

# 1 Задача коммивояжера. Постановка, математическая модель. Общая идея решения методом ветвей и границ. Приведенная матрица.

## 1.1 Общая идея методов ветвей и границ

Задача:

$$f(x) \rightarrow \min_{x \in X}$$

1. В зависимости от специфики задачи выбирается некоторый способ вычисления оценок снизу  $d(X')$  функции  $f(x)$  на множествах  $X' \subset X$ : (в частности, может быть  $X' = X$ )

$$f(x) \geq d(X'), x \in X'.$$

Оценка снизу часто вычисляется путем релаксации, т.е. замены задачи минимизации  $f(x)$  по множеству  $X'$  задачей минимизации по некоторому более широкому множеству. (Например, релаксация целочисленной или частично целочисленной задачи может состоять в отбрасывании требования целочисленности.)

2. Выбирается также правило ветвления, состоящее в выборе разветвляемого подмножества  $X'$  из числа подмножеств, на которые к данному шагу разбито множество  $X$ , и выборе способа разбиения  $X'$  на непересекающиеся подмножества.

Обычно из числа кандидатов на ветвление выбирается множество  $X'$  с наименьшей оценкой, поскольку именно в таком множестве естественно искать минимум в первую очередь.

При этом рассматриваются только такие способы вычисления оценок снизу, в которых оценки для подмножеств, получившихся в результате разветвления  $X'$ , не меньше  $d(X')$ .

## Метод ветвей и границ решения задачи коммивояжера

Дано  $n$  городов,  $C = |c_{i,j}|$ ,  $i, j = \overline{1, n}$  - матрица стоимостей переездов из  $i$ -х городов в  $j$ -е. Коммивояжер должен выехать из своего города, заехать в каждый город только один раз и вернуться в исходный город. Нужно найти

замкнутый маршрут объезда всех городов минимальной стоимости.

$$x_{i,j} = \begin{cases} 1, & \text{если коммивояжер едет из } i \text{ в } j \\ 0, & \text{иначе} \end{cases}$$

$$F = \sum_{i=1}^n \sum_{j=1}^n c_{i,j} \cdot x_{i,j} \rightarrow \min$$

$$\sum_{i=1}^n x_{i,j} = 1 \quad \forall j = \overline{1, n}, \quad \sum_{j=1}^n x_{i,j} = 1 \quad \forall i = \overline{1, n}$$

Из каждого  $i$ -го города только один выезд, в каждый  $j$ -й город, только один въезд.

$u_i \in \overline{1, n}$  каким по счету мы посетим город  $i$ .

$u_i - u_j + n \cdot x_{i,j} \leq n - 1$  обеспечивает замкнутость маршрута и отсутствие петель

## 1.2 Постановка задачи

1. Покажем, что любой набор переменных  $u_i, x_{ij}$ , удовлетворяющий ограничениям этой ЦЛП, определяет допустимый маршрут коммивояжеров. Пусть некоторый допустимый набор переменных определяет маршрут, распадающийся на не связанные между собой подциклы. Возьмем любой подцикл. Сложим ограничения неравенства, соответствующие переездам, входящим в этот подцикл. **Например:** 1-2-3-1

$$u_1 - u_2 + n \leq n - 1$$

$$u_2 - u_3 + n \leq n - 1$$

$$u_3 - u_1 + n \leq n - 1$$

Складываем уравнения  $\Rightarrow 3n \leq 3(n - 1)$  — неверное неравенство. Итак, разности  $u_i - u_j$  взаимно уничтожаются, и получается неверное неравенство:

$$kn \leq k(n - 1), \quad \text{где } k \text{ — число переездов.}$$

2. Покажем, что произвольному допустимому маршруту коммивояжера соответствует некоторый набор переменных, удовлетворяющих ограничению задачи. Положим  $u_i = p$ , если коммивояжер на маршруте прибывает в  $i$ -ый город после  $p$ -го переезда.  $\Rightarrow u_i - u_j \leq n - 1 \quad \forall i, j$ , при  $x_{ij} = 0$

$$(p - (p + 1)) \leq n - 1$$

$$u_i - u_j + nx_{ij} = p - (p + 1) + n = n - 1 \Rightarrow \text{Задача ЛП — правильная.}$$

**Определение 1.** Циклом  $t$  назовем набор из  $n$  упорядоченных пар городов, образующих маршрут, проходящий через каждый город только один раз.

$$t = \{(i_1, i_2), (i_2, i_3), \dots, (i_{n-1}, i_n), (i_n, i_1)\}$$

$$z(t) = \sum_{k=1}^{n-1} c_{i_k, i_{k+1}} + c_{i_n, i_1}$$

Для каждого допустимого маршрута каждая строка и каждый столбец содержат ровно по одному элементу, соответствующему этому маршруту.

