Венгерский алгоритм: решение задачи назначения

Постановка задачи

Дана квадратная матрица стоимости $C = (c_{ij})$ размера $n \times n$, где c_{ij} — стоимость выполнения задачи j исполнителем i. Требуется найти биекцию $\sigma : \{1, \ldots, n\} \to \{1, \ldots, n\}$, минимизирующую

$$\sum_{i=1}^{n} c_{i,\sigma(i)}.$$

Алгоритм

Алгоритм состоит из пяти этапов:

1. **Нормализация строк:** для каждой строки i найдём $m_i = \min_j c_{ij}$ и вычтем из строки: $c_{ij} \leftarrow c_{ij} - m_i$. Теперь в каждой строке есть хотя бы один нуль.

2. **Нормализация столбцов:** для каждого столбца j найдём $m'_j = \min_i c_{ij}$ и вычтем: $c_{ij} \leftarrow c_{ij} - m'_j$. В итоге в каждой строке и столбце как минимум один нуль.

3. **Покрытие нулей и построение максимального паросочетания:**

   - Построим двудольный граф $G = (U, V, E)$, где U соответствует строкам, V — столбцам. Рёбра E соединяют $i \in U$ и $j \in V$ тогда и только тогда, когда $c_{ij} = 0$.
   - Найдём *максимальное паросочетание* в этом графе (например, с помощью алгоритма Хопкрофта–Карпа). Пусть размер максимального паросочетания равен $\nu(G)$.
   - По теореме Кёнига – Эгерсона для двудольных графов размер максимального паросочетания $\nu(G)$ равен размеру минимального вершинного покрытия $\tau(G)$. Вершины покрытия соответствуют строкам и столбцам, покрывающим все рёбра (нули). Количество таких линий равно $\tau(G) = \nu(G)$.
   - Если $\nu(G) = n$, то среди нулей найдена система независимых рёбер, покрывающая все строки. Это оптимальное назначение. Переходим к шагу 5.

- Иначе ($\nu(G) < n$) число линий меньше n, и требуется перейти к модификации матрицы.

4. **Модификация матрицы:** Пусть R и C — множества строк и столбцов, входящие в найденное вершинное покрытие размера $\nu(G) < n$. Найдём
$$\Delta = \min\{c_{ij} \mid i \notin R,\ j \notin C\} > 0\}.$$
Вычтем Δ из всех непокрытых элементов ($i \notin R, j \notin C$) и прибавим Δ к элементам, покрытым двумя линиями ($i \in R, j \in C$). Новые нули появляются на непокрытых позициях. Вернёмся к шагу 3.

5. **Построение назначения:** когда $\nu(G) = n$, максимальное паросочетание даёт n независимых рёбер $(i, \sigma(i))$, что и есть оптимальное решение исходной задачи.

Сложность Венгерского алгоритма

Венгерский алгоритм решает задачу назначения за время $O(n^3)$, где n — размерность квадратной матрицы. Основными причинами такой сложности являются:

- Нормализация строк и столбцов занимает $O(n^2)$.
- На каждом шаге поиска покрытия и модификации выполняется поиск максимального паросочетания (алгоритм Хопкрофта–Карпа) за $O(n^{2.5})$ и возможная корректировка матрицы за $O(n^2)$.
- В худшем случае число итераций покрытия–модификации составляет $O(n)$.

Примечание

Алгоритм Хопкрофта–Карпа Это эффективный алгоритм для поиска максимального паросочетания в двудольном графе $G = (U, V, E)$. Идея состоит в повторном поиске наборов максимально множества минимальных увеличивающих путей. Основные шаги:

(a) **Построение слоёв (BFS):** от всех свободных вершин в U строится слоёный ориентированный граф до первых встреченных свободных вершин в V по рёбрам, не принадлежащим текущему паросочетанию. Полученные слои организуют уровни вершин по расстоянию (в рёбрах) от множества свободных вершин.

(b) **Поиск увеличивающих путей (DFS):** из каждого свободного узла U по уровням строятся не пересекающиеся увеличивающие пути к свободным вершинам V. При обнаружении такого пути выполняется инверсия принадлежности рёбер — добавляем рёбра пути в паросочетание, удаляем те, что его пересекают.

(c) **Итерация:** повторяем BFS+DFS, пока существуют увеличивающие пути. Алгоритм завершается, когда не найден ни один новый увеличивающий путь.

(d) **Сложность:** за $O(E\sqrt{V})$, где E — число рёбер, $V = |U| + |V|$ — число вершин.