Лекция 4. Задача коммивояжера

Одна из самых известных и важных задач транспортной логистики (и класса задач оптимизации в целом) — задача коммивояжера (англ. “Travelling salesman problem”, TSP). Также встречается название «задача о бродячем торговце».

Суть задачи сводится к поиску оптимального, то есть кратчайшего пути, проходящего через некие пункты по одному разу. Например, задача коммивояжера может применяться для нахождения самого выгодного маршрута, позволяющего коммивояжеру объехать определенные города со своим товаром по одному разу и вернуться в исходную точку. Мерой выгодности маршрута будет минимальное время, проведенное в пути, минимальные расходы на дорогу или, в простейшем случае, минимальная длина пути. Кто и когда впервые начал исследовать задачу коммивояжера – неизвестно, но одним из первых предложил решение подобной проблемы выдающийся математик XIX в. – Уильям Гамильтон. Здесь мы рассмотрим замкнутый вариант задачи (т.е. такой, когда в итоге мы возвращаемся в исходную точку) и ее решение методом ветвей и границ.

Для решения задачи коммивояжера методом ветвей и границ необходимо выполнить следующий алгоритм (последовательность действий):

1. Построение матрицы с исходными данными.
2. Нахождение минимума по строкам.
3. Редукция строк.
4. Нахождение минимума по столбцам.
5. Редукция столбцов.
6. Вычисление оценок нулевых клеток.
7. Редукция матрицы.
8. Если полный путь еще не найден, переходим к пункту 2, если найден – к пункту 9.
9. Вычисление итоговой длины пути и построение маршрута.

Более подробно эти этапы решения задачи о бродячем торговце раскрыты ниже.

В целях лучшего понимания задачи будем оперировать не понятиями графа, его вершин и т.д., а понятиями простыми и максимально приближенными к реальности: вершины графа будут называться “города”, ребра, их соединяющие — “дороги”.

Итак, методика решения задачи коммивояжера:

1. Построение матрицы с исходными данными

Сначала необходимо представить длины дорог, соединяющих города, в виде следующей таблицы:

Город	1	2	3	4
1	∞	5	11	9
2	10	∞	8	7
3	7	14	∞	8
4	12	6	15	∞

В нашем примере 4 города, и в таблице указано расстояние от каждого города до трех других, в зависимости от направления движения (т.к. некоторые ж/д пути могут быть с односторонним движением и т.д.). Расстояние от города до этого же города обозначено знаком ∞. Это сделано для того, чтобы данный отрезок путь был условно принят за бесконечно длинный. Тогда не будет смысла выбрать движение от 1-ого города к 1-му, от 2-ого ко 2-му, и т.п. в качестве отрезка маршрута.

2. Нахождение минимума по строкам

Находим минимальное значение в каждой строке (di) и выписываем его в отдельный столбец.

Город	1	2	3	4	di
1	∞	5	11	9	5
2	10	∞	8	7	7
3	7	14	∞	8	7
4	12	6	15	∞	6

3. Редукция строк. Производим редукцию строк — из каждого элемента в строке вычитаем соответствующее значение найденного минимума (di).

Город	1	2	3	4	di
1	∞	0	6	4	5
2	3	∞	1	0	7
3	0	7	∞	1	7
4	6	0	9	∞	6

В итоге в каждой строке будет хотя бы одна нулевая клетка.

4. Нахождение минимума по столбцам. Далее находим минимальные значения в каждом столбце (dj). Эти минимумы выписываем в отдельную строку.

Город	1	2	3	4	di
1	∞	0	6	4	5
2	3	∞	1	0	7
3	0	7	∞	1	7
4	6	0	9	∞	6
dj	0	0	1	0	

5. Редукция столбцов. Вычитаем из каждого элемента матрицы соответствующее ему dj.

Город	1	2	3	4	di
1	∞	0	5	4	5
2	3	∞	0	0	7
3	0	7	∞	1	7
4	6	0	8	∞	6
dj	0	0	1	0	

В итоге в каждом столбце будет хотя бы одна нулевая клетка.

6. Вычисление оценок нулевых клеток

Для каждой нулевой клетки получившейся преобразованной матрицы находим «оценку». Ею будет сумма минимального элемента по строке и минимального элемента по столбцу, в которых размещена данная нулевая клетка. Сама она при этом не учитывается. Найденные ранее di и dj не учитываются. Полученную оценку записываем рядом с нулем, в скобках.

Город	1	2	3	4
1	∞	0 (4)	5	4
2	3	∞	0	0
3	0	7	∞	1
4	6	0	8	∞

И так по всем нулевым клеткам:

Город	1	2	3	4
1	∞	0 (4)	5	4
2	3	∞	0 (5)	0 (1)
3	0 (4)	7	∞	1
4	6	0 (6)	8	∞

7. Редукция матрицы. Выбираем нулевую клетку с наибольшей оценкой. Мы нашли один из отрезков пути. Выписываем его (от какого города к какому движемся, в нашем примере от 4-го к 2-му). Ту строку и тот столбец, на пересечении которых находится нулевая клетка с наибольшей оценкой, полностью вычеркиваем. В клетку, соответствующую обратному пути, ставим еще один знак ∞ (т.к. мы уже не будем возвращаться обратно).

Город	1	3	4
1	∞	5	4
2	3	0 (5)	∞
3	0 (4)	∞	1

8. Если полный путь еще не найден, переходим к пункту 2, если найден – к пункту 9. Если мы еще не нашли все отрезки пути, то возвращаемся ко 2-му пункту и вновь ищем минимумы по строкам и столбцам, проводим их редукцию, считаем оценки нулевых клеток и т.д. Если все отрезки пути найдены (или найдены еще не все отрезки, но оставшаяся часть пути очевидна) – переходим к пункту 9.

9. Вычисление итоговой длины пути и построение маршрута

Найдя все отрезки пути, остается только соединить их между собой и рассчитать общую длину пути (стоимость поездки по этому маршруту, затраченное время и т.д.). Длины дорог, соединяющих города, берем из самой первой таблицы с исходными данными. В нашем примере маршрут получился следующий: $4 \to 2 \to 3 \to 1 \to 4$. Общая длина пути: $L = 30$.

Применение задачи коммивояжера на практике довольно обширно. В частности, ее можно использовать для поиска кратчайшего маршрута при гастролях эстрадной группы по городам, нахождения последовательности технологических операций обеспечивающей наименьшее время выполнения всего производственного цикла и пр.

Источник: Галяутдинов Р.Р. Задача коммивояжера - метод ветвей и границ // Сайт преподавателя экономики. [2013]. URL: http://galyautdinov.ru/post/zadacha-kommivoyazhera (дата обращения: 11.03.2018).