

НУЛП, ІКНІ, САП		Тема	оцінка	підпис
КН-406	1(номер лаб)	МЕТОДИ РОЗВ'ЯЗАННЯ ЗАДАЧ БАГАТОКРИТЕРІАЛЬНОЇ ОПТИМІЗАЦІЇ НА ОСНОВІ ПОБУДОВИ МНОЖИНИ ПАРЕТО		
Ваврик Р.Р.				
№ залікової:				
Методи багатокритеріальної оптимізації			Викладач: к.т.н., доц. каф. Мельник М.Р.	

Мета

Метою роботи є ознайомлення студентів з теоретичними основами побудови множини Парето та отримання навичок розв'язання ЗБО.

Завдання:

Деяка фірма планує організувати виробництво дверей. На даний момент існує можливість придбати 4 варіанти станків. В процесі розв'язання задачі використати принцип Парето.

Параметри станків

Таблиця 1.

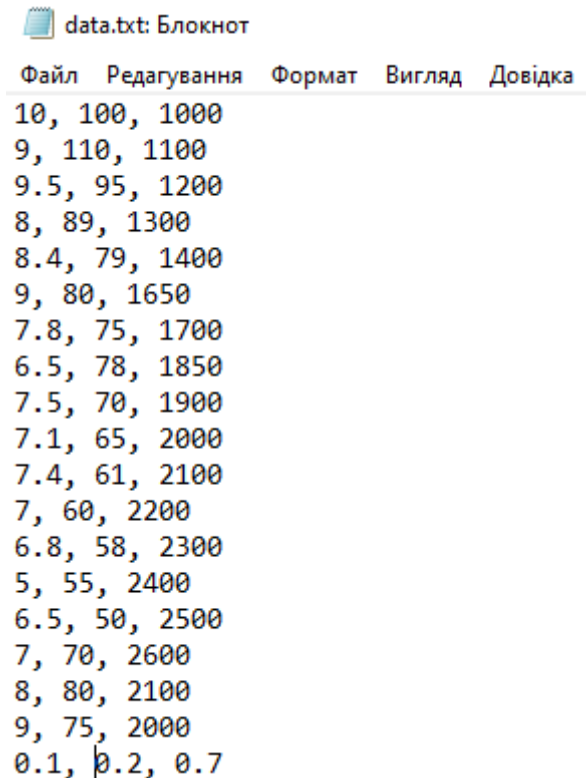
Вартість тис.грн.	Надійність год.	Продуктивність шт./год.	Таблиця 2.		α
10	100	1000		Станок 1	
9	110	1100		Станок 2	
9,5	95	1200		Станок 3	
8	89	1300		Станок 4	
8,4	79	1400		Станок 5	
9	80	1650		Станок 6	
7,8	75	1700		Станок 7	
6,5	78	1850		Станок 8	
7,5	70	1900		Станок 9	
7,1	65	2000		Станок 10	
7,4	61	2100		Станок 11	
7	60	2200		Станок 12	
6,8	58	2300		Станок 13	
5	55	2400		Станок 14	
6,5	50	2500		Станок 15	
7	70	2600		Станок 16	
8	80	2100		Станок 17	
9	75	2000		Станок 18	

Варіант 1

$a_1=0,1$ $a_2=0,2$ $a_3=0,7$

Виконання роботи:

В першу чергу, я записав відповідні дані у текстовий файл data.txt. Після цього програмно зчитав їх для подальшої обробки



data.txt: Блокнот

Файл	Редагування	Формат	Вигляд	Довідка
10, 100, 1000				
9, 110, 1100				
9.5, 95, 1200				
8, 89, 1300				
8.4, 79, 1400				
9, 80, 1650				
7.8, 75, 1700				
6.5, 78, 1850				
7.5, 70, 1900				
7.1, 65, 2000				
7.4, 61, 2100				
7, 60, 2200				
6.8, 58, 2300				
5, 55, 2400				
6.5, 50, 2500				
7, 70, 2600				
8, 80, 2100				
9, 75, 2000				
0.1, 0.2, 0.7				

Рис. 1. Дані, записані у файл data.txt.

