МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

З В І Т
Лабораторна робота №1(М)
«Лінійне програмування»

з дисципліни
«Математичне моделювання»

Виконавець:

студент групи КІ-22м Косей М.П.

Керівник:

викладач Вдовиченко І. Н.

2023

Лабораторна робота №1(М)

Тема: ***Лінійне програмування***
Мета: ***Вирішити задачі***

ЗАВДАННЯ:

1) Ознайомитись з теоретичними відомостями до лабораторної роботи.

Математичне програмування - це метод вирішення оптимізаційних задач з використанням математичних моделей та алгоритмів.

Лінійне програмування є одним з методів математичного програмування та використовується для оптимізації лінійних функцій від змінних з обмеженнями в формі лінійних рівнянь або нерівностей.

При розв'язуванні задачі лінійного програмування необхідно побудувати математичну модель, яка описує цільову функцію та обмеження

2) Задача 1

У майстерні "ПромАрт" освоєно виробництво столів та тумбочок для торгової мережі. Для їх виготовлення є два види деревини: **I - 72 м³** та **II - 56 м³**. Кожен виріб вимагає обох видів деревини в м³:

№	Найменування	I	II
1	Стіл	0,18	0,08
2	Тумбочка	0,09	0,28

З виробництва одного столу "ПромАрт" отримує чистого доходу **1,1 гривні**, а від виробництва однієї тумбочки - **70 копійок**.

Визначте, скільки столів та тумбочок повинна виготовити майстерня з наявної деревини, щоб забезпечити найбільший дохід.

Вирішення

Для задачі про виробництво столів та тумбочок з обмеженими запасами деревини можна побудувати наступну математичну модель лінійного програмування:

Змінні:

$\mathbf{x_1}$ **-** кількість столів, що будуть вироблені
$\mathbf{x_2}$ **-** кількість тумбочок, що будуть вироблені

Цільова функція:

maximize $\mathbf{1.11 \times x_1 + 0.7 \times x_2}$

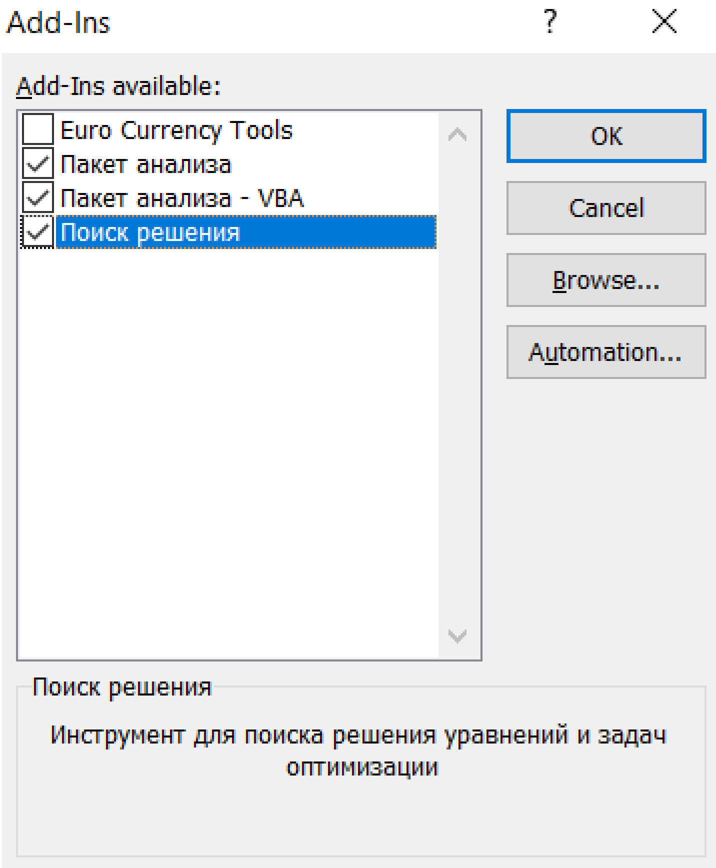
Обмеження:

