1. Проблема

Постановка задачи
Задача построения оптимальных планов
При этом заданы:

- план производства продукции, перевыполнять который нежелательно;
- производительность работ для каждого варианта технологии;
- технологическая схема расположения АГР;
- фонд свободного времени для каждого АГР.

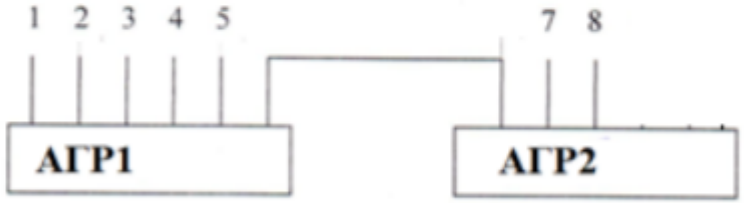

Рисунок 1- функциональная схема производства

Продукт / Вариант	L_1,Т	L_2,Т	L_3,Т	L_4,Т	L_5,Т	L_6,Т	отходы	Р(т/час)
1	0	0	0,135	0,214	0,633	0	0,018	7
2	0	0	0,665	0,318	0	0	0,017	6
3	0	0,355	0	0,212	0,416	0	0,017	5
4	0,497	0	0,27	0,214	0	0	0,019	8
5	0	0,12	0,54	0,321	0	0	0,019	9
6	0,24	0	0	0,321	0	0,42	0,019	10
7	0	0,24	0	0,107	0,633	0	0,02	11

8	0		0,135	0	0,844	0	0,021	12
b,т, план	150	50	100	50	150	100		

T_1=6 ч.T_2=3 ч.

2. Содержательная постановка задачи

I. Задача:

- выполнить план
- минимизировать ресурсы
- уложиться в ограничения по времени

II. Если решения нет:

Рассмотреть минимизицию

1. ресурсов
2. времени
   — рассмотреть при замене первого агрегата на второй
3. отходов
   — без ограничения по времени, но с выполнением плана.

Рассмотреть максимизацию

4. полученного продукта
   — без ограничений по выполнению плана, но с ограничением по времени.
5. прибыли при уступке в количестве ресурсов на 5%

	L1,T	L2,T	L3,T	L4,T	L5,T	L6,T	отходы	P(T/час)	S, у.е.
1	0	0	0.135	0.214	0.633	0	0.018	7	1.8
2	0	0	0.665	0.318	0	0	0.017	6	
3	0	0.355	0	0.212	0.416	0	0.017	5	
4	0.497	0.000	0.270	0.214	0	0	0.019	8	
5	0	0.120	0.540	0.321	0	0	0.019	9	
6	0.240	0	0	0.321	0	0.420	0.019	10	
7	0.240	0	0	0.321	0	0.420	0.019	10	
8	0	0.240	0	0.107	0.633	0	0.020	11	
9	0	0	0.135	0	0.844	0	0.021	12	
план, Т	150	50	100	50	150	100			
С, у.е.	0.9	0.7	0.6	0.6	1.0	0.9			

$$T_1 = 6\text{ч}$$

$$T_2 = 3\text{ч}$$

3. Формальная мат. модель

Пусть

$X = x_1, \ldots, x_9$ — кол-во тонн используемого ресурса по $1, \ldots, 9$ варианту соответственно;

$t = t_1, \ldots, t_9$ — время выполнения каждого варианта технологии;

$t_i = \frac{x_i}{p}$, где p — производительность каждого варианта технологии;

$p = \begin{pmatrix} 7 & 6 & 5 & 8 & 9 & 10 & 10 & 11 & 12 \end{pmatrix}$

$A = ||a_{ij}||$ — матрица кол-ва тонн j-го продукта, получаемого из одной тонны ресурса по i-ой технологии;

$$A = \begin{pmatrix} 0 & 0 & 0.135 & 0.214 & 0.633 & 0 \\ 0 & 0 & 0.665 & 0.318 & 0 & 0 \\ 0 & 0.355 & 0 & 0.212 & 0.416 & 0 \\ 0.497 & 0 & 0.270 & 0.214 & 0 & 0 \\ 0 & 0.12 & 0.540 & 0.321 & 0 & 0 \\ 0.24 & 0 & 0 & 0.321 & 0 & 0.42 \\ 0.24 & 0 & 0 & 0.321 & 0 & 0.42 \\ 0 & 0.24 & 0 & 0.107 & 0.633 & 0 \\ 0 & 0 & 0.135 & 0 & 0.844 & 0 \end{pmatrix}$$

b — план производства продукции;

