

# 1. Проблема

## Задача 5

### О распределении

Требуется распределить 4 единицы исходного ресурса между 4 однотипными агрегатами. Объем получаемой прибыли для каждого агрегата в зависимости от объема переработанных ресурсов представлен в таблице:

| № агрегата | Получаемая прибыль при заданном объеме ресурсов |     |    |    |
|------------|---|-----|----|----|
|            | 1   | 2   | 3  | 4  |
| 1          | 5   | 7,5 | 9  | 11 |
| 2          | 4   | 5,5 | 8  | 10 |
| 3          | 6   | 8   | 10 | 12 |
| 4          | 4,5   | 7   | 9  | 11 |

## 2. Содержательная постановка

1. Составить план работы агрегатов для получения максимальной прибыли, используя приведенную таблицу объема получаемой прибыли.
2. Посчитать максимальную прибыль.

## 3. Формальная постановка

Пусть

- $X = x_1, x_2, x_3$  - двумерное пространство использования заданных объемов.
- $C = c_1, c_2, c_3$  - двумерное пространство прибыли при заданном объеме ресурсов.

Задача:

$$(C, X) = \sum_{i=1}^4 \sum_{j=1}^4 c_{ij} x_{ij} \rightarrow \max$$

Ограничения нормированности по ГОСТу (условия):

$$\left\{ \begin{array}{l} x \geq 0 \\ x \leq 1 \\ \forall i : \sum_{j=0}^4 x_{ij} = 1 \\ \forall j : \sum_{i=0}^4 x_{ij} = 1 \end{array} \right.$$

## 4. Алгоритм и ПО

В качестве ПО будем использовать ЯП **Python** с подключенными модулями:

- **numpy** - для работы с линейной алгеброй
- **cvxpy** - для работы с линейным программированием

В качестве среды разработки **Jupyter Lab**,

## 5. Решение

Приведем в качестве решения скрипт:

```
In [1]: def profit(c, accuracy=1):  
    '''  
    Функция, подсчитывающая объем получаемой прибыли  
    при использовании оптимального распределения  
    работы агрегатов, а также план их работы  
  
    c - 2D-список  
    получаемой прибыли при  
    заданном объеме ресурсов  
    для агрегатов  
  
    accuracy - число  
    запятых, которое оставляем после запятой  
    '''  
  
    import cvxpy  
    import numpy as np  
  
    c = np.array(c)  
  
    x = cvxpy.Variable(shape=np.shape(c), integer = True)  
  
    col_n_e = np.array([1]*np.shape(c)[0])  
    row_n_e = np.array([1]*np.shape(c)[1])  
  
    constraints = [(x >= 0),  
                  (x <= 1),  
                  (col_n_e @ x == row_n_e),  
                  (x @ row_n_e.T == col_n_e)]  
  
    total_value = -cvxpy.sum(cvxpy.multiply(c, x))  
    problem = cvxpy.Problem(cvxpy.Minimize(total_value), constraints=constraints)  
  
    print('Макимальная прибыль: {}'.format(abs(round(problem.solve(),accuracy))))  
  
    print('\nМатрица распределения работы агрегатов:\n')  
    for i in range(len(x.value)):  
        print(' '.join([str(j) for j in x.value[i].astype(int)]))  
  
    return (abs(problem.solve()))
```

## 6. Анализ

Запустим скрипт на наших данных:

```
In [2]: c = [[5,7.5,9,11],  
             [4,5.5,8,10],
```

```
[6,8,10,12],  
[4.5,7,9,11]]
```

```
profit(c);
```

Макимальная прибыль: 32.5\$

Матрица распределения работы агрегатов:

```
0 1 0 0  
1 0 0 0  
0 0 0 1  
0 0 1 0
```

Получаем максимальную суммарную прибыль в

32.5\$

Получаем распределение работы агрегатов по объему ресурсов:

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$