

**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 м<sup>3</sup>** та **II - 56 м<sup>3</sup>**. Кожен виріб вимагає обох видів деревини в м<sup>3</sup>:

№	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:

$x_1$  - кількість столів, що будуть вироблені

$x_2$  - кількість тумбочок, що будуть вироблені

#### Цільова функція:

**maximize**  $1.11 \times x_1 + 0.7 \times x_2$

#### Обмеження:

$0.18 \times x_1 + 0.09 \times x_2 \leq 72$  (обмеження на деревину I)

$0.08 \times x_1 + 0.28 \times x_2 \leq 56$  (обмеження на деревину II)

$x_1, x_2 \geq 0$  (неможливо виробити від'ємну кількість виробів)

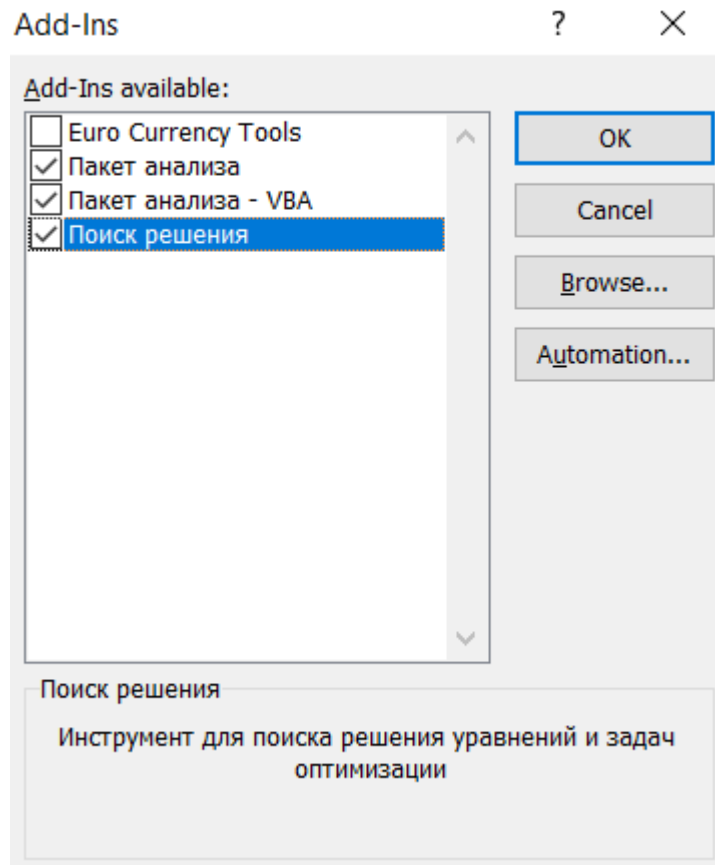
$x_1, x_2$  – **цілі числа** (неможливо виробити дрібну кількість виробів)

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 задачі



Вносимо вхідні дані.

	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		=C12*C7 + D12*D7	≤	72
8		Деревина II	0.08	0.28		=C12*C8 + D8*D12	≤	56
9								
10								
11			Стіл	Тумбочка				Максимальний дохід
12		Обсяг виробництва	0	0				=C12*C4+D12*D4
13								

## Запускаємо вирішувач.

Labat1(M).xlsx - Microsoft Excel (Product Activation Failed)

File Home Menu Insert Page Layout Formulas Data Review View Developer Office Tab Foxit PDF Terabox

From Access From Web From Text From Other Sources Existing Connections Refresh All Properties Edit Links Connections Sort & Filter Sort Filter Reapply Advanced Text to Columns Remove Duplicates Data Consolidate What-if Analysis Group Ungroup Subtotal Show Detail Hide Detail

H11 Максимальний дохід

	A	B	C	D	E
1	ПромАрт				
2					
3			Стіл	Тумбочка	

Поиск решения  
Средство анализа "что если" определяет оптимальное значение целевой ячейки, изменяя значения ячеек, которые используются для расчета значения целевой ячейки.  
SOLVER.XLAM  
Press F1 for add-in help.