$b = \begin{pmatrix} 150 & 50 & 100 & 50 & 150 & 100 \end{pmatrix}$

ε — кол-во отходов по каждой технологии;

$\varepsilon = \begin{pmatrix} 0.018 & 0.017 & 0.017 & 0.019 & 0.019 & 0.019 & 0.019 & 0.02 & 0.021 \end{pmatrix}$

c — стоимость каждой произведенной продукции;

$\mathrm{c} = \begin{pmatrix} 0.9 & 0.7 & 0.6 & 0.6 & 1.0 & 0.9 \end{pmatrix}$

s — стоимость ресурса

1. Целевая функция — минимальное кол-во использованных ресурсов:

$$\sum_{i=1}^{9} x_i \to \min$$

$$X \cdot A \geq b$$

$$\sum_{i=1}^{6} t_i \leq T_1$$

$$\sum_{i=7}^{9} t_i \leq T_2$$

$$X \geq 0$$

2. Целевая функция — минимальное кол-во отходов:

$$\sum_{i=1}^{9} \varepsilon_i x_i \to \min$$
$$X \cdot A \geq b$$
$$X \geq 0$$

3. Целевая функция — минимальное время работы агрегатов:

$$\sum_{i=1}^{6} t_i + \sum_{j=7}^{9} t_j \to \min$$
$$X \cdot A \geq b$$
$$X \geq 0$$

4. Целевая функция — максимальное кол-во произведенных ресурсов при временных ограничениях:

$$\sum_{i=1}^{9} x_i \to \max$$
$$\sum_{i=1}^{6} t_i \leq T_1$$
$$\sum_{i=7}^{9} t_i \leq T_2$$
$$X \geq 0$$

5. Целевая функция — максимальная прибыль от производства при уступке в количестве ресурсов на 5%:

$$\left(s \cdot \sum_{i=1}^{9} x_i \right) - \sum_{i=1}^{6} c_j \to \max$$
$$X \cdot A \geq b$$
$$X \geq 0$$
$$\sum_{i=1}^{9} x_i \leq \min_{\substack{X \cdot A \geq b \\ X \geq 0}} \left(\sum_{i=1}^{9} x_i \right) \cdot 105\%$$

Запишем ограничения на языке линейной алгебры:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 0.135 & 0.214 & 0.633 & 0 \\ 0 & 0 & 0.665 & 0.318 & 0 & 0 \\ 0 & 0.355 & 0 & 0.212 & 0.416 & 0 \\ 0.497 & 0 & 0.270 & 0.214 & 0 & 0 \\ 0 & 0.12 & 0.540 & 0.321 & 0 & 0 \\ 0.24 & 0 & 0 & 0.321 & 0 & 0.42 \\ 0.24 & 0 & 0 & 0.321 & 0 & 0.42 \\ 0 & 0.24 & 0 & 0.107 & 0.633 & 0 \\ 0 & 0 & 0.135 & 0 & 0.844 & 0 \end{pmatrix} \geq \begin{pmatrix} 150 \\ 50 \\ 100 \\ 50 \\ 150 \\ 100 \end{pmatrix}$$

$$\begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{pmatrix} \cdot \begin{pmatrix} 1/7 \\ 1/6 \\ 1/5 \\ 1/8 \\ 1/9 \\ 1/10 \end{pmatrix} \leq 6$$

$$\begin{pmatrix} x_7 & x_8 & x_9 \end{pmatrix} \cdot \begin{pmatrix} 1/10 \\ 1/11 \\ 1/12 \end{pmatrix} \leq 3$$

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

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

Функция для решения задач минимизации.

In [2]:
```
def problem_solver(total_value, constraints):
    problem = cvxpy.Problem(cvxpy.Minimize(total_value), constraints = constraints)
    return problem
```

Функция для вывода результатов.