Першим кроком є перемноження стовпців таблиці на відповідні коефіцієнти. Код представлений нижче.

```
3 f_string = [i.strip('\n').split(',') for i in open('data.txt')]
4
5 for i in range(0, 18):
6     for j in range(0, 3):
7         f_string[i][j] = float(f_string[i][j]) * float(f_string[18][j])
8
9 f_string.pop(18)
```

Рис. 2. Код, необхідний для множення вагів на значення стовпців.

```
D:\rour4n\education\nulp\7_semestr\bco\lab3>py -3 lab3.py
[1.0, 20.0, 700.0]
[0.9, 22.0, 770.0]
[0.9500000000000001, 19.0, 840.0]
[0.8, 17.8, 909.9999999999999]
[0.8400000000000001, 15.8, 979.9999999999999]
[0.9, 16.0, 1155.0]
[0.78, 15.0, 1190.0]
[0.65, 15.600000000000001, 1295.0]
[0.75, 14.0, 1330.0]
[0.71, 13.0, 1400.0]
[0.7400000000000001, 12.200000000000001, 1470.0]
[0.7000000000000001, 12.0, 1540.0]
[0.68, 11.600000000000001, 1610.0]
[0.5, 11.0, 1680.0]
[0.65, 10.0, 1750.0]
[0.7000000000000001, 14.0, 1819.9999999999998]
[0.8, 16.0, 1470.0]
[0.9, 15.0, 1400.0]
```

Рис. 3. Результат перемноження вагів на значення стовпців.

Наступним кроком є спрощення даної таблиці. Для цього необхідно усунути усі домінантні рядки. Тобто якщо відповідні значення стовпців певного рядка більші за відповідні значення стовпців іншого рядка то цей рядок є домінануючим, і його потрібно залишати. Рядок який є домінантним, необхідно видалити

```
18 for i in range(0, 18):
19     test = a_test[i]
20     for j in range(0, 18):
21         if test[0] > a_test[j][0] and test[1] > a_test[j][1] and test[2] > a_test[j][2]:
22             # f_string.remove(a_test[i])
23             res = a_test[i]
24             res_arr.append(res)
25         else:
26             continue
27
28 res_arr.sort()
29 res_arr = list(res_arr for res_arr, _ in itertools.groupby(res_arr))
30
```

Рис. 3. Код для знаходження домінуючих рядків.

Отримуємо домінантні рядки і заносимо їх в окремий масив і очищуємо від дублікатів.

```
D:\rour4n\education\nulp\7_semestr\bco\lab3>py -3 lab3.py
[0.7000000000000001, 14.0, 1819.9999999999998]
[0.8, 16.0, 1470.0]
[0.9, 15.0, 1400.0]
[0.9, 16.0, 1155.0]
```

Рис. 4. Результат знаходження домінованих рядків.

Після цього усуваємо дані рядки із нашої таблиці і отримуємо таблицю, яка є сформованою. Множиною Паретто, тобто це і є наш результат.

```
D:\rour4n\education\nulp\7_semestr\bco\lab3>py -3 lab3.py
[1.0, 20.0, 700.0]
[0.9, 22.0, 770.0]
[0.9500000000000001, 19.0, 840.0]
[0.8, 17.8, 909.9999999999999]
[0.8400000000000001, 15.8, 979.9999999999999]
[0.78, 15.0, 1190.0]
[0.65, 15.600000000000001, 1295.0]
[0.75, 14.0, 1330.0]
[0.71, 13.0, 1400.0]
[0.7400000000000001, 12.200000000000001, 1470.0]
[0.7000000000000001, 12.0, 1540.0]
[0.68, 11.600000000000001, 1610.0]
[0.5, 11.0, 1680.0]
[0.65, 10.0, 1750.0]
```

Рис. 5. Сформована множина Паретто.

Висновки: Під виконання лабораторної роботи я ознайомився з теоретичними основами побудови множини Парето. Також я виконав програмну реалізацію мовою Python. За допомогою порівняння станків за 3 параметрами, я виключи станки, які по усім трьом параметрам програють іншим станкам. Вхідна вибірка містила 18 станків, а після оптимізації залишилось 14. Дана вибірка і є множиною Паретто.