$\mathbf{0.18 \times x_1 + 0.09 \times x_2 <= 72}$ (обмеження на деревину I)
$\mathbf{0.08 \times x_1 + 0.28 \times x_2 <= 56}$ (обмеження на деревину II)
$\mathbf{x_1, x_2}$ $>=$ **0** (неможливо виробити від'ємну кількість виробів)
$\mathbf{x_1, x_2}$ **– цілі числа** (неможливо виробити дрібну кількість виробів)

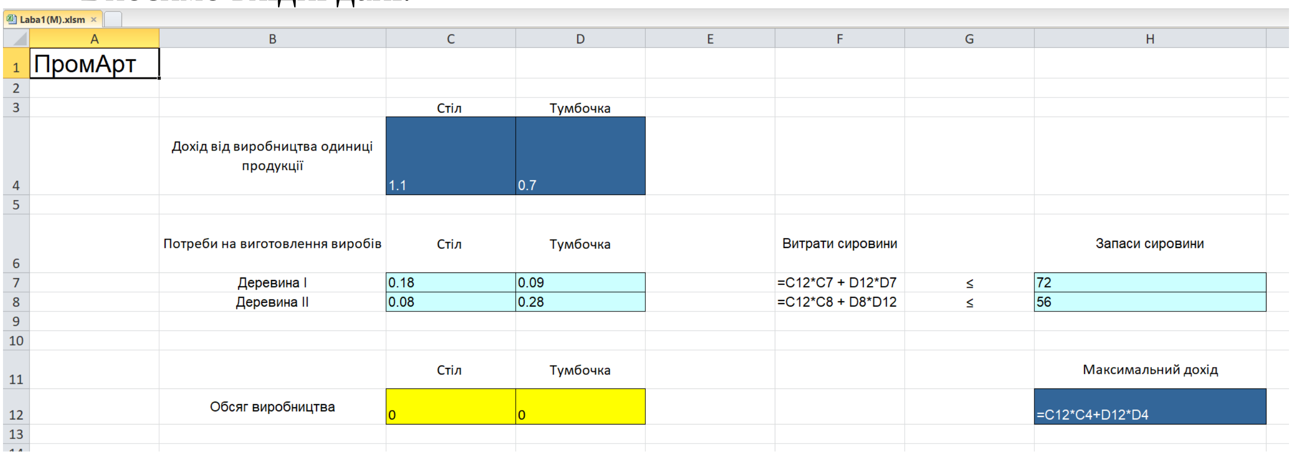
Ця модель має ціль максимізувати прибуток, який залежить від кількості вироблених столів та тумбочок, при тому, що обсяг використання кожного типу деревини не може перевищувати обмежень.

Задача може бути вирішена за допомогою спеціального програмного забезпечення, такого як «MS Excel», яке має вбудовані інструменти розв'язування задач лінійного програмування.