In [3]:
```
def print_results(res_sum, result,T1,T2, waste, overfulfillment, profit):
    print('Кол-во затраченных ресурсов (в тоннах)\n-----------------------------------\n {}\n'.format(res_sum))
    print('Время работы (в часах)\n-----------------------------------\n 1го агрегата: {}\n 2го агрегата: {}\n'.fo
    print('Кол-во отходов (в тоннах)\n-----------------------------------\n {}\n'.format(waste))
    print('Кол-во ресурсов для каждой технологии (в тоннах)\n-----------------------------------\n 1: {}\n 2: {}\n
                                                                                                    result[0][1],
                                                                                                    result[0][2],
                                                                                                    result[0][3],
                                                                                                    result[0][4],
                                                                                                    result[0][5],
                                                                                                    result[0][6],
                                                                                                    result[0][7],
                                                                                                    result[0][8])
    print('Перевыполнение плана (в тоннах)\n-----------------------------------\n {}\n'.format(overfulfillment[0])
    print('Прибыль (в у.е.)\n-----------------------------------\n {}\n'.format(profit))
    print('Прибыль в час (в у.е.)\n-----------------------------------\n {}\n'.format(profit/(max((T1,T2)))))

```

Алгоритм решения задачи минимизации ресурсов.

In [4]:

```python
def minimize_res(A, b, p, eps, T1, T2, prod_cost, res_cost = 1.8):
    x = cvxpy.Variable(shape= (1, A.shape[0]))
    b = b.reshape(1,-1)
    total_value = cvxpy.sum(x)
    x_pos = (x>=0)
    plan = (x@A >= b)
    time_1 = (x[0][:5]@(1/p)[:5] <= T1)
    time_2 = (x[0][5:]@(1/p)[5:] <= T2)
    constraints = [x_pos, plan, time_1, time_2]
    problem = problem_solver(total_value, constraints)
    if problem.status not in ["infeasible", "unbounded"]:
        print('Нет решения.\n\nРешение без ограничения на время.\n')
        constraints = list(set(constraints)-set([time_1, time_2]))
        problem = problem_solver(total_value, constraints)
    problem.solve()
    result = abs(np.round(x.value, 4))
    res_sum = np.sum(result)
    waste = result[0]@eps
    T1 = result[0][:5]@(1/p)[:5]
    T2 = result[0][5:]@(1/p)[5:]
    overfulfillment = np.round(np.sum(result@A, axis=0), 3)-b
    profit = np.sum(res_cost*result[0])-np.sum(prod_cost)
    print_results(res_sum, result, T1, T2, waste, overfulfillment, profit)
    return np.sum(x.value)
```

Алгоритм решения задачи максимизации ресурсов при заданных ограничениях по времени.

In [5]:

```
def maximize_res_with_cons(A, b, p, eps, T1, T2, prod_cost, res_cost = 1.8):
    x = cvxpy.Variable(shape= (1, A.shape[0]))
    b = b.reshape(1,-1)
    total_value = cvxpy.sum(x@A)
    x_pos = (x>=0)
    time_1 = (x[0][:5]@(1/p)[:5] <= T1)
    time_2 = (x[0][5:]@(1/p)[5:] <= T2)
    constraints = [x_pos, time_1, time_2]
    problem = cvxpy.Problem(cvxpy.Maximize(total_value), constraints = constraints)
    problem.solve()
    result = abs(np.round(x.value, 4))
    res_sum = np.sum(result)
    waste = result[0]@eps
    T1 = result[0][:5]@(1/p)[:5]
    T2 = result[0][5:]@(1/p)[5:]
    overfulfillment = np.round(np.sum(result@A, axis=0), 3)-b
    profit = np.sum(res_cost*result[0])-np.sum(prod_cost)
    print_results(res_sum, result, T1, T2, waste, overfulfillment, profit)
```

Алгоритм решения задачи минимизации времени работы агрегатов.

