1. Проблема

Задача 4

Планирование производства

Есть участок производства, на котором из трех видов ресурсов по четырем технологиям производится продукция. Расход ресурса на единицу продукции по каждому виду технологии, количество ресурсов на складе и их себестоимость приведены в таблице:

Технология / Ресурс	1	2	3	4	Запас ресурсов на складе	Себестоимость единицы ресурсов
1	0	3	9	11	300	0,18
2	3	5	7	0	400	0,22
3	4	8	0	13	450	0,19
Прибыль	15	20	17	21		

Требуется:

1) **Составить оптимальный план работы данного участка.**
2) **Предложить алгоритм решения задачи о целесообразности приобретения 2 единиц одного из видов ресурсов.**

2. Содержательная постановка задачи

Необходимо вычислить максимальную прибыль на данном участке и рассмотреть целесообразноть приобретения 2 единиц одного из видов ресурсов. Нужно не привысить запас ресурсов на складе.

3. Формальная мат. модель

Пусть
$X = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 \end{pmatrix}$ - кол-во произведенного продукта по 4-м технологиям

$s = s_1, s_2, s_3$ - себестоимость 1ой единицы ресурсов
$s = \begin{pmatrix} 0.18 & 0.22 & 0.19 \end{pmatrix}$

A - матрица расходов ресурса на единицу продукции по каждому виду технологии
$A = \begin{pmatrix} 0 & 3 & 9 & 11 \\ 3 & 5 & 7 & 0 \\ 4 & 8 & 0 & 13 \end{pmatrix}$

$V = V_1, V_2, V_3, V_4$ - выручка по каждой технологии производства
$V = \begin{pmatrix} 15 & 20 & 17 & 21 \end{pmatrix}$

$b = b_1, b_2, b_3$ - кол-во ресурса на складе
$b = \begin{pmatrix} 300 & 400 & 450 \end{pmatrix}$

Целевая функция - максимально возможная прибыль:

$$\text{Прибыль} = \text{Выручка} - \text{Стоим. исп. ресурсов} \rightarrow \max$$

Вся выручка:

$$v = V \cdot X$$

$$v = \begin{pmatrix} 15 & 20 & 17 & 21 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

Стоимость ресурсов:

$$c = s \cdot A \cdot X$$

$$c = \begin{pmatrix} 0.18 & 0.22 & 0.19 \end{pmatrix} \cdot \begin{pmatrix} 0 & 3 & 9 & 11 \\ 3 & 5 & 7 & 0 \\ 4 & 8 & 0 & 13 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

В итоге наша задача сводится к следующей:

$$f(x) = v - c \rightarrow max$$

$$A \cdot X \leq b$$
$$x \geq 0$$

4. Алгоритм и ПО

В качестве ПО будем использовать Python с подключенными модулями:

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

5. Решение задачи

Подключаем необходимые модули:

In [1]:
```
import cvxpy
import numpy as np
```

In [2]:
```python
def solution(A, b, v, res_cost):
    x = cvxpy.Variable(shape = len(v), integer = True)

    constraints = [(A[0] @ x <= b[0]),
                   (A[1] @ x <= b[1]),
                   (A[2] @ x <= b[2]),
                   (x >= 0)]

    c = np.sum(A*res_cost.reshape(-1,1), axis=0)
    total_value = sum(cvxpy.multiply(v,x)-cvxpy.multiply(c,x))
    problem = cvxpy.Problem(cvxpy.Maximize(total_value), constraints=constraints)
    print('Максимальная прибыль:', problem.solve())
    print('Кол-во ресурсов на складе:', b)
    print('Остаток ресурсов на складе:', b-np.sum(A*x.value, axis=1))
    print('Кол-во произведенных продуктов:\n По 1ой технологии: {}\n По 2ой технологии: {}\n По 3ей технологии: {}\n По 4
```

In [3]:
```python
def solution2(A, b, v, n, res_cost):
    x = cvxpy.Variable(shape = len(v), integer = True)

    constraints = [(A[0] @ x <= b[0]),
                   (A[1] @ x <= b[1]),
                   (A[2] @ x <= b[2]),
                   (x >= 0)]

    c = np.sum(A*res_cost.reshape(-1,1), axis=0)
    total_value = sum(cvxpy.multiply(v,x)-cvxpy.multiply(c,x))
    problem = cvxpy.Problem(cvxpy.Maximize(total_value), constraints=constraints)
    print('Максимальная прибыль:', problem.solve())
    print('Кол-во ресурсов на складе:', b)
    print('Остаток ресурсов на складе:', b-np.sum(A*x.value, axis=1))
    print('Кол-во произведенных продуктов:\n По 1ой технологии: {}\n По 2ой технологии: {}\n По 3ей технологии: {}\n По 4
```

In [4]:
```
def add_res_iterations(b, sum_up=2):
    for i in range(len(b)):
        a=np.zeros(len(b))
        a[i] = sum_up
        yield (b+a)
```

6. Анализ

Проверим наш алгоритм на реальных данных:

In [5]:
```
A = np.array([[0,3,9,11],[3,5,7,0],[4,8,0,13]])
b = np.array([300,400,450])
res_cost = np.array([0.18,0.22,0.19])
v = np.array([15,20,17,21])
n = 2
```

1 задача

In [6]:
```
solution(A,b,v, res_cost)
```

```
Максимальная прибыль: 1645.52
Кол-во ресурсов на складе: [300 400 450]
Остаток ресурсов на складе: [219.   1.   2.]
Кол-во произведенных продуктов:
 По 1ой технологии: 112.0
 По 2ой технологии: 0.0
 По 3ей технологии: 9.0
 По 4ой технологии: 0.0
```

Можем заметить что выгоднее всего использовать только 1ую и 3ю технологию производства. Также стоит отметить, что на скаладе осталось очень много 1го ресурса. Необходимо снизить количество закупки 1 ресурса и отказаться от 2 и 4 технологии.

2 задача

In [7]:
```
for iteration in add_res_iterations(b):
    solution(A, iteration, v, res_cost)
```

```
Максимальная прибыль: 1645.52
Кол-во ресурсов на складе: [302. 400. 450.]
Остаток ресурсов на складе: [221.   1.   2.]
Кол-во произведенных продуктов:
 По 1ой технологии: 112.0
 По 2ой технологии: 0.0
 По 3ей технологии: 9.0
 По 4ой технологии: 0.0

Максимальная прибыль: 1645.52
Кол-во ресурсов на складе: [300. 402. 450.]
Остаток ресурсов на складе: [219.   3.   2.]
Кол-во произведенных продуктов:
 По 1ой технологии: 112.0
 По 2ой технологии: 0.0
 По 3ей технологии: 9.0
 По 4ой технологии: 0.0

Максимальная прибыль: 1645.52
Кол-во ресурсов на складе: [300. 400. 452.]
Остаток ресурсов на складе: [219.   1.   4.]
Кол-во произведенных продуктов:
 По 1ой технологии: 112.0
 По 2ой технологии: 0.0
 По 3ей технологии: 9.0
 По 4ой технологии: 0.0
```

Как мы видим, добавление 2 любых ресурсов никаким образом не влияет на максимальную прибыль.