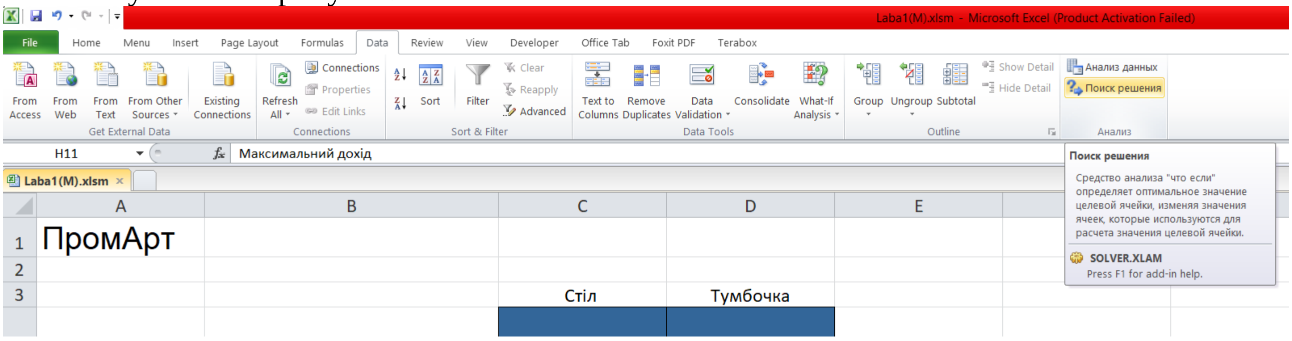
Вмикаємо додаток для вирішення задачи

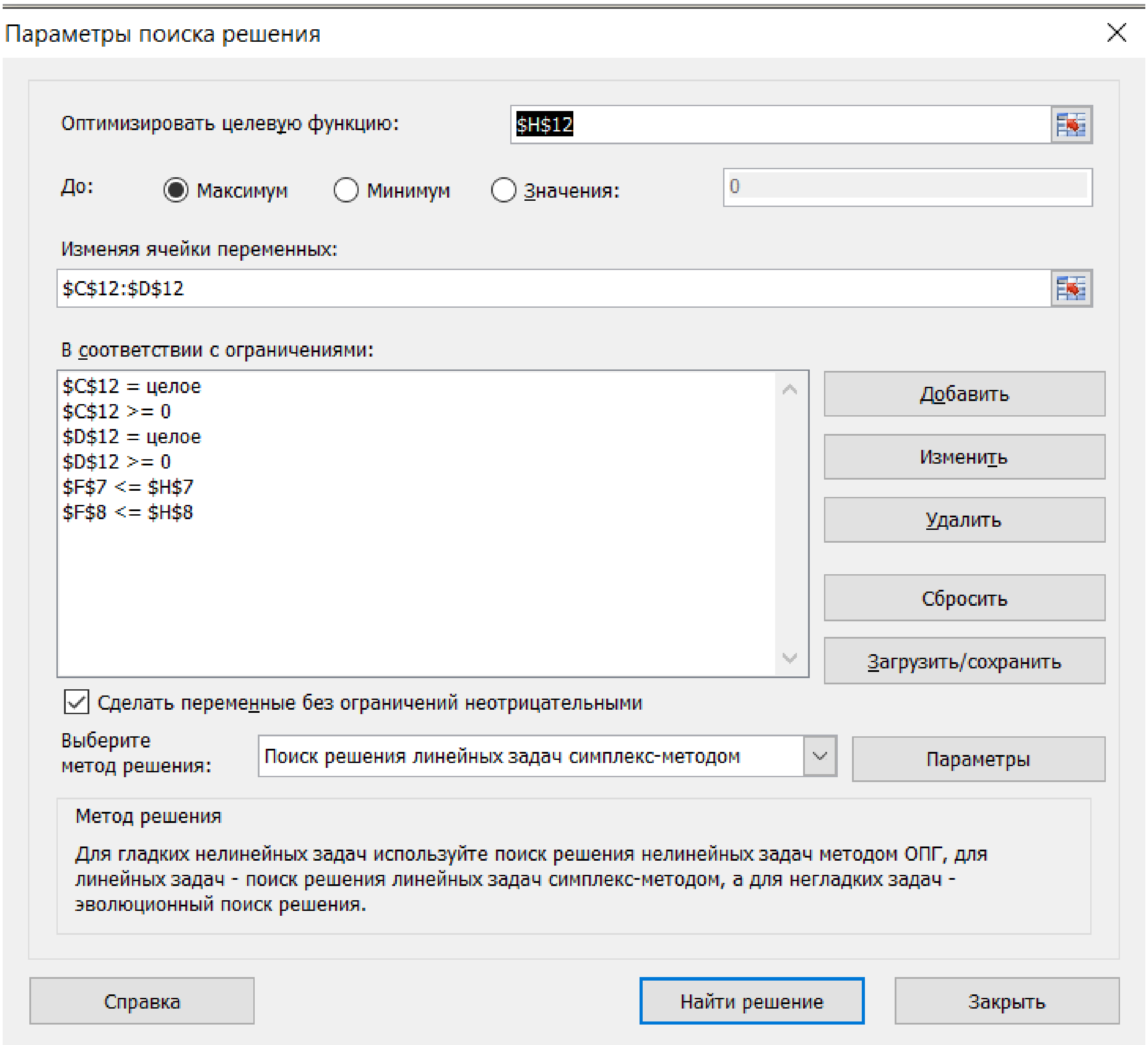


Вносимо вхідні дані.