In [6]:

```
def minimize_time(A,b,p,eps, prod_cost, res_cost = 1.8):
    x = cvxpy.Variable(shape= (1, A.shape[0]))
    b = b.reshape(1,-1)
    time_1 = x[0][:5]@(1/p)[:5]
    time_2 = x[0][5:]@(1/p)[5:]
    total_value = time_1 + time_2
    x_pos = (x>=0)
    plan = (x@A >= b)
    constraints = [x_pos, plan]
    problem = problem_solver(total_value, constraints)
    problem.solve()
    result = abs(np.round(x.value, 4))
    res_sum = np.sum(result)
    waste = result[0]@eps
    T1 = result[0][:5]@(1/p)[:5]
    T2 = result[0][5:]@(1/p)[5:]
    overfulfillment = np.round(np.sum(result@A, axis=0), 3)-b
    profit = np.sum(res_cost*result[0])-np.sum(prod_cost)
    print_results(res_sum, result, T1, T2, waste, overfulfillment, profit)
```

Для двух одинаковых агрегатов.

In [7]:

```
def minimize_time_same_agr(A,b,p,eps, prod_cost, res_cost = 1.8):
    x = cvxpy.Variable(shape= (1, A.shape[0]))
    b = b.reshape(1,-1)
    time_1 = x[0][5:]@(1/p)[5:]
    time_2 = x[0][5:]@(1/p)[5:]
    total_value = time_1 + time_2
    x_pos = (x>=0)
    plan = (x@A >= b)
    constraints = [x_pos, plan]
    problem = problem_solver(total_value, constraints)
    problem.solve()
    result = abs(np.round(x.value, 4))
    res_sum = np.sum(result)
    waste = result[0]@eps
    T1 = result[0][5:]@(1/p)[5:]
    T2 = result[0][5:]@(1/p)[5:]
    overfulfillment = np.round(np.sum(result@A, axis=0), 3)-b
    profit = np.sum(res_cost*result[0])-np.sum(prod_cost)
    print_results(res_sum, result, T1, T2, waste, overfulfillment, profit)
```

Алгоритм решения задачи минимизации отходов.

In [8]:

```
def minimize_waste(A, b, p, eps, prod_cost, res_cost = 1.8):
    x = cvxpy.Variable(shape= (1, A.shape[0]))
    b = b.reshape(1,-1)
    eps = eps.reshape(-1,1)
    total_value = x@eps
    x_pos = (x>=0)
    plan = (x@A >= b)
    constraints = [x_pos, plan]
    problem = problem_solver(total_value, constraints)
    waste = problem.solve()
    result = abs(np.round(x.value, 4))
    res_sum = np.sum(result)
    T1 = result[0][:5]@(1/p)[:5]
    T2 = result[0][5:]@(1/p)[5:]
    overfulfillment = np.round(np.sum(result@A, axis=0), 3)-b
    profit = np.sum(res_cost*result[0])-np.sum(prod_cost)
    print_results(res_sum, result, T1, T2, waste, overfulfillment, profit)
```

Алгоритм решения задачи максимизации прибыли.

In [9]:

```python
def maximize_profit(A, b, p, eps, prod_cost, min_num_res, percentage = 1,  res_cost = 1.8):
    x = cvxpy.Variable(shape= (1, A.shape[0]))
    b = b.reshape(1,-1)
    time_1 = x[0][:5]@(1/p)[:5]
    time_2 = x[0][5:]@(1/p)[5:]
    total_value = cvxpy.sum(res_cost*x)-cvxpy.sum(prod_cost)
    x_pos = (x>=0)
    plan = (x@A >= b)
    res_num = (cvxpy.sum(x) <= min_num_res*percentage)
    constraints = [x_pos, plan, res_num]
    problem = cvxpy.Problem(cvxpy.Maximize(total_value), constraints = constraints)
    profit = problem.solve()
    result = abs(np.round(x.value, 4))
    res_sum = np.sum(result)
    waste = result[0]@eps
    T1 = result[0][:5]@(1/p)[:5]
    T2 = result[0][5:]@(1/p)[5:]
    overfulfillment = np.round(np.sum(result@A, axis=0), 3)-b
    print_results(res_sum, result, T1, T2, waste, overfulfillment, profit)
```

6. Анализ

Проверим наш алгоритм на данных:

