



**Министерство образования Российской Федерации  
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ  
им. Н.Э. БАУМАНА**

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

**Теория принятия решений в условиях информационных конфликтов**

**Лабораторная работа №5**

“Матричные игры с нулевой суммой. Смешанные стратегии.”

Вариант 3

**Преподаватель: Коннова Н. С.**

**Студент: Андреев Г.С.**

**Группа: ИУ8-71**

2021 г.

[illegible]

	S0	x1	x2	x7	x5
x4	0.014	0.816	-1	0.075	-0.09
x6	-0.311	-6.953	2	-0.66	-0.028
x3	0.052	0.325	1	-0.057	0.005
x8	-0.575	14.481	-20	1.264	-1.689
x9	0.156	12.976	-8	-0.17	-0.986
F	0.066	0.142	-1	0.019	-0.085

	S0	x9	x2	x7	x5
x4	0.004	-0.063	-0.497	0.086	-0.028
x6	-0.228	0.536	-2.286	-0.751	-0.557
x3	0.048	-0.025	1.201	-0.052	0.029
x8	-0.749	-1.116	-11.072	1.454	-0.589
x1	0.012	0.077	-0.617	-0.013	-0.076
F	0.064	-0.011	-0.913	0.021	-0.074

	S0	x9	x3	x7	x5
x4	0.024	-0.073	0.414	0.064	-0.015
x6	-0.137	0.488	1.904	-0.851	-0.5
x2	0.04	-0.021	0.833	-0.044	0.025
x8	-0.307	-1.347	9.222	0.971	-0.317
x1	0.037	0.064	0.513	-0.04	-0.061
F	0.101	-0.03	0.76	-0.019	-0.052

	S0	x9	x3	x6	x5
x4	0.014	-0.036	0.558	0.076	-0.053
x7	0.16	-0.573	-2.238	-1.175	0.588
x2	0.047	-0.046	0.735	-0.051	0.05
x8	-0.462	-0.79	11.395	1.141	-0.888
x1	0.043	0.041	0.424	-0.047	-0.037
F	0.104	-0.041	0.718	-0.022	-0.041

	S0	x9	x3	x6	x7
x4	0.028	-0.088	0.355	-0.031	0.091
x5	0.273	-0.975	-3.805	-1.998	1.701
x2	0.033	0.003	0.926	0.049	-0.085
x8	-0.22	-1.656	8.016	-0.633	1.51
x1	0.053	0.005	0.282	-0.122	0.064
F	0.115	-0.08	0.563	-0.103	0.069

	S0	x8	x3	x6	x7
x4	0.04	-0.053	-0.072	0.003	0.01
x5	0.402	-0.589	-8.524	-1.625	0.812
x2	0.033	0.002	0.941	0.048	-0.083
x9	0.133	-0.604	-4.839	0.382	-0.912
x1	0.053	0.003	0.305	-0.123	0.068
F	0.126	-0.049	0.174	-0.073	-0.004

[illegible]

	S0	x8	x2	x6	x7
x4	0.043	-0.053	0.077	0.007	0.004
x5	0.7	-0.572	9.061	-1.192	0.064
x3	0.035	0.002	1.063	0.051	-0.088
x9	0.302	-0.594	5.144	0.628	-1.336
x1	0.042	0.002	-0.325	-0.139	0.095
F	0.12	-0.049	-0.185	-0.082	0.011

	S0	x8	x2	x6	x1
x4	0.041	-0.053	0.09	0.012	-0.041
x5	0.672	-0.574	9.279	-1.098	-0.672
x3	0.074	0.004	0.762	-0.078	0.926
x9	0.893	-0.561	0.566	-1.332	14.107
x7	0.443	0.025	-3.426	-1.467	10.557
F	0.115	-0.049	-0.148	-0.066	-0.115

[illegible]

Найдем оптимальную стратегию игрока В

	S0	x1	x2	x3	x4	x5
x6	1	15	12	7	2	3
x7	1	4	10	18	4	15
x8	1	16	13	19	3	19
x9	1	12	1	1	19	12
F	0	-1	-1	-1	-1	-1

## Поиск оптимального решения

Индекс разрешающего элемента:  $(2, 1)$

	S0	x8	x2	x3	x4	x5
x6	0.062	-0.938	-0.188	-10.812	-0.812	-14.812
x7	0.75	-0.25	6.75	13.25	3.25	10.25
x1	0.062	0.062	0.812	1.188	0.188	1.188
x9	0.25	-0.75	-8.75	-13.25	16.75	-2.25
F	0.062	0.062	-0.188	0.188	-0.812	0.188

