

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа программной инженерии

Отчет по лабораторной работе

Выполнил студент гр. в3530904/00321 <подпись> В.Я. Копылов
Руководитель <подпись> В.В. Амосов

Санкт-Петербург
2022г.

Задание

Д/З по Критериям

Задача 1

Дано:

1. Матрица доходов (решений)

	R1	R2	R3	R4	R5
X1	15	10	0	-6	17
X2	3	14	8	9	2
X3	1	5	14	20	-3
X4	7	19	10	2	0

Сравните варианты решений по следующим критериям:

1. ММ 2. Севиджа (S) 3. Гурвица (HW), где $C = \dots$

Задача 2

Ежегодный спрос на мерседесы в совместном предприятии в 1 половине года задаётся следующим распределением вероятностей:

X(спрос)	100	150	200	250	300
P(X)	0.2	0.25	0.3	0.15	0.1

Если мерседес не продан в 1 половину года, он может быть реализован за 15 т. \$ к концу года. С другой стороны, мерседесы с конвейера (в 1 половину года) продаются по 49 т. \$ за штуку. Затраты совместного предприятия на 1 мерседес составляют 25 т. \$.

Определите: какое оптимальное число мерседесов целесообразно заказывать ежегодно и какова оптимальная ожидаемая прибыль?

Решение:

Спрос в 1 пол.г. Заказ	100	150	200	250	300
100	2400 т. \$	2400 т. \$	2400 т. \$	2400 т. \$	2400 т. \$
150	1900 т. \$
200
250
300

$$2400 \text{ т. \$} = 49 \text{ т. \$} \cdot 100 - 25 \text{ т. \$} \cdot 100$$

$$1900 \text{ т. \$} = 49 \text{ т. \$} \cdot 100 - 25 \text{ т. \$} \cdot 150 + 15 \text{ т. \$} \cdot 50$$

Задача 1 ($C=0.7$ для критерия Гурвица)

---Вариант решения по ММ (минимаксный критерий)---

множество E вариантов:

-6

+2

-3

+0

Оптимальная стратегия: X2,

значение ММ: 2

---Вариант решения по критерию Севиджа S---

Матрица потерь:

0 9 14 26 0

12 5 6 11 15

14 14 0 0 20

8 0 4 18 17

Матрица наибольших разностей:

26

15

20

18

Оптимальная стратегия: X2

значение критерия Севиджа S: 15

---Вариант решения по критерию Гурвица (HW), где $C=0.7$ ---

Матрица значений:

0.90

5.60

3.90

5.70

Оптимальная стратегия: X4,

значение HW: 5.70

Задача 2.

Вероятности: 0.25, 0.25, 0.35, 0.1, 0.05

Матрица решений:

```
[[2400, 2400, 2400, 2400, 2400],  
 [1900, 3600, 3600, 3600, 3600],  
 [1400, 3100, 4800, 4800, 4800],  
 [900, 2600, 4300, 6000, 6000],  
 [400, 2100, 3800, 5500, 7200]]
```

Мат. ожидания:

2400.0

3175.0

3525.0

3280.0

2865.0

Оптимальное число закупок: 200,

Наибольшее мат. ожидание: 3525.00

Листинг кода:

task1.py

```
import pprint
from copy import deepcopy

matrix = [
    [15, 10, 0, -6, 17],
    [3, 14, 8, 9, 2],
    [1, 5, 14, 20, -3],
    [7, 19, 10, 2, 0]
]

# Минимаксный критерий
print(f'---Вариант решения по ММ (минимаксный критерий)---')
e = list() # множество E вариантов
for idx, str in enumerate(matrix, start=1):
    e.append([min(str), idx])
print('\tмножество E вариантов:')
for el, idx in e:
    print(f'\t{el:+3.0f}')
print(f'\tОптимальная стратегия: X{max(e, key=lambda x:
x[0])[1]},\n\tзначение ММ: {max(e, key=lambda x: x[0])[0]}',
      end='\n\n')

# критерий Севиджа S
print(f'---Вариант решения по критерию Севиджа S---')
Z = deepcopy(matrix) # матрица потерь
Z_col = list()
for i in range(len(Z[0])):
    col = [row[i] for row in Z]
    Z_col.append(col)

for col in Z_col:
    max_el = max(col)
    for i in range(len(col)):
        col[i] = max_el - col[i]

print('\tМатрица потерь:')
for i in range(len(Z_col[0])):
    print('\t', end='')
    for j in range(len(Z_col)):
        print(f'{Z_col[j][i]:2.0f}', end=' ')
    print()

for i in range(len(Z_col[0])):
    for j in range(len(Z_col)):
        Z[i][j] = Z_col[j][i]
```

```

e = [[max(str), idx] for idx, str in enumerate(Z, start=1)] #
матрица наибольших разностей
print('\tМатрица наибольших разностей:')
for el in e:
    print(f'\t{el[0]}')

print(f'\tОптимальная стратегия: X{min(e)[1]}\n\tзначение критерия
Севиджа S: {min(e)[0]}', end='\n\n')

# Критерий Гурвица с C=0.7
print(f'---Вариант решения по критерию Гурвица (HW), где
C=0.7---')
C = 0.7

arr_str = deepcopy(matrix)
arr_col = list()
for i in range(len(arr_str[0])):
    col = [row[i] for row in arr_str]
    arr_col.append(col)

e = [
    [C * min(col) + (1 - C) * max(col), idx] for idx, col in
    enumerate(arr_str, start=1)
]
print('\tМатрица значений:')
for el, idx in e:
    print(f'\t{el:.2f}')
print(f'\tОптимальная стратегия: X{max(e, key=lambda x:
x[0])[1]}\n\tзначение HW: {max(e, key=lambda x: x[0])[0]:.2f}')

```

task2.py

```
import numpy as np
import pprint

prob = [0.25, 0.25, 0.35, 0.1, 0.05]
money = []

for order in [100, 150, 200, 250, 300]:
    row = []
    for demand in [100, 150, 200, 250, 300]:
        if (bad := (order - demand)) <= 0:
            row.append(order * 49 - order * 25)
        else:
            row.append(demand * 49 + bad * 15 - order * 25)
    money.append(row)

print('Матрица решений: ')
pprint.pprint(money)

e = []
for idx, row in enumerate(money):
    acc = 0
    for i in range(len(prob)):
        acc += row[i] * prob[i]
    e.append([acc, idx])

print('Мат. ожидания:')
for el, idx in e:
    print(el)

print(f'Оптимальное число закупое: {[100, 150, 200, 250, 300][max(e, key=lambda x: x[0])[1]}], \nНаибольшее мат. ожидание: {max(e, key=lambda x: x[0])[0]:.2f}')
```