In [10]:

```
A = np.array([[0,0,0.135,0.214,0.633,0],
              [0,0,0.665,0.318,0,0],
              [0,0.355,0,0.212,0.416,0],
              [0.497,0,0.27,0.214,0,0],
              [0,0.12,0.54,0.321,0,0],
              [0.24,0,0,0.321,0,0.42],
              [0.24,0,0,0.321,0,0.42],
              [0,0.24,0,0.107,0.633,0],
              [0,0,0.135,0,0.844,0]])
b = np.array([150,50,100,50,150,100])
p = np.array([7,6,5,8,9,10,10,11,12])
eps = np.array([0.018, 0.017, 0.017, 0.019, 0.019, 0.019, 0.019, 0.02, 0.021])
prod_cost = np.array([0.9,0.7,0.6,0.6,1,0.9])
T1 = 6
T2 = 3
```

1 пункт.

Минимизация ресурсов.

In [11]:
```
1 min_num_res = minimize_res(A,b,p,eps,T1,T2, prod_cost)
```

```
Нет решения.

Решение без ограничения на время.

Кол-во затраченных ресурсов (в тоннах)
-------------------------------------
 723.8579

Время работы (в часах)
-------------------------------------
 1го агрегата: 32.13142361111111
 2го агрегата: 43.416445

Кол-во отходов (в тоннах)
-------------------------------------
 14.0243319

Кол-во ресурсов для каждой технологии (в тоннах)
-------------------------------------
 1: 0.0
 2: 0.0
 3: 0.0
 4: 186.8353
 5: 78.9931
 6: 119.0476
 7: 119.0476
 8: 168.8368
 9: 51.0975

Перевыполнение плана (в тоннах)
-------------------------------------
 [  0.      0.      0.    109.834   0.      0.   ]

Прибыль (в у.е.)
-------------------------------------
 1298.2442199999998

Прибыль в час (в у.е.)
-------------------------------------
```

```
29.90213086308655
```

Происходит перевыполнение плана по 4 продукту. Первые три технологии на первом агрегате не используются. Так как минимизировалось использование ресурсов и был отказ от ограничения по времени, план будет перевыполняться всегда.

2 пункт.

Минимизация времени работы агрегатов.

In [12]:
```
1 minimize_time(A, b, p, eps, prod_cost)
```

```
Кол-во затраченных ресурсов (в тоннах)
-------------------------------------
 723.8579

Время работы (в часах)
-------------------------------------
 1го агрегата: 32.13142361111111
 2го агрегата: 43.416445

Кол-во отходов (в тоннах)
-------------------------------------
 14.0243319

Кол-во ресурсов для каждой технологии (в тоннах)
-------------------------------------
 1: 0.0
 2: 0.0
 3: 0.0
 4: 186.8353
 5: 78.9931
 6: 119.0476
 7: 119.0476
 8: 168.8368
 9: 51.0975

Перевыполнение плана (в тоннах)
-------------------------------------
 [  0.      0.      0.    109.834   0.      0.   ]

Прибыль (в у.е.)
-------------------------------------
 1298.2442199999998

Прибыль в час (в у.е.)
-------------------------------------
 29.90213086308655
```

При минимизации времени агрегатов мы получаем такой же ответ, как и при минимизации использованных ресурсов, следовательно, мы нашли оптимальный вариант решения, где достигается минимум затрат по времени и минимум использованных ресурсов.

Заменим первый агрегат на второй и минимизируем время:

In [13]:
```
1 minimize_time_same_agr(A, b, p, eps, prod_cost)
```

```
Кол-во затраченных ресурсов (в тоннах)
--------------------------------------
 929.5078000000001

Время работы (в часах)
--------------------------------------
 1го агрегата: 23.809520000000003
 2го агрегата: 23.809520000000003

Кол-во отходов (в тоннах)
--------------------------------------
 17.0040784

Кол-во ресурсов для каждой технологии (в тоннах)
--------------------------------------
 1: 197.5682
 2: 59.9073
 3: 169.5935
 4: 198.6519
 5: 65.6917
 6: 119.0476
 7: 119.0476
 8: 0.0
 9: 0.0

Перевыполнение плана (в тоннах)
--------------------------------------
 [  5.873  18.089  55.62  187.311  45.612   0.   ]

Прибыль (в у.е.)
--------------------------------------
 1668.41404

Прибыль в час (в у.е.)
--------------------------------------
 70.07340089174414
```

Два одинаковых агрегата выполняют план в разы быстрее, однако при этом тратят больше ресурсов. Прибыль в час в разы больше, стоит задуматься над заменой первого агрегата.

