

Notes

Kopnov Alexandr

19 марта 2023 г.

Содержание

1	Постановка	1
2	Представление данных	2
2.1	Ввод из файла	2
2.2	Предобработка	2
3	Метод северо-западного угла	3
4	Метод потенциалов	3
4.1	Проверка оптимальности	3
4.2	Цикл пересчёта	4
4.3	Построение следующего приближения.	4

1 Постановка

Решить транспортную задачу

Транспортная таблица:

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

Все задачи сбалансированы: объём хранимого груза совпадает с требованиями

Задания:

1. Решить методом потенциалов с выбором начального плана методом С-З угла

2. Решить симплекс-методом
3. Внести дизбаланс в исходные данные: уменьшить количества, чтобы было доступно меньше, чем требуется
4. Предусмотреть наличие штрафа за недопоставку, который может различаться у разных потребителей.

а) Штраф за недопоставку меняется в зависимости от объёма

2 Представление данных

Матрица коэффициентов, вектор запасов поставщиков, вектор требований потребителей, матрица поставок. Заполненные (возможно нулём) клетки имеют значения объёма поставок, незаполненные — -1

2.1 Ввод из файла

1. В первой строке размерность задачи: кол-во поставщиков и кол-во потребителей (m, n)
2. m строк по $n + 1$ разделённых запятой чисел: коэффициенты матрицы и запасы поставщика.
3. n значений, означающих требования потребителей.
4. опционально n значений, означающих штрафы за недопоставку

Поставленная задача будет записана как

```
4,5
8,2,5,3,14,8
10,4,5,7,15,3
5,1,2,1,10,11
6,3,2,4,15,5
4,5,4,9,5
```

2.2 Предобработка

Проверяется закрытость задачи. Если задача является открытой, т.е. $\sum_{i=1}^m a_i - \sum_{j=1}^n b_j \neq 0$, вводится фиктивный потребитель или фиктивный поставщик (в зависимости от знака разности)

При наличии штрафа за недопоставку коэффициенты цены фиктивного поставщика не 0, а равны штрафу за недопоставку единицы продукта.

3 Метод северо-западного угла

Используется для построения начального вектора.

k -я итерация:

1. $x_k = \min\{a[i], b[j]\}$, $a[i]- = x_k$, $b[j]- = x_k$, сохраняем пару $\{(i, j), x_k\}$
2. Если $a_i = 0$, $i+ = 1$, переход на следующую итерацию
3. Если $b_j = 0$, $j+ = 1$, переход на следующую итерацию.

Такая реализация «обходит» диагональные переходы: за счёт того, что условия проверяются последовательно и после каждого происходит переход, если $a_i = 0$ и $b_j = 0$, сначала увеличится i , потом j . Т.о. диагональный переход всегда заменяется на переход вниз и переход направо.

В начале алгоритма $k = 0$, $i = 0$, $j = 0$ и он завершается, когда $k = n + m - 2$.

4 Метод потенциалов

4.1 Проверка оптимальности

Используем вспомогательные вектора: $v[N], u[M]$. $u[0] = 0$.

Каждой ячейке матрицы поставок с неотрицательным значением можем сопоставить ограничение:

$$v[j] - u[i] = c_{ij}$$

Начиная с ячеек, в которых $i = 0$, определяем значения u, v , перемещаясь по зап. ячейкам, в которых совпадает одна из координат. Это реализовано с помощью стека, в котором хранятся координаты ячейки и направление, которое надо проверить.

Так как матрица значений поставок хранится отдельно и не влияет на условие оптимальности, и для контроля ошибок, проверяются все элементы матрицы.

$$\Delta_{ij} = c_{ij} - v_j + u_i$$

В цикле проверяем существование $\Delta_{ij} < 0$ и запоминаем координаты ячейки в которой Δ имеет наибольший модуль.

Если существуют $\Delta_{ij} < 0$, строим цикл пересчёта.

4.2 Цикл пересчёта

Начиная с клетки, координаты которой были найдены на прошлом шаге выполняем поиск с возвратом.

Первая итерация происходит отдельно, так как есть только одна начальная клетка. На последующих итерациях принимается список координат с которых можно продолжить поиск, список координат, входящих в цикл, координаты конечной клетки и направление поиска.

Для каждой пары координат:

1. Если поиск идёт по строке и строка совпадает со строкой, в которой находится конечная клетка, либо поиск идёт по столбцу и совпадает столбец, координаты добавляются в список координат, входящих в цикл. Функция завершается, возвращая значение **true**
2. Строится список кандидатов в заданном направлении: если координаты отличны от начальных и значение в таблице перевозок неотрицательно, координаты добавляются в список кандидатов.
3. Вызов функции с новым списком координат и другим направлением. Если функция вернула **true**, координаты добавляются в цикл и этот вызов также возвращает **true**.
4. Переход к следующей паре координат.

Если ни одна из пар координат не вернула **true**, функция возвращает значение **false**.

Таким образом, функция вернёт **false** тогда и только тогда, когда не существует такого продолжения, которое приводит к замыканию цикла.

Начальный вызов всегда будет возвращать **true**, так как цикл всегда существует.

4.3 Построение следующего приближения.

Начиная со второго элемента найденного цикла проверяем значения через одно. Минимальное из них — θ . Затем:

1. Присваиваем клетке начала цикла значение θ .
2. Начиная со второго элемента цикла $x_k + = \theta \cdot \text{sign}$, $\text{sign} = -1$
3. Элемент, значение которого было выбрано как θ заменяется -1