

Министерство образования Российской Федерации

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ
им. Н.Э. БАУМАНА

Факультет: Информатика и системы управления
Кафедра: Информационная безопасность (ИУ8)

Методы оптимизации

Домашнее задание №2 на тему:
«Исследование генетических алгоритмов в задачах
поиска экстремумов»

Вариант 5

Преподаватель:
Коннова Н.С.

Студент:
Девяткин Е.Д.

Группа:
ИУ8-34

Репозиторий работы: <https://github.com/ledibonibell/MO-lab05>

Москва 2023

Цель работы

Ознакомление с основными понятиями «игр с природой»; получение навыков применения различных критериев (Бернулли, Вальда, максимума, Гурвица, Сэвиджа) для выбора стратегии в условиях полной неопределенности.

Постановка задачи

В «игре с природой» вторым игроком является природа, которая действует («выбирает» стратегии) случайным образом (т.е. она может или улучшать положение первого игрока, или ухудшать его). Поэтому существует несколько критериев оценки результатов исследования игровой модели.

Ход работы

Рассмотрим нашу матрицу стратегий (таблица 1):

Стратегии	b_1	b_2	b_3	b_4
a_1	8	12	4	17
a_2	1	6	19	19
a_3	17	11	11	6
a_4	8	10	15	17
a_5	1	16	2	16

Таблица 1.

Критерий Бернулли (принцип недостаточного основания)

Если воспользоваться критерием Бернулли, то следует руководствоваться стратегией a_4 . Соответствующее математическое ожидание выигрыша при этом максимально и равно 12,5

Критерий Вальда (пессимистический)

$$\alpha = \max_i \min_j a_{ij} = 8$$

Пессимистическая стратегия определяет выбор a_4 , где нижняя цена игры будет равна 8

Критерий максимума (оптимистический)

$$a = \max_i \max_j a_{ij} = 19$$

Оптимистическая стратегия соответствует выбору a_4 , где максимально возможный выигрыш равняется 19

Критерий Гурвица

$$a = \max_i \left(\alpha \min_j a_{ij} + (1 - \alpha) \max_j a_{ij} \right) = 12,5$$

Критерий Гурвица определим из условия равновероятной реализации пессимистической и оптимистической гипотез (пусть $\alpha = 0,5$). Наилучшая стратегия a_2 , где ожидаемый выигрыш равен 12,5

Критерий Сэвиджа (критерий рисков)

Составим таблицу рисков (таблица 2)

Стратегии	b_1	b_2	b_3	b_4
a_1	9	4	15	2
a_2	16	10	0	0
a_3	0	5	8	13
a_4	9	6	4	2
a_5	16	0	17	3

Таблица 2.

Исходя из данной таблицы можно сделать следующие выводы для оптимума:

$$c = \min_i \max_j \left(\max_i c_{ij} - c_{ij} \right) = \min[15; 16; 13; 9; 17] = 9$$

Таким образом оптимальная стратегия - a_4

Поиск оптимума

Для удобства получения конечных итогов сделаем еще одну таблицу с результатами вычислений (таблица 3)

Реализуемые методы	Цена игры	Оптимальная стратегия
Бернулли	12,5	a_4
Вальда	8	a_4
Максимума	19	a_4
Гурвица	12,5	a_2
Сэвиджа	9	a_4

Таблица 3.

Вывод

В ходе выполнения работы были проделаны расчеты с игрой с природой для нахождения оптимальной стратегии (оптимальной цены игры) пятью разными способами.

При расчетах, данные показали, что 4 из 5 методов назвали четвертую стратегию оптимальной, из чего можно сделать вывод, что данная стратегия с большей долей вероятности является оптимальной.

Приложение А

Файл 'Main.py':

```
import numpy as np

def bernoulli_criterion(matrix):
    row_with_max_mean = np.argmax(np.mean(matrix, axis=1))
    max_mean = np.max(np.mean(matrix, axis=1))
    return row_with_max_mean + 1, max_mean

def wald_criterion(matrix):
    min_values = np.min(matrix, axis=1)
    max_min_value = np.max(min_values)
    row_with_max_min_value = np.argmax(min_values)
    return row_with_max_min_value + 1, max_min_value

def max_criterion(matrix):
    max_values = np.max(matrix, axis=1)
    max_max_value = np.max(max_values)
    row_with_max_max_value = np.argmax(max_values)
    return row_with_max_max_value + 1, max_max_value

def hurwicz_criterion(matrix, a):
    min_values = np.min(matrix, axis=1)
    max_values = np.max(matrix, axis=1)
    row_with_best_value = np.argmax(a * min_values + (1 - a) * max_values)
    hurwicz = np.max(a * min_values + (1 - a) * max_values)
    return row_with_best_value + 1, hurwicz

def savage_criterion(matrix):
    max_values = np.max(matrix, axis=1)
    min_max_value = np.min(max_values)
    row_with_min_max_value = np.argmin(max_values)
    return row_with_min_max_value + 1, max_values, min_max_value

"""#####
Подсчет значений
#####"""

game_matrix = np.array([[8, 12, 4, 17],
                        [1, 6, 19, 19],
```

```
[17, 11, 11, 6],  
[8, 10, 15, 17],  
[1, 16, 2, 16]])
```

```
risk_table = np.max(game_matrix, axis=0) - game_matrix  
alpha = 0.5
```

```
"#####
```

```
Вывод результатов
```

```
#####"
```

```
print("Начальная матрица стратегий:", "\n", game_matrix)
```

```
print(f"\nМетод Бернулли (принцип недостаточного основания): \nНомер  
стратегии игрока - {bernoulli_criterion(game_matrix)[0]} \nМатематическое  
ожидание - {bernoulli_criterion(game_matrix)[1]}")
```