Запускаємо вирішувач.





Результати вирішення

	A	B	C	D	E	F	G	H
1	ПромАрт							
2								
3			Стіл	Тумбочка				
4		Дохід від виробництва одиниці продукції	1.1	0.7				
5								
6		Потреби на виготовлення виробів	Стіл	Тумбочка		Витрати сировини		Запаси сировини
7		Деревина I	0.18	0.09		72	≤	72
8		Деревина II	0.08	0.28		56	≤	56
9								
10								
11			Стіл	Тумбочка				Максимальний дохід
12		Обсяг виробництва	350	100				455
13								

Графічне вирішення задачі.

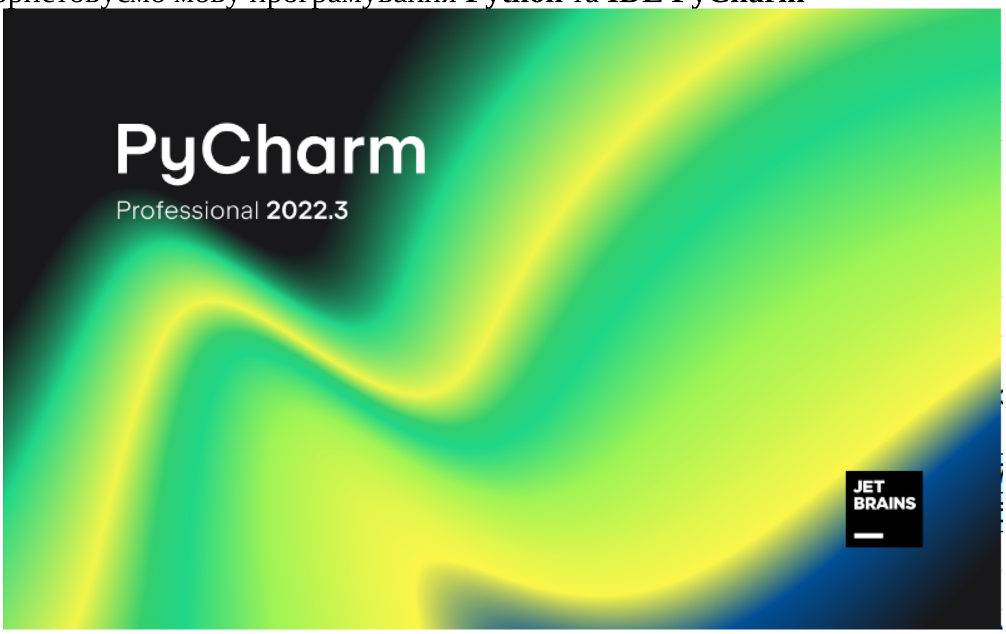
Для вирішення задачі потрібно побудувати графіки функцій:

- цільова функція $\mathbf{1.11 \times x_1 + 0.7 \times x_2}$
- та обмежень

  $\mathbf{0.18\times x_1 + 0.09\times x_2 <= 72}$ (обмеження на деревину I)

  $\mathbf{0.08\times x_1 + 0.28\times x_2 <= 56}$ (обмеження на деревину II)