3 пункт.

Минимизация отходов.

In [14]:
```
1 minimize_waste(A, b, p, eps, prod_cost)
```

```
Кол-во затраченных ресурсов (в тоннах)
-------------------------------------
 726.611

Время работы (в часах)
-------------------------------------
 1го агрегата: 60.2786825
 2го агрегата: 32.834845

Кол-во отходов (в тоннах)
-------------------------------------
 13.635462812144064

Кол-во ресурсов для каждой технологии (в тоннах)
-------------------------------------
 1: 0.0
 2: 52.5315
 3: 140.8451
 4: 186.8353
 5: 0.0
 6: 119.0476
 7: 119.0476
 8: 0.0
 9: 108.3039

Перевыполнение плана (в тоннах)
-------------------------------------
 [  0.      0.      0.    112.975   0.      0.   ]

Прибыль (в у.е.)
-------------------------------------
 1303.1998

Прибыль в час (в у.е.)
-------------------------------------
 21.619580023169885
```

Как мы видим, минимизация отходов ведёт к увеличению используемых ресурсов. Также происходит скачок во времени работы первого агрегата. Это связано с тем, что в его технологиях выходит наименьшее кол-во отходов. План 4 продукта также перевыполняется.

4 пункт.

Узнаем, сколько можно произвести продуктов при заданных ограничениях.

In [15]:
```
1 maximize_res_with_cons(A, b, p, eps, T1, T2, prod_cost)
```

```
Кол-во затраченных ресурсов (в тоннах)
--------------------------------------
 90.0

Время работы (в часах)
--------------------------------------
 1го агрегата: 6.0
 2го агрегата: 3.0

Кол-во отходов (в тоннах)
--------------------------------------
 1.782

Кол-во ресурсов для каждой технологии (в тоннах)
--------------------------------------
 1: 0.0
 2: 0.0
 3: 0.0
 4: 0.0
 5: 54.0
 6: 0.0
 7: 0.0
 8: 0.0
 9: 36.0

Перевыполнение плана (в тоннах)
--------------------------------------
 [-150.     -43.52   -65.98   -32.666 -119.616 -100.   ]

Прибыль (в у.е.)
--------------------------------------
 157.3

Прибыль в час (в у.е.)
--------------------------------------
 26.21666666666667
```

При данных ограничениях производится совсем немного продукции, и то не каждого типа.

5 пункт.

Посчитаем максимальную прибыль при уступке в количестве ресурса в 5%

In [16]:
```
1 maximize_profit(A, b, p, eps, prod_cost, min_num_res, percentage=1.05)
```

```
Кол-во затраченных ресурсов (в тоннах)
-------------------------------------
 760.051

Время работы (в часах)
-------------------------------------
 1го агрегата: 52.498645714285715
 2го агрегата: 37.78253939393939

Кол-во отходов (в тоннах)
-------------------------------------
 14.416146799999998

Кол-во ресурсов для каждой технологии (в тоннах)
-------------------------------------
 1: 28.1231
 2: 31.9736
 3: 77.5873
 4: 186.8936
 5: 38.4567
 6: 121.252
 7: 121.252
 8: 86.6027
 9: 67.91

Перевыполнение плана (в тоннах)
-------------------------------------
 [  1.087   2.943   5.455 122.085  12.214   1.852]

Прибыль (в у.е.)
-------------------------------------
 1363.391605374793

Прибыль в час (в у.е.)
-------------------------------------
 25.970033832773566
```

Суммарная прибыль возрастает не сильно, однако прибыль в час уменьшилась по сравнению с использованием минимального количества ресурсов. Для достижения плана, лучше не делать уступку.

Итог

Стоит пересмотреть план по 4 продукту в сторону его увеличения. Для скорости производства предлагается заменить первый агрегат на второй. Предлагаемые ограничения по работе агрегатов существенно отличается от реальных оптимальных значений.