

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра РАПС

ОТЧЕТ
по практической работе № 5
по дисциплине «Теория принятия решений»
ТЕМА: ТРАНСПОРТНАЯ ЗАДАЧА ЛИНЕЙНОГО
ПРОГРАММИРОВАНИЯ С ОТКРЫТОЙ МОДЕЛЬЮ
Вариант 1

Студент гр. 9492

Викторов А.Д.

Преподаватель

Белов А.М.

Санкт-Петербург

2023

Дана следующая матрица тарифов и значения объема грузов:

14	5	27	29	23	18
17	7	16	19	2	14
20	12	15	29	5	16
14	24	18	7	14	22
8	11	11	9	21	

Составим таблицу

Таблица 1

	B ₁	B ₂	B ₃	B ₄	B ₅	a _i
A ₁	14	5	27	29	23	18
A ₂	17	7	16	19	2	14
A ₃	20	12	15	29	5	16
A ₄	14	24	18	7	14	22
b _j	8	11	11	9	21	

Так как сумма поставок и сумма потребностей различается делаем вывод, что данная транспортная задача обладает открытой моделью.

Для решения этой и подобных задач была разработана программа в среде Matlab, реализующая следующий алгоритм:

0. Изменение топологии матриц для формирования закрытой модели.
1. Построить опорный план по одному из правил: метод северо-западного угла, метод минимального элемента.
2. Вычислить потенциалы поставщиков и потребителей U_i и V_j , решив систему уравнений вида $U_i + V_j = c_{ij}$ для занятых клеток.
3. Вычислить оценки S_{ij} для всех свободных клеток по формуле:
$$S_{ij} = C_{ij} - (U_i + V_j).$$

Если все $S_{ij} \geq 0$, то полученный план – оптимальный, при этом если все $S_{ij} > 0$, то этот план единственный.

Если хотя бы одна оценка $S_{ij} = 0$, имеем бесчисленное множество оптимальных планов с одним и тем же значением целевой функции.
4. Если хотя бы одна оценка $S_{ij} < 0$, то план неоптимальный. Переходим к другому плану. Для этого выбираем $\min\{S_{ij} < 0\}$ и эта соответствующая клетка

будет перспективной. Строим для нее цикл. Получаем новый план. Для нового плана находим потенциалы и т. д.

Код программы представлен в листинге 1.

Листинг 1 – Исходный код программы.

```
clc, clear
% Исходные данные
A0 = [18; 14; 16; 22]; % Запасы на базах A1, A2, A3
B0 = [8; 11; 11; 9; 21]; % Потребности в пунктах B1, B2, B3, B4, B5
C0 = [14, 5, 27, 29, 23;
      17, 7, 16, 19, 2;
      20, 12, 15, 29, 5;
      14, 24, 18, 7, 14]; % Матрица тарифов

if sum(A0) == sum(B0)
    disp("closed")
elseif sum(A0) > sum(B0)
    disp("open")
    B0 = [B0; sum(A0) - sum(B0)];
    C0 = [C0 zeros(size(A0,1),1)];
elseif sum(A0) < sum(B0)
    % error
end

A = A0;
B = B0;
C = C0;

% Создание пустой матрицы для плана перевозок
plan = zeros(size(A, 1), size(B, 1));
totalCost = 0; % Переменная для хранения общей стоимости перевозок

% Цикл для заполнения плана перевозок методом минимального элемента
while any(A) && any(B)
    [minCost, minIndex] = min(C(:)); % Находим минимальный тариф
    [i, j] = ind2sub(size(C), minIndex); % Получаем индексы минимального
    элемcostсента

    % Вычисляем количество груза для перевозки
    amount = min(A(i), B(j));
    plan(i, j) = amount; % Заполняем ячейку плана перевозок

    % Обновляем остаток груза на базе и в пункте назначения
    A(i) = A(i) - amount;
    B(j) = B(j) - amount;

    % Подсчет стоимости для текущей перевозки
    totalCost = totalCost + amount * C(i, j);

    % Помечаем использованный тариф как бесконечность
    C(i, j) = inf;
end

disp('План перевозок методом минимального элемента:');
```

Листинг 1 – Продолжение.

```

disp(plan);

fprintf('Общая стоимость перевозок: %d\n', totalCost); % Вывод общей стоимости
перевозок

A = A0;
B = B0;
C = C0;

% Инициализация переменных
m = numel(A);
n = numel(B);
U = zeros(m, 1); % Потенциалы для баз
V = zeros(n, 1); % Потенциалы для пунктов
totalCost = 0; % Переменная для хранения общей стоимости перевозок
optimalPlan = zeros(m, n); % Пустая матрица для оптимального плана перевозок

% Основной цикл метода потенциалов
while true
    % Поиск потенциалов
    for i = 1:m
        for j = 1:n
            if A(i) > 0 && B(j) > 0
                V(j) = C(i, j) - U(i);
            end
        end
    end

    for j = 1:n
        for i = 1:m
            if A(i) > 0 && B(j) > 0
                U(i) = C(i, j) - V(j);
            end
        end
    end

    % Нахождение минимальной оценки
    S = inf; % оценка
    for i = 1:m
        for j = 1:n
            if A(i) > 0 && B(j) > 0
                S = min(S, C(i, j) - U(i) - V(j));
            end
        end
    end

    % Обновление плана перевозок и подсчет общей стоимости
    for i = 1:m
        for j = 1:n
            if A(i) > 0 && B(j) > 0
                if C(i, j) - U(i) - V(j) == S
                    amount = min(A(i), B(j));
                    A(i) = A(i) - amount;
                    B(j) = B(j) - amount;
                    totalCost = totalCost + amount * C(i, j); % Подсчет стоимости для
текущей перевозки
                    optimalPlan(i, j) = amount; % Заполнение оптимального плана
перевозок

```

Листинг 1 – Окончание.

```

                                end
                            end
                        end
                    end

% Если все запасы и потребности выполнены, выход из цикла
if S == inf
    break;
end
end

disp('Оптимальный план перевозок:');
disp(optimalPlan); % Вывод оптимального плана перевозок
fprintf('Общая стоимость перевозок: %d\n', totalCost); % Вывод общей стоимости
перевозок после завершения алгоритма

```

В свою очередь в программе предусмотрено определение типа задачи (открытая или закрытая). В случае открытой задачи, матрицы автоматически модифицируются, модель получается закрытой по методологии из учебного пособия.

Результат работы программы представлен в листинге 2

Листинг 2 – Результат работы программы.

```

type: open
План перевозок методом минимального элемента:
    0      8      0      0      0     10
    0      0      0      0     14      0
    0      3      6      0      7      0
    8      0      5      9      0      0

Общая стоимость перевозок: 494
Оптимальный план перевозок:
    7     11      0      0      0      0
    0      0      0      0     14      0
    0      0      9      0      7      0
    1      0      2      9      0     10

Общая стоимость перевозок: 464

```