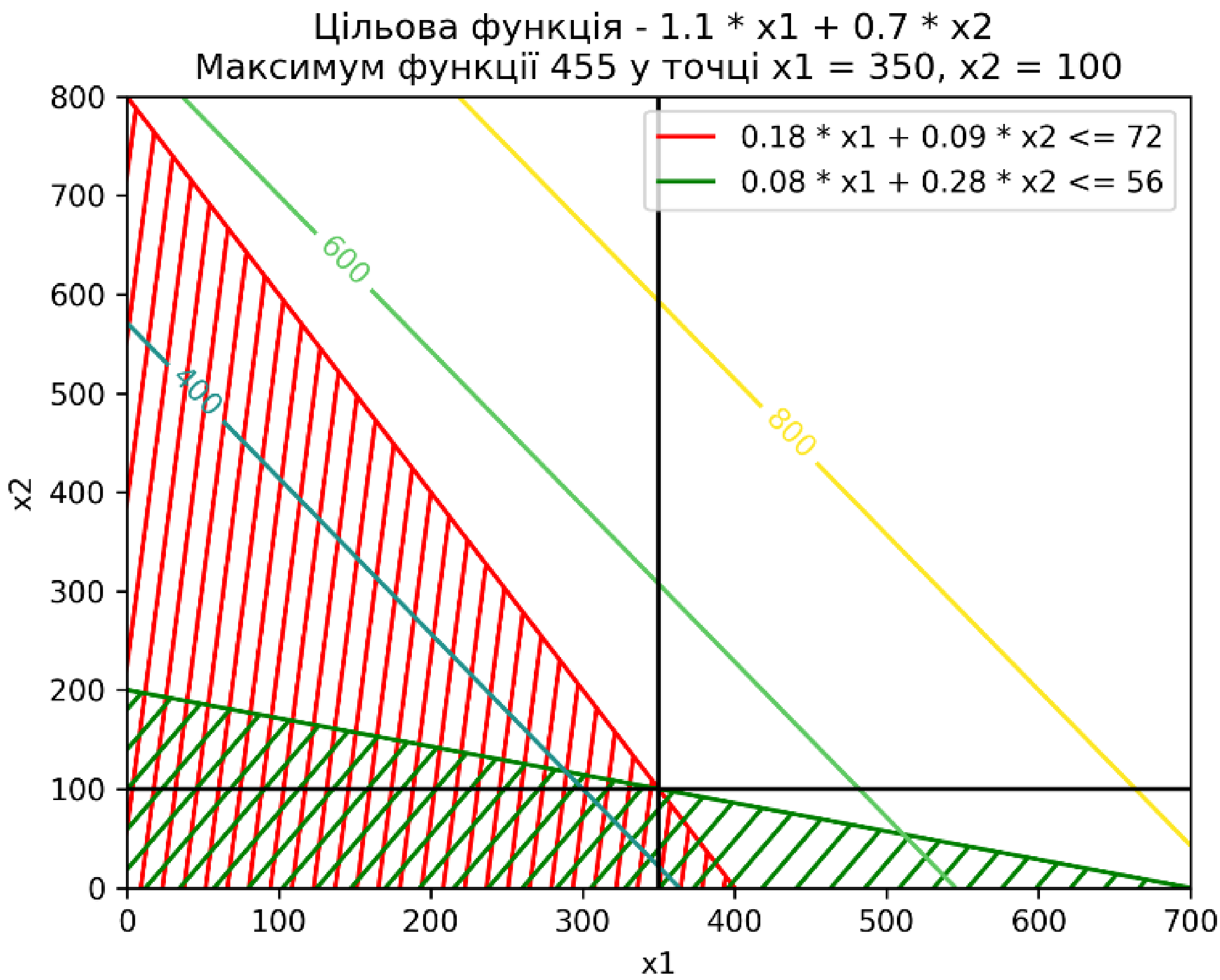
Використовуємо мову програмування **Python** та **IDE PyCharm**



main.py

```python
import matplotlib.pyplot as plt
import matplotlib.axes as axes
import matplotlib.figure as figure
from matplotlib import patheffects
import numpy as np


# Визначення функцій
2 usages
def target_func(x1, x2):
    return 1.1 * x1 + 0.7 * x2


1 usage
def constraint1(x1, x2):
    return 72 - 0.18 * x1 - 0.09 * x2


1 usage
def constraint2(x1, x2):
    return 56 - 0.08 * x1 - 0.28 * x2

