Notes

Kopnov Alexandr

17 марта 2023 г.

Содержание

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.
    a) Штраф за недопоставку меняется в зависимости от объёма

2 Представление данных

Матрица коэфициентов, вектор запасов поставщиков, вектор требований потребителей.

Известно, что если размерность задачи m, n, будет $m + n - 1$ элементов в векторе решения. Таким образом, хранится $m + n - 1$ пар, состоящих из координат ячейки и значения в ней.

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: коэфициенты 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. $\sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j != 0$, вводится фиктивный потребитель или фиктивный поставщик (в зависимости от знака разности)

При наличии штрафа за недопоставку коэфициенты 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 налево.

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 паре $\{(i,j), x_k\}$ можем сопоставить ограничение:

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

Начиная с пары, в которой $i = 0$, определяем значения u, v, перемещаясь по парам в которых совпадает одна из координат.

Так как вектор решения хранится отдельно и не влияет на условие оптимальности, быстрее проверить 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.

Затем строим цикл пересчёта.

4.2 Цикл пересчёта

Начиная с клетки, координаты которой были найдены на прошлом шаге выполняем поиск с возвратом. Рассмотрим одну итерацию:

Функция принимает координаты начальной клетки, вектор заполненных клеток, текущее направление и текущую клетку. На первой итерации ищем по вертикали и текущая клетка является начальной.

1. Находим все заполненные клетки в заданном направлении.
2. Из этих клеток выбираем те, которые позволяют сделать по крайней мере ещё один шаг, т.е. ища в другом направлении найдём по крайней мере одну клетку.
3. Переходим на следующую итерацию, используя другое направление и одну из найденных на шаге 2 клеток. Если поиск заходит в тупик, т.е. на шаге 2 клеток не найдено, возвращаемся и пробуем другую клетку.