

1 Задачи целочисленного линейного программирования. Задача о назначениях. Постановка, математическая модель. Венгерский алгоритм решения задачи о назначениях.

1.1 Постановка задачи

Пусть имеется n видов ресурсов, которые нужно распределить на n объектов. C_{ij} — затраты (выигрыш или прибыль), связанные с назначением i -го ресурса на j -ый объект. Предполагается, что каждый ресурс назначается ровно один раз и каждому объекту приписывается ровно один ресурс. **Требуется:** Минимизировать стоимость назначений.

1.2 Определение неизвестных

Определим неизвестные X_{ij} :

$$X_{ij} = \begin{cases} 1, & \text{если } i\text{-ый ресурс назначается на } j\text{-ый объект,} \\ 0, & \text{иначе} \end{cases}$$

$$\sum_{i=1}^n \sum_{j=1}^n C_{ij} X_{ij} \rightarrow \min(\max)$$

$$\begin{aligned} \sum_{i=1}^n X_{ij} &= 1, & j = \overline{1, n} & - \text{на каждый } j\text{-ый объект — один вид ресурса.} \\ \sum_{j=1}^n X_{ij} &= 1, & i = \overline{1, n} & - \text{каждый } i\text{-ый ресурс на один объект.} \end{aligned}$$

Определение 1. Допустимое решение называется назначением.

Допустимое решение строится путем выбора ровно одного элемента в каждой строке матрицы $X = [X_{ij}]$ и ровно одного элемента в каждом столбце этой матрицы.

Пример

$$C = \begin{pmatrix} 4 & 7 & 0 \\ 0 & 3 & 8 \\ 6 & 3 & 9 \end{pmatrix} \quad X = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

C — соответствующие затраты, X — допустимое решение.

Доказательство. Рассмотрим задачу о назначениях с матрицей стоимостей $C = [C_{ij}]$.

Предположим, что каждый элемент i -ой строки складывается с действительным числом γ_i , а каждый элемент j -го столбца — с действительным числом δ_j .

В результате будет получена новая матрица стоимостей D :

$$D_{ij} = C_{ij} + \gamma_i + \delta_j$$

Покажем, что задача минимизации функции

$$\sum_i \sum_j c_{ij} x_{ij}$$

эквивалентна минимизации функции

$$\sum_i \sum_j d_{ij} x_{ij}$$

(т.е. такое преобразование не меняет точку минимума целевой функции).

$$d_{ij} = c_{ij} + \gamma_i + \delta_j$$

$$c_{ij} = d_{ij} - \gamma_i - \delta_j$$

$$c_{ij} X_{ij} = d_{ij} X_{ij} - \gamma_i X_{ij} - \delta_j X_{ij}$$

$$\begin{aligned} \sum_i \sum_j c_{ij} X_{ij} &= \sum_i \sum_j d_{ij} X_{ij} - \sum_i \sum_j \gamma_i X_{ij} - \sum_i \sum_j \delta_j X_{ij} = \\ &= \sum_i \sum_j d_{ij} X_{ij} - \sum_i \gamma_i - \sum_j \delta_j \end{aligned}$$

□

1.2.1 Идея венгерского алгоритма

Из элементов каждой строки и каждого столбца матрицы стоимостей вычитаются их наименьшие элементы, после чего ведется поиск допустимого решения, единичным элементам которого соответствуют нулевые элементы модифицированной матрицы стоимостей.

Если такое допустимое решение существует, то оно является оптимальным назначением.

Иначе матрица стоимостей модифицируется еще раз с целью получить в ней большее число нулевых элементов.

Алгоритм состоит из трех шагов:

1. Редукция строк и столбцов

Из каждого элемента строки вычитается ее \min элемент.

Из каждого элемента столбца вычитается его \min элемент.

Цель шага – получить в каждой строке и столбце хотя бы один нулевой элемент.

2. Построение назначения Если в каждой строке и в каждом столбце матрицы стоимостей можно выбрать по одному нулевому элементу \Rightarrow соответствующее допустимое решение будет оптимальным, иначе goto п. 3.
- а) Рассмотреть строки в порядке возрастания их номеров. Найти строки, содержащие ровно один не вычеркнутый нулевой элемент. В каждой такой строке произвести назначение, соответствующее не вычеркнутому нулевому элементу. В каждом столбце, в котором было произведено назначение, вычеркнуть все не вычеркнутые ранее нулевые элементы.
- б) Рассмотреть столбцы в порядке возрастания их номеров. Найти столбцы, содержащие ровно один не вычеркнутый нулевой элемент. В каждом таком столбце произвести назначение, соответствующее этому нулевому элементу. В каждой строке, в которой было произведено назначение, вычеркнуть все не вычеркнутые ранее нулевые элементы.
- в) Выполнять а), б) до тех пор, пока не будет вычеркнуто \max возможное число нулей. Если построенное назначение полное \Rightarrow оно оптимально, иначе goto п. 3.

3. Модификация матрицы стоимостей

Определим для редуцированной матрицы стоимостей минимальное множество строк и столбцов, содержащих все нулевые элементы, и найдем минимальный элемент вне данного множества.

Если значение данного элемента вычесть из всех остальных элементов матрицы, то на месте нулей будут стоять отрицательные величины и, по крайней мере, один элемент, не принадлежащий выделенному множеству строк и столбцов, станет равным нулю. Однако, теперь назначение нулевой стоимости может не быть оптимальным, поскольку матрица содержит отрицательные элементы. Для того, чтобы матрица не содержала отрицательных элементов, прибавим абсолютную величину наименьшего отрицательного элемента ко всем элементам выделенных строк и столбцов.

Тогда к элементам расположенным на пересечении выделенных строк и столбцов, данная величина будет прибавляться дважды. Кроме того, как и раньше, все отрицательные элементы будут преобразованы в нулевые или положительные элементы.

⇒ Новая матрица стала содержать больше нулей, расположенных вне строк и столбцов, соответствующих нулевым элементам текущего неоптимального решения.

а) Вычислить число 0-ей в каждой не вычеркнутой строке и каждом не вычеркнутом столбце.

б) Вычеркнуть строку или столбец с max числом нулей. В случае равенства числа 0 в нескольких строках и столбцах вычеркнуть любую из этих строк (или любую из столбцов).

в) Выполнять а), б) до тех пор, пока не будут вычеркнуты все нули.

г) Из всех не вычеркнутых элементов вычесть \min не вычеркнутый элемент и прибавить его к каждому элементу, расположенному на пересечении двух линий.