```

```python
# Побудова графіків
fig, ax = plt.subplots() # type:figure.Figure, axes.Axes

x1 = np.linspace(0, 700, 701)
x2 = np.linspace(0, 800, 801)
X1, X2 = np.meshgrid(x1, x2)

# Обмеження
Z1 = constraint1(X1, X2)
Z2 = constraint2(X1, X2)

C_Z1=ax.contour(X1, X2, Z1, levels=[0], colors='r')
C_Z2=ax.contour(X1, X2, Z2, levels=[0], colors='g')
plt.setp(C_Z2.collections, path_effects=[patheffects.withTickedStroke(angle=60, length=30)])
plt.setp(C_Z1.collections, path_effects=[patheffects.withTickedStroke(angle=135, length=30)])
h1,l1 = C_Z1.legend_elements()
h2,l1 = C_Z2.legend_elements()
plt.legend([h1[0], h2[0]], ['0.18 * x1 + 0.09 * x2 <= 72', '0.08 * x1 + 0.28 * x2 <= 56'])

ax.set_xlabel('x1')
ax.set_ylabel('x2')

# Цільова функція
Z = target_func(X1, X2)
CS_Z = ax.contour(X1, X2, Z, levels=[0, 400, 600, 800])
ax.clabel(CS_Z, inline=True, fontsize=10)


#Максимум цільової функції  у точці перетину '0.18 * x1 + 0.09 * x2 = 72 та '0.08 * x1 + 0.28 * x2 = 56
A = np.array([[0.18, 0.09],
              [0.08, 0.28]])
y = np.array([72, 56])

x = np.linalg.solve(A, y)
S_x1 = x[0]
S_x2 = x[1]

ax.axvline(x=S_x1, color='black')
ax.axhline(y=S_x2, color='black')

ax.set_title('Цільова функція - 1.1 * x1 + 0.7 * x2 \n' +
             'Максимум функції ' + str(round(target_func(S_x1, S_x2))) + ' у точці x1 = ' + str(round(S_x1)) +
             ', x2 = ' + str(round(S_x2)))

plt.savefig('figure.png', dpi=300)
```



3) Задача 2

Є три екскаватори різних марок. З їх допомогою треба виконати три види земляних робіт обсягом 20 000 м/3. Час роботи екскаваторів однаковий, продуктивність у м^3-годину по кожному виду робіт наведена в таблиці:

Екскаватор	Вид роботи		
	A	B	C
I	105	56	56
II	107	66	83
III	64	38	53

Необхідно розподілити час роботи так, щоб завдання було виконано за найкоротший час.

Вирішення

Для розв'язання цієї задачі ми можемо скористатися методом лінійного програмування. Давайте визначимо змінні:

Визначимо змінні:

x_1 - час, який витрачає екскаватор I на вид робіт A

x_2 - час, який витрачає екскаватор II на вид робіт A

x_3 - час, який витрачає екскаватор III на вид робіт А
y_1 - час, який витрачає екскаватор I на вид робіт B
y_2 - час, який витрачає екскаватор II на вид робіт B
y_3 - час, який витрачає екскаватор III на вид робіт B
z_1 - час, який витрачає екскаватор I на вид робіт C
z_2 - час, який витрачає екскаватор II на вид робіт C
z_3 - час, який витрачає екскаватор III на вид робіт C