Индекс разрешающего элемента:  $(3, 4)$

	S0	x8	x2	x3	x9	x5
x6	0.075	-0.974	-0.612	-11.455	0.049	-14.922
x7	0.701	-0.104	8.448	15.821	-0.194	10.687
x1	0.06	0.071	0.91	1.336	-0.011	1.213
x4	0.015	-0.045	-0.522	-0.791	0.06	-0.134
F	0.075	0.026	-0.612	-0.455	0.049	0.078

Индекс разрешающего элемента:  $(2, 2)$

	S0	x8	x1	x3	x9	x5
x6	0.115	-0.926	0.672	-10.557	0.041	-14.107
x7	0.148	-0.762	-9.279	3.426	-0.09	-0.566
x2	0.066	0.078	1.098	1.467	-0.012	1.332
x4	0.049	-0.004	0.574	-0.025	0.053	0.561
F	0.115	0.074	0.672	0.443	0.041	0.893

Оптимальная стратегия игрока А:

$g = 8.714$   
 $x_1 = 0.0$   
 $x_2 = 0.0$   
 $x_3 = 0.643$   
 $x_4 = 0.357$

Оптимальная стратегия игрока В:

$g = 8.714$   
 $y_1 = 0.0$   
 $y_2 = 0.571$   
 $y_3 = 0.0$   
 $y_4 = 0.429$   
 $y_5 = 0.0$

## **Вывод**

В ходе выполнения лабораторной работы была изучена постановка антагонистической игры двух лиц в нормальной форме. В результате работы были найдены стратегии игры для двух игроков. В стратегиях обоих игроков встречаются нулевые значения, что говорит нам о том, что конкретные стратегии не используются.

## Листинг

### main.py

([https://github.com/andreev-g/iu8-decision-theory/blob/master/lab\\_4/main.py](https://github.com/andreev-g/iu8-decision-theory/blob/master/lab_4/main.py))

```
import numpy as np
```

```
from lab_5.simplex import Simplex
```

```
class PlayersGame:
```

```
    def __init__(self, strategy_matrix):
```

```
        self.strategy_matrix = np.array(strategy_matrix)
```

```
        self.result_a = []
```

```
        self.result_b = []
```

```
        self.simplex_table_a = None
```

```
        self.simplex_table_b = None
```

```
        self.b_a = None
```

```
        self.b_b = None
```

```
        self.c_a = None
```

```
        self.c_b = None
```

```
    def preparation_before_simplex(self):
```

```
        self.simplex_table_a = np.transpose(self.strategy_matrix) * (-1)
```

```
        size_b, size_c = self.simplex_table_a.shape
```

```
        self.b_a = np.ones(size_b) * (-1)
```

```
        self.c_a = np.ones(size_c)
```

```
        self.simplex_table_b = self.strategy_matrix
```

```
        size_b, size_c = self.simplex_table_b.shape
```

```
        self.b_b = np.ones(size_b)
```

```
        self.c_b = np.ones(size_c)
```

```
    def preparation_after_simplex(self, result, size):
```

```
        g = 1/result[0]
```

```
        result[0] = g
```

```
        for idx, el in enumerate(result):
```

```
            if idx == 0:
```

```
                continue
```

```
            result[idx] = el * g
```

```
        return result[:size + 1]
```

```
    def simplex_method(self):
```

```
        print("Найдем оптимальную стратегию игрока А")
```

```
        player_a = Simplex(self.simplex_table_a, self.b_a, self.c_a, "min")
```

```
        try:
```

```
            self.result_a = player_a.get_result()
```

```
        except AssertionError:
```

```
            print("Для игрока А решения нет!")
```

```
        self.result_a = self.preparation_after_simplex(self.result_a, self.c_a.size)
```

```
        print("\nНайдем оптимальную стратегию игрока В\n")
```

```
        player_b = Simplex(self.simplex_table_b, self.b_b, self.c_b, "max")
```

```
        try:
```

```
            self.result_b = player_b.get_result()
```

```

except AssertionError:
    print("Для игрока В решения нет!")
    self.result_b = self.preparation_after_simplex(self.result_b, self.c_b.size)

def solve(self):
    self.preparation_before_simplex()
    self.simplex_method()
    self.result_a = [round(x, 3) for x in self.result_a]
    self.result_b = [round(x, 3) for x in self.result_b]
    self.check_solution()
    return self.result_a, self.result_b

def print_solution(self):
    print("Оптимальная стратегия игрока А:")
    print(f"g = {self.result_a[0]}")
    for idx, el in enumerate(self.result_a[1:]):
        print(f"x{idx + 1} = {el}")
    print("\nОптимальная стратегия игрока В:")
    print(f"g = {self.result_b[0]}")
    for idx, el in enumerate(self.result_b[1:]):
        print(f"y{idx + 1} = {el}")

def check_solution(self):
    assert np.sum(self.result_a[1:]) == 1.0, f"Решение игрока А не верно"
    assert np.sum(self.result_b[1:]) == 1.0, f"Решение игрока В не верно"

```