## Параметры поиска решения

Оптимизировать целевую функцию:

До: ☒ Максимум ☐ Минимум ☐ Значения:

Изменяя ячейки переменных:

В соответствии с ограничениями:

\$C\$12 = целое  
\$C\$12 >= 0  
\$D\$12 = целое  
\$D\$12 >= 0  
\$F\$7 <= \$H\$7  
\$F\$8 <= \$H\$8

Добавить  
Изменить  
Удалить  
Сбросить  
Загрузить/сохранить

☒ Сделать переменные без ограничений неотрицательными

Выберите метод решения:

Метод решения  
Для гладких нелинейных задач используйте поиск решения нелинейных задач методом ОПГ, для линейных задач - поиск решения линейных задач симплекс-методом, а для негладких задач - эволюционный поиск решения.

Справка

### Результати вирішення

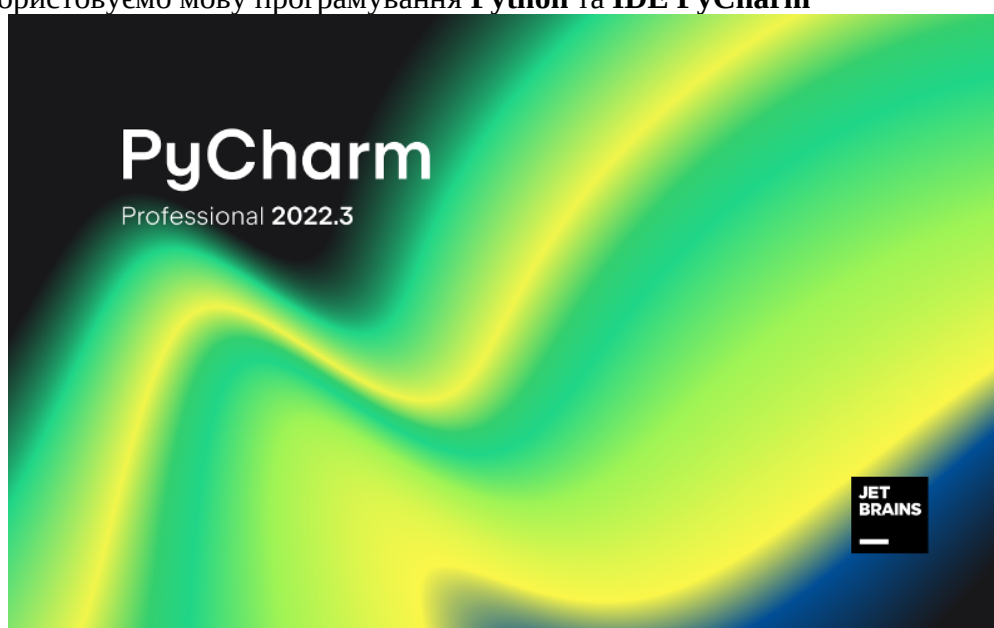
	A	B	C	D	E	F	G	H	
1	ПромАрт								
2									
3			Стіл	Тумбочка					
4		Дохід від виробництва одиниці продукції	1.1	0.7					
5									
6		Потреби на виготовлення виробів	Стіл	Тумбочка		Витрати сировини		Запаси сировини	
7		Деревина I	0.18	0.09		72	≤	72	
8		Деревина II	0.08	0.28		56	≤	56	
9									
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  $1.11 \times x_1 + 0.7 \times x_2$
- та обмежень
  - $0.18 \times x_1 + 0.09 \times x_2 \leq 72$  (обмеження на деревину I)
  - $0.08 \times x_1 + 0.28 \times x_2 \leq 56$  (обмеження на деревину II)