Цільова функція:
minimize $x_1 + x_2 + x_3 + y_1 + y_2 + y_3 + z_1 + z_2 + z_3$

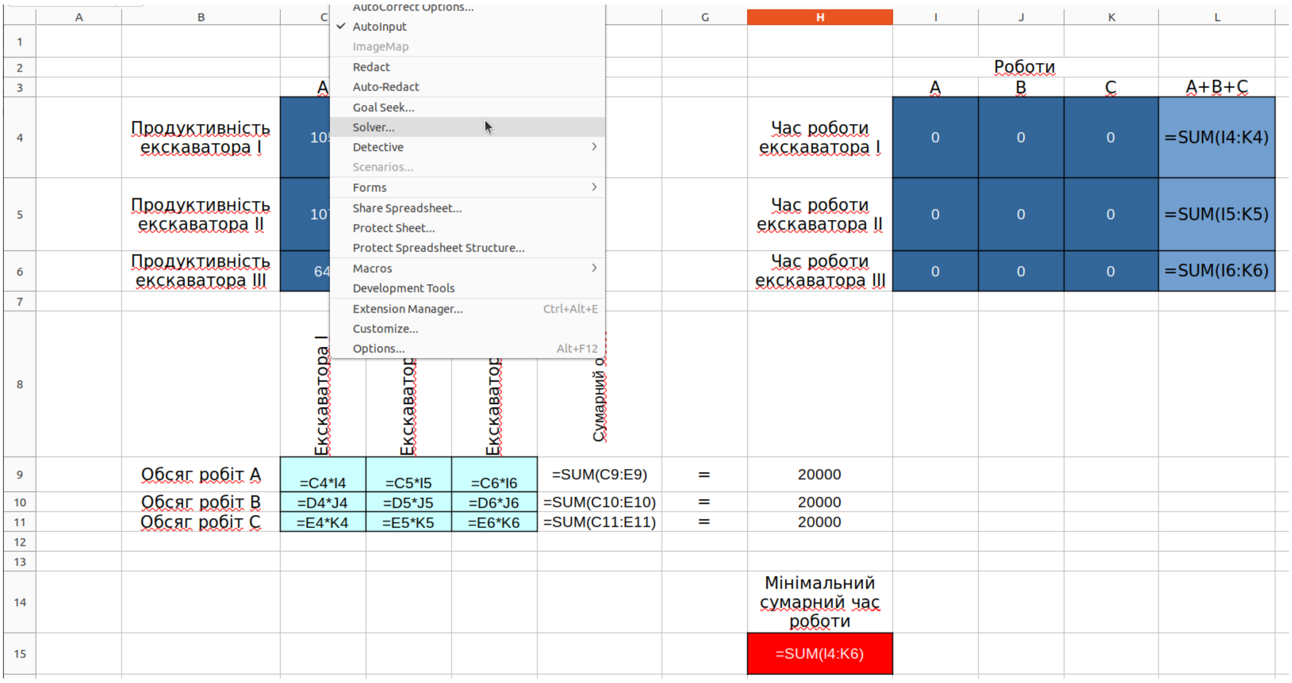
Обмеження:
$105 \times x_1 + 107 \times y_1 + 64 \times z_1 = 20\,000$ (обмеження на обсяг робіт A)
$56 \times x_2 + 107 \times y_2 + 107 \times z_2 = 20\,000$ (обмеження на обсяг робіт B)
$56 \times x_3 + 83 \times y_3 + 53 \times z_3 = 20\,000$ (обмеження на обсяг робіт C)
$x_1 + y_1 + z_1 = x_2 + y_2 + z_2 = x_3 + y_3 + z_3$ (час роботи екскаваторів однакова)
$x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3 >= 0$ (час роботи повинен бути більше 0)

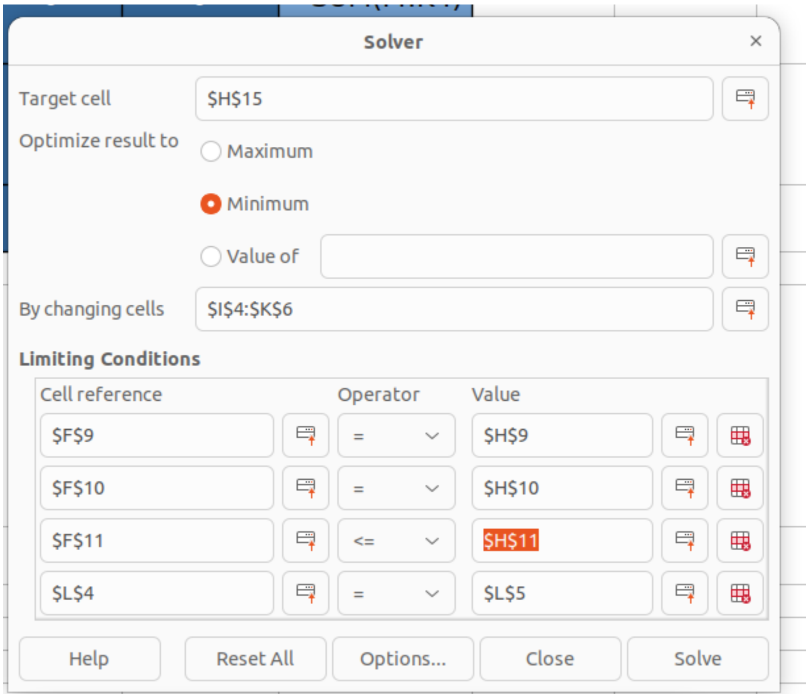
Задача може бути вирішена за допомогою спеціального програмного забезпечення, такого як LiberOffice Calc(аналог «MS Excel»), яке має вбудовані інструменти розв'язування задач лінійного програмування.