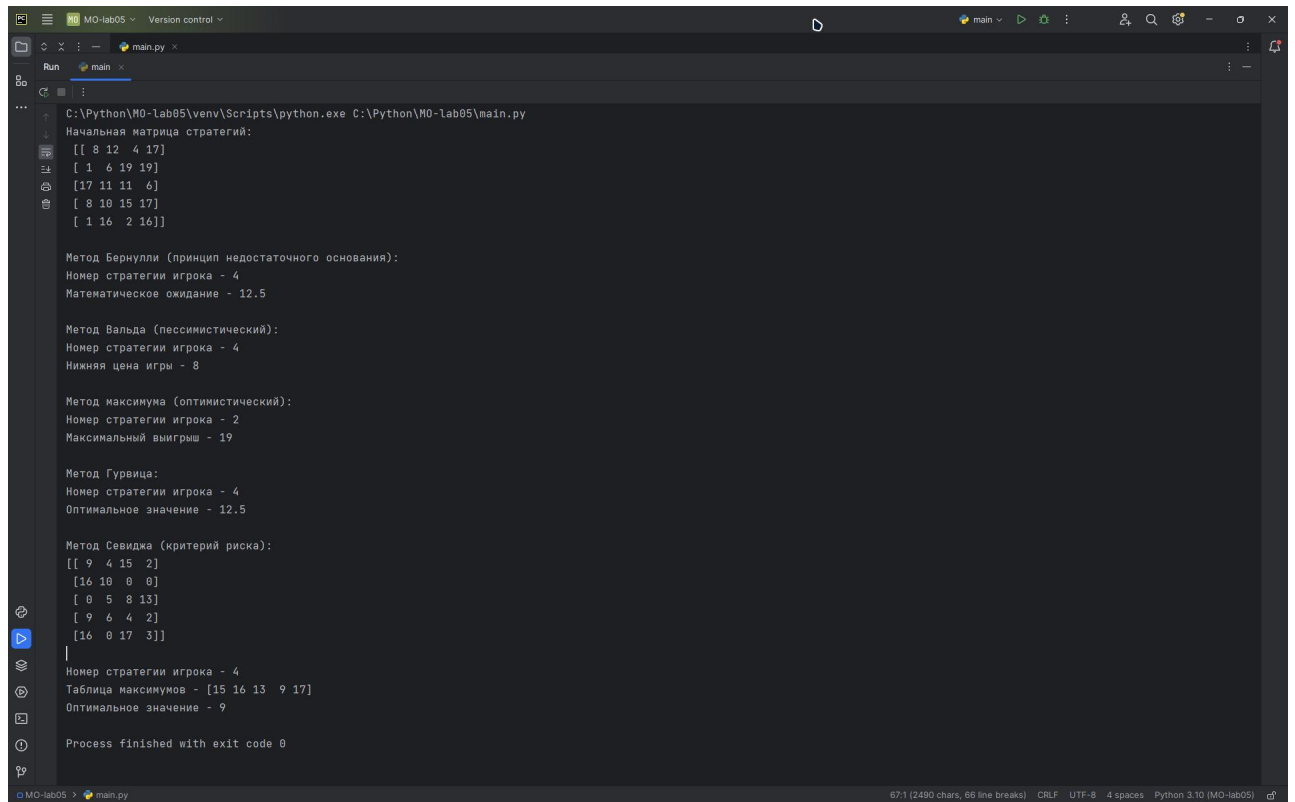
```
print(f"\nМетод Вальда (пессимистический): \nНомер стратегии игрока -  
{wald_criterion(game_matrix)[0]} \nНижняя цена игры -  
{wald_criterion(game_matrix)[1]}")
```

```
print(f"\nМетод максимума (оптимистический): \nНомер стратегии игрока -  
{max_criterion(game_matrix)[0]} \nМаксимальный выигрыш -  
{max_criterion(game_matrix)[1]}")
```

```
print(f"\nМетод Гурвица: \nНомер стратегии игрока -  
{hurwicz_criterion(game_matrix, alpha)[0]} \nОптимальное значение -  
{hurwicz_criterion(game_matrix, alpha)[1]}")
```

```
print(f"\nМетод Севиджа (критерий риска): \n{risk_table} \n\nНомер стратегии  
игрока - {savage_criterion(risk_table)[0]} \nТаблица максимумов -  
{savage_criterion(risk_table)[1]} \nОптимальное значение -  
{savage_criterion(risk_table)[2]}")
```

Приложение Б



```
C:\Python\MO-lab05\venv\Scripts\python.exe C:\Python\MO-lab05\main.py
Начальная матрица стратегий:
[[ 8 12  4 17]
 [ 1  6 19 19]
 [17 11 11  6]
 [ 8 10 15 17]
 [ 1 16  2 16]]

Метод Бернулли (принцип недостаточного основания):
Номер стратегии игрока - 4
Математическое ожидание - 12.5

Метод Вальда (пессимистический):
Номер стратегии игрока - 4
Нижняя цена игры - 8

Метод максимума (оптимистический):
Номер стратегии игрока - 2
Максимальный выигрыш - 19

Метод Гурвица:
Номер стратегии игрока - 4
Оптимальное значение - 12.5

Метод Севиджа (критерий риска):
[[ 9  4 15  2]
 [16 10  0  0]
 [ 0  5  8 13]
 [ 9  6  4  2]
 [16  0 17  3]]

Номер стратегии игрока - 4
Таблица максимумов - [15 16 13  9 17]
Оптимальное значение - 9

Process finished with exit code 0
```

MO-lab05 > main.py 67:1 (2490 chars, 66 line breaks) CRLF UTF-8 4 spaces Python 3.10 (MO-lab05)