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



```

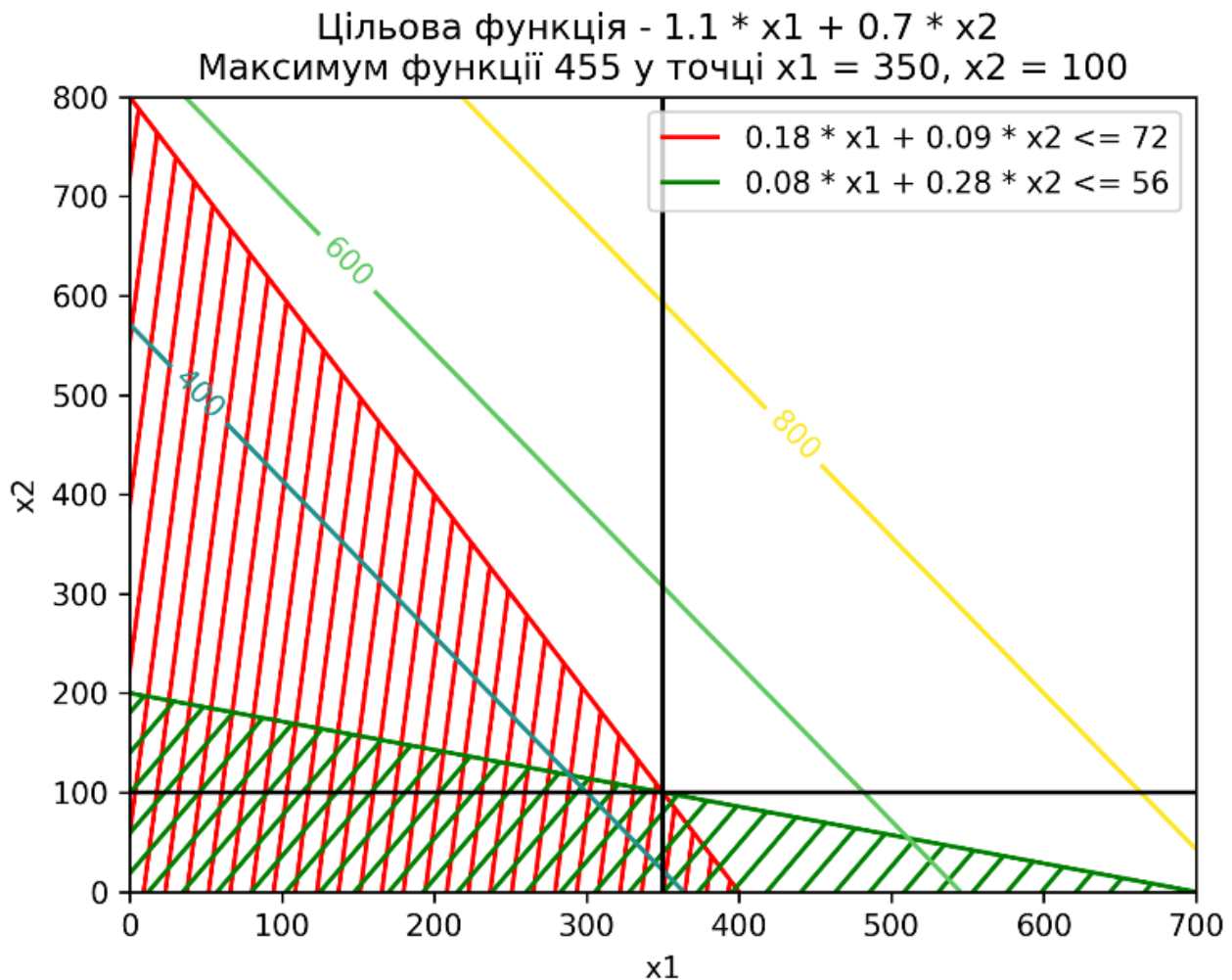
main.py x
1 import matplotlib.pyplot as plt
2 import matplotlib.axes as axes
3 import matplotlib.figure as figure
4 from matplotlib import patheffects
5 import numpy as np
6
7
8 # Визначення функцій
9
10 def target_func(x1, x2):
11     return 1.1 * x1 + 0.7 * x2
12
13
14 def constraint1(x1, x2):
15     return 72 - 0.18 * x1 - 0.09 * x2
16
17
18 def constraint2(x1, x2):
19     return 56 - 0.08 * x1 - 0.28 * x2
20