$$C = \begin{pmatrix} \infty & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{n1} & \cdots & \infty \end{pmatrix}$$

**Определение 2. Определение:** Матрица, получаемая из данной вычитанием из элементов каждой строки минимального элемента этой строки, а затем вычитанием из элементов каждого столбца минимального элемента этого столбца, называется приведенной матрицей:

$$\begin{cases} |c_{i,j}| = \left| c_{i,j} - \min_{i=\overline{1,n}} c_{i,j} \right| \\ |c''_{i,j}| = \left| c'_{i,j} - \min_{j=\overline{1,n}} c'_{i,j} \right| \end{cases}$$

Приводящая константа:

$$h = \sum_{i=1}^n \min_{j=\overline{1,n}} c_{ij} + \sum_{j=1}^n \min_{i=\overline{1,n}} c'_{ij}$$

$h$  является нижней границей издержек для всех циклов  $t$  исходной матрицы расстояний, поскольку  $h$  — сумма минимальных элементов строк и столбцов.

## Алгоритм

### 1. Приведение матрицы

## 2. Ветвление

На каждом шаге алгоритма будет строиться одно звено оптимального маршрута. Для построения решения в начале целесообразно выбрать звено нулевой длины, а затем последовательно добавлять звенья нулевой или минимальной длины.

Процесс ветвления можно представить в виде дерева, каждая вершина которого соответствует некоторому множеству маршрутов, являющегося подмножеством множества  $X$ .

Пусть  $G_0$  - множество всех маршрутов. Разобьем  $G_0$  на два подмножества:

- Множество маршрутов, включающих переезд из  $i$  в  $j$ . Обозначение:  $\{(i, j)\}$
- Множество маршрутов, не включающих переезд из  $i$  в  $j$ . Обозначение:  $\{\overline{(i, j)}\}$

Пару городов  $(i, j)$  для ветвления будем выбирать среди тех пар, которым в приведенной матрице соответствуют нулевые элементы, причем выбирается такая пара  $(i, j)$ , чтобы подмножество  $\{\overline{(i, j)}\}$  имело максимальную оценку.

Разветвляя далее множество с меньшей оценкой, в конце концов будет получено подмножество, содержащее один маршрут. Двигаясь по дереву в обратном направлении, получим маршрут.

Выбор пары городов  $(i, j)$  для ветвления:

- $c''_{i,j} = 0$ .
- Оценка множества  $\{\overline{(i, j)}\}$  должна быть максимальной. Рассмотрим маршруты, которые будут включены в  $\{\overline{(i, j)}\}$ .

Поскольку город  $i$  должен быть связан с некоторым другим городом, то каждый маршрут из  $\{\overline{(i, j)}\}$  должен содержать звено, длина которого не меньше минимального элемента  $i$ -ой строки, не считая  $c''_{i,j}$ .

Вычисляем сумму этих минимальных элементов для каждого  $c''_{i,j} = 0$ . Назовем эту сумму оценкой пары  $(i, j)$  или штрафом за исполь-

зование звена  $(i, j)$ :

$$\Theta_{i,j} = \min_{\forall j' \neq j} c''_{i,j'} + \min_{\forall i' \neq i} c''_{i',j}$$

В качестве пары городов для ветвления выбираем ту пару, для которой оценка будет максимальной.

### 3. Вычисление нижней границы множества $\{\overline{(i, j)}\}$

Нижняя граница множества  $\{\overline{(i, j)}\}$  определяется как сумма оценки разветвляемого множества и максимального значения  $\Theta_{i,j}$ .

### 4. Исключение строки и столбца.

Так как из каждого города можно выезжать только один раз и в каждый город можно въезжать только один раз, то строку  $i$  и столбец  $j$  из дальнейшего рассмотрения исключаем. Чтобы не получить замкнутых неполных циклов, нужно наложить необходимые запреты, в частности, на переезд из  $j$  в  $i$ , т.е. положить  $C_{i,j} = \infty$ .

### 5. Завершение

Если усеченная матрица ещё не имеет размерности  $2 \times 2$ , то приводим полученную матрицу и находим оценку множества  $\{(i, j)\}$  как сумму оценки разветвляемого множества и полученной приводящей константы, переход к пункту 2.

Иначе

определяемые усеченной матрицы пары городов завершают маршрут. Приводя эту матрицу и добавляя приводящую константу к оценке последнего разветвляемого множества, получим оценку маршрута.

### Критерий оптимальности:

Если оценка не больше оценок всех тупиковых ветвей, то маршрут, описанный деревом ветвей, является оптимальным. Иначе процесс ветвления должен быть продолжен, исходя из множества с меньшей оценкой.