм

Суть метода состоит в построении кратчайшего остовного дерева. Погрешность такого алгоритма равна единице, скорость высокая.

Полный перебор

Практически применим только в задачах малого размера. Напомним, что ЗК с n городами требует при полном переборе рассмотрения (n-1)!/2 туров в симметричной задаче и (n-1)! туров в несимметричной задачи.

Метод ветвей и границ

К идее метода ветвей и границ приходили многие исследователи, но Литтл с соавторами на основе указанного метода разработали удачный алгоритм решения ЗК и тем самым способствовали популяризации подхода. С тех пор метод ветвей и границ был успешно применен ко многим задачам, для решения ЗК было придумано несколько других модификаций метода, но в большинстве учебников излагается пионерская работа Литтла.

Общая идея тривиальна: нужно разделить огромное число перебираемых вариантов на классы и получить оценки (снизу – в задаче минимизации, сверху – в задаче максимизации) для этих классов, чтобы иметь возможность отбрасывать варианты не по одному, а целыми классами. Трудность состоит в том, чтобы найти такое разделение на классы (ветви) и такие оценки (границы), чтобы процедура была эффективной.

Алгоритм Дейкстры

Одним из вариантов решения ЗК является вариант нахождения кратчайшей цепи, содержащей все города. Затем полученная цепь дополняется начальным городом – получается искомый тур.

В ориентированной, неориентированной или смешанной (т. е. такой, где часть дорог имеет одностороннее движение) сети найти кратчайший путь между двумя заданными вершинами.

Алгоритм использует три массива из n (= числу вершин сети) чисел каждый. Первый массив *a* содержит метки с двумя значениями: 0 (вершина ещё не рассмотрена) и 1 (вершина уже рассмотрена); второй массив *b* содержит расстояния – текущие кратчайшие расстояния от v_i до соответствующей вершины; третий массив *c* содержит номера вершин – *k*-й элемент c_k есть номер предпоследней вершины на текущем кратчайшем пути из v_i в v_k. Матрица расстояний D_{ik} задаёт длины дуг d_{ik}; если такой дуги нет, то d_{ik} присваивается большое число, равное «машинной бесконечности».

Математическая модель задачи

N – число городов.

$C_{i\,j}$, i, j=1,…,N – матрица затрат, где $C_{i\,j}$ – затраты на переход из i-го города в j-й.

$X_{i\,j}$ – матрица переходов с компонентами:

$X_{i\,j}$ = 1, если коммивояжер совершает переход из i-го города в j-й,

$X_{i\,j}$ = 0, если не совершает перехода,

где i, j = 1,...,N и i≠j.

Критерий:

$$F(X) = \sum_{i=1}^{N}\sum_{j=1}^{N} C_{ij} \cdot X_{ij} \to \min \qquad (1)$$

,

где C_{ij} – матрица стоимости переходов,

X_{ij} – матрица переходов, где x_{ij}=0, если переход совершен и x_{ij}=1 в противном случае

Ограничения:

$$\sum_{i=1}^{N} X_{ij} = 1,\ i = 1,...,N. \qquad (2)$$

$$\sum_{j=1}^{N} X_{ij} = 1,\ j = 1,...,N. \qquad (3)$$

$$U_i - U_j + N \cdot X_{i\,j} \le N - 1,\ i, j = 1,...,N,\ i \ne j. \qquad (4)$$

Условие (2) означает, что коммивояжер из каждого города выезжает только один раз; условие (3) – въезжает в каждый город только один раз; условие (4) обеспечивает замкнутость маршрута, содержащего N городов, и не содержащего замкнутых внутренних петель.

Полный перебор

Для решения задачи коммивояжера я применял метод полного перебора. Полный перебор практически применим только в задачах малого размера.

Задача коммивояжера с n городами требует при полном переборе рассмотрения (n-1)!/2 туров в симметричной задаче и (n-1)! туров в несимметричной, а факториал, как показано в таблице 1, растет очень быстро.

Асимметричная и симметричная задачи

В общем случае, асимметричная задача коммивояжёра отличается тем, что она моделируется ориентированным графом. Таким образом, следует также учитывать, в каком направлении находятся ребра.

В случае симметричной задачи все пары ребер между одними и теми же вершинами имеют одинаковую длину, то есть, для ребра (i,j) одинаковы длины $c_{i\,j} = c_{j\,i}$. В симметричном случае количество возможных маршрутов вдвое меньше асимметричного случая. Симметричная задача моделируется неориентированным графом.

Задача коммивояжёра в случае реальных городов может быть как симметричной (рис. 1), так и асимметричной в зависимости от длительности или длины маршрутов и от направления движения.

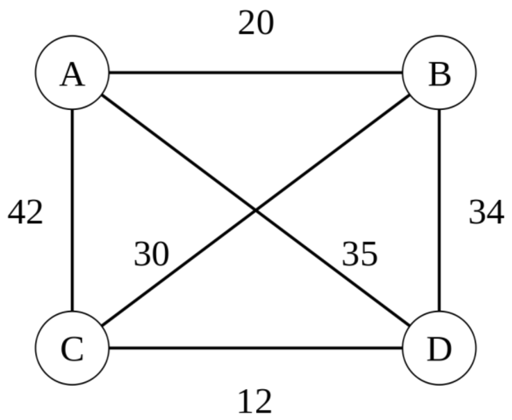

Рис. 1. Симметричная задача для четырёх городов

Таблица 1 – Примерные значения факториала

5!	10!	15!	20!	25!	30!	35!	40!	45!	50!
~10^2	~10^6	~10^{12}	~10^{18}	~10^{125}	~10^{32}	~10^{40}	~10^{47}	~10^{56}	~10^{64}

Чтобы проводить полный перебор в ЗК, необходимо без повторений генерировать все перестановки заданного числа m элементов. Это можно сделать несколькими способами, но самый распространенный (т.е. приложимый для переборных алгоритмов решения других задач) – это перебор в лексикографическом порядке.

Пусть имеется некоторый алфавит и наборы символов алфавита (букв), называемые словами. Буквы в алфавите упорядочены: например, в русском алфавите порядок букв а$\propto$б$\propto$я (символ $\propto$ читается «предшествует)». Если задан порядок букв, можно упорядочить и слова.

Пусть дано слово $u=(u_1,u_2,..,u_m)$, состоящее из букв $u_1,u_2,..,u_m$, и слово $v=(v_1,v_2,..,v_b)$. Тогда если $u_1 \propto v_{1,}$ то и $u \propto v$, если же $u_1=v_{1,}$ то сравнивают вторые буквы и т.д. Этот порядок слов и называется лексикографическим. Поэтому в русских словарях (лексиконах) слово «абажур» стоит раньше слова «абака». Слово «бур» стоит раньше слова «бура», потому что пробел считается предшествующим любой букве алфавита.

Рассмотрим перестановки из пяти элементов, обозначенных цифрами 1,...,5. Лексикографически первой перестановкой является 1-2-3-4-5, второй – 1-2-3-5-4, …, последней – 5-4-3-2-1.