```

```

21 # Побудова графіків
22 fig, ax = plt.subplots() # type: figure.Figure, axes.Axes
23
24 x1 = np.linspace(0, 700, 701)
25 x2 = np.linspace(0, 800, 801)
26 X1, X2 = np.meshgrid(x1, x2)
27
28 # Обмеження
29 Z1 = constraint1(X1, X2)
30 Z2 = constraint2(X1, X2)
31
32 C_Z1=ax.contour(X1, X2, Z1, levels=[0], colors='r')
33 C_Z2=ax.contour(X1, X2, Z2, levels=[0], colors='g')
34 plt.setp(C_Z2.collections, path_effects=[patheffects.withTickedStroke(angle=60, length=30)])
35 plt.setp(C_Z1.collections, path_effects=[patheffects.withTickedStroke(angle=135, length=30)])
36 h1,l1 = C_Z1.legend_elements()
37 h2,l1 = C_Z2.legend_elements()
38 plt.legend([h1[0], h2[0]], ['0.18 * x1 + 0.09 * x2 <= 72', '0.08 * x1 + 0.28 * x2 <= 56'])
39
40 ax.set_xlabel('x1')
41 ax.set_ylabel('x2')
42
43 # Цільова функція
44 Z = target_func(X1, X2)
45 CS_Z = ax.contour(X1, X2, Z, levels=[0, 400, 600, 800])
46 ax.clabel(CS_Z, inline=True, fontsize=10)
47
48
49 #Максимум цільової функції у точці перетину '0.18 * x1 + 0.09 * x2 = 72 та '0.08 * x1 + 0.28 * x2 = 56
50 A = np.array([[0.18, 0.09],
51               [0.08, 0.28]])
52 y = np.array([72, 56])
53
54 x = np.linalg.solve(A, y)
55 S_x1 = x[0]
56 S_x2 = x[1]
57
58 ax.axvline(x=S_x1, color='black')
59 ax.axhline(y=S_x2, color='black')
60
61 ax.set_title('Цільова функція - 1.1 * x1 + 0.7 * x2 \n' +
62             'Максимум функції ' + str(round(target_func(S_x1, S_x2))) + ' у точці x1 = ' + str(round(S_x1)) +
63             ', x2 = ' + str(round(S_x2)))
64
65 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<sup>3</sup>. Час роботи екскаваторів однаковий, продуктивність у м<sup>3</sup>-годину по кожному виду робіт наведена в таблиці:

Екскаватор	Вид роботи		
	А	В	С
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 А

$x_2$  - час, який витрачає екскаватор II на вид робіт А

$x_3$  - час, який витрачає екскаватор III на вид робіт A  
 $y_1$  - 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)

$x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3 \geq 0$  (час роботи повинен бути більше 0)

Задача може бути вирішена за допомогою спеціального програмного забезпечення, такого як LibreOffice 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

	A	B	C		G	H	I	J	K	L
1										
2										
3										
4		Продуктивність екскаватора I	10			Час роботи екскаватора I	0	0	0	=SUM(I4:K4)
5		Продуктивність екскаватора II	10			Час роботи екскаватора II	0	0	0	=SUM(I5:K5)
6		Продуктивність екскаватора III	64			Час роботи екскаватора III	0	0	0	=SUM(I6:K6)
7										
8										
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)				

**Solver**

Target cell:

Optimize result to: ☐ Maximum ☒ Minimum ☐ Value of

By changing cells:

**Limiting Conditions**

Cell reference	Operator	Value
<input type="text" value="\$F\$9"/>	<input type="text" value="="/>	<input type="text" value="\$H\$9"/>
<input type="text" value="\$F\$10"/>	<input type="text" value="="/>	<input type="text" value="\$H\$10"/>
<input type="text" value="\$F\$11"/>	<input type="text" value="&lt;="/>	<input type="text" value="\$H\$11"/>
<input type="text" value="\$L\$4"/>	<input type="text" value="="/>	<input type="text" value="\$L\$5"/>

Help Reset All Options... Close Solve

**Options**

Solver engine:

Settings:

☐ Assume variables as integer

☒ Assume variables as non-negative

Epsilon level (0-3): 0

☒ Limit branch-and-bound depth

Solving time limit (seconds): 100

Edit... Help Cancel OK

## Результати вирішення

	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	190.48	93.24	0.00	283.72
5		Продуктивність екскаватора II	107	66	83			Час роботи екскаватора II	0.00	223.92	59.80	283.72
6		Продуктивність екскаватора III	64	38	53			Час роботи екскаватора III	0.00	0.00	283.72	283.72
7												
8			Екскаватора I	Екскаватора II	Екскаватора III	Сумарний обсяг						
9		Обсяг робіт A	20000.00	0.00	0.00	20000	=	20000				
10		Обсяг робіт B	5221.37	14778.63	0.00	20000	=	20000				
11		Обсяг робіт C	0.00	4963.10	15036.90	20000	=	20000				
12												
13												
14								Мінімальний сумарний час роботи				
15								851.15				
16												

## **ВИСНОВКИ**

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

**Усі матеріали викладенні у репозиторії GitHub, за посиланням <https://github.com/Max11mus/Mathematical-Modeling-Linear-Programming-Lab1.git>.**