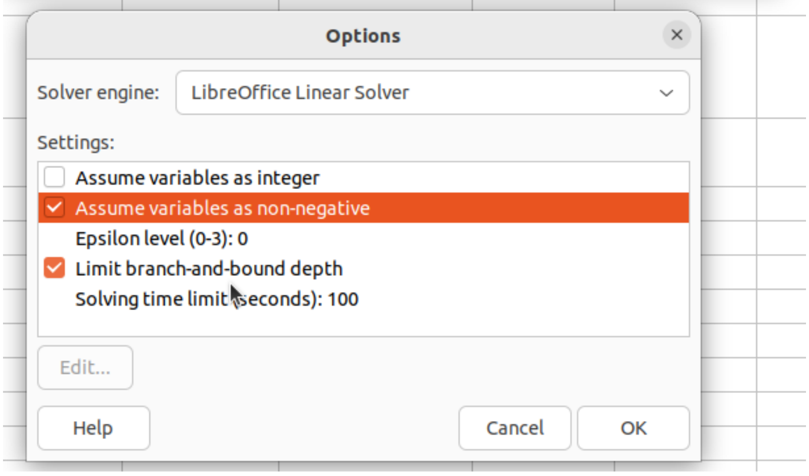
Вносимо вхідні дані

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2				Роботи						Роботи		
3			A	B	C				A	B	C	A+B+C
4		Продуктивність екскаватора I	105	56	56			Час роботи екскаватора I	0	0	0	=SUM(I4:K4)
5		Продуктивність екскаватора II	107	66	83			Час роботи екскаватора II	0	0	0	=SUM(I5:K5)
6		Продуктивність екскаватора III	64	38	53			Час роботи екскаватора III	0	0	0	=SUM(I6:K6)
7												
8			Екскаватора I	Екскаватора II	Екскаватора III	Сумарний обсяг						
9		Обсяг робіт A	=C4*I4	=C5*I5	=C6*I6	=SUM(C9:E9)	=	20000				
10		Обсяг робіт B	=D4*J4	=D5*J5	=D6*J6	=SUM(C10:E10)	=	20000				
11		Обсяг робіт C	=E4*K4	=E5*K5	=E6*K6	=SUM(C11:E11)	=	20000				
12												
13												
14								Мінімальний сумарний час роботи				
15								=SUM(I4:K6)				

Запускаємо вирішувач







Результати вирішення

	Роботи A	Роботи B	Роботи C
Продуктивність екскаватора I	105	56	56
Продуктивність екскаватора II	107	66	83
Продуктивність екскаватора III	64	38	53

	Роботи A	Роботи B	Роботи C	A+B+C
Час роботи екскаватора I	190.48	93.24	0.00	283.72
Час роботи екскаватора II	0.00	223.92	59.80	283.72
Час роботи екскаватора III	0.00	0.00	283.72	283.72

	Екскаватора I	Екскаватора II	Екскаватора III	Сумарний обсяг		
Обсяг робіт A	20000.00	0.00	0.00	20000	=	20000
Обсяг робіт B	5221.37	14778.63	0.00	20000	=	20000
Обсяг робіт C	0.00	4963.10	15036.90	20000	=	20000

Мінімальний сумарний час роботи
851.15

ВИСНОВКИ

В результаті виконаної лабораторної роботи було розглянуто один з методів математичного програмування - лінійне програмування, а також вирішенні задачі оптимізаціїї.

Усі матеріали викладенні у репозіторії GitHub, за посиланням https://github.com/Max11mus/Mathematical-Modeling-Lenear-Programming-Lab1.git.