УНИВЕРСИТЕТ ИТМО

Факультет цифровых трансформаций

Отчет по практической работе №4

# Программная реализация алгоритма решения матричной игры в чистых стратегиях путем сведения к задачам линейного программирования

По дисциплине:

***Технологии поддержки принятия решений на финансовых рынках***

Студента группы J42113

**Осадчего Дмитрий**

# Цель работы и постановка задачи



- **Целью** работы является укрепление представления и отработка навыка записи задач линейного программирования, соответствующей заданной платежной матрице игры.

## Постановка задачи

Разработка на языке программирования высокого уровня программы для ЭВМ, которая решает биматричную игру в смешанных стратегиях.

# Алгоритм



- Проверить наличие решения в чистых стратегиях:
  - Игрок 1: найти минимум от максимальных значений  столбцов
  - Игрока 2: найти максимум от минимальных значения строк
  - При совпадении значений вывести чистую стратегию
- Если есть отрицательные элементы привести к каноническому виду платежную матрицу
- Решить 1 ЗЛП

$$W_1 = \sum_{i=1}^{m} u_i \to \min,$$
$$\sum_{i=1}^{m} a_{ij} u_i \geq 1,\ j = \overline{1,n},$$
$$u_i \geq 0,\ i = \overline{1,m}.$$

Решить 2 ЗЛП

$$W_2 = \sum_{j=1}^{n} v_j \to \max,$$
$$\sum_{j=1}^{n} a_{ij} v_j \leq 1,\ i = \overline{1,m},$$
$$v_j \geq 0,\ j = \overline{1,n}.$$

- Найти цену игры $I = \frac{1}{\sum u}$.
- Найти частоты стратегий $p = I * u_i, i = 1..m$. Для q аналогично.
- Восстановить цену игры $I = I - a$

# Пример расчета

```
matrix = np.array([[-2, 6, 7, 8],
                   [-1, 3, -5, 0],
                   [7, 5, 4, -3]])
```

```
Игра не решается в чистых стратегиях.
1player:
MINIMIZE
1*1p_0 + 1*1p_1 + 1*1p_2 + 0
SUBJECT TO
_C1: 4 1p_0 + 5 1p_1 + 13 1p_2 >= 1

_C2: 12 1p_0 + 9 1p_1 + 11 1p_2 >= 1

_C3: 13 1p_0 + 1p_1 + 10 1p_2 >= 1

_C4: 14 1p_0 + 6 1p_1 + 3 1p_2 >= 1

VARIABLES
1p_0 Continuous
1p_1 Continuous
1p_2 Continuous

2player:
MAXIMIZE
1*2p_0 + 1*2p_1 + 1*2p_2 + 1*2p_3 + 0
SUBJECT TO
_C1: 4 2p_0 + 12 2p_1 + 13 2p_2 + 14 2p_3 <= 1

_C2: 5 2p_0 + 9 2p_1 + 2p_2 + 6 2p_3 <= 1

_C3: 13 2p_0 + 11 2p_1 + 10 2p_2 + 3 2p_3 <= 1

VARIABLES
2p_0 Continuous
2p_1 Continuous
2p_2 Continuous
2p_3 Continuous
```

```
Игрок 1:

Сратегия 1  0.5
Сратегия 2  0.0
Сратегия 3  0.5

Игрок 2:

Сратегия 1  0.55
Сратегия 2  0.00
Сратегия 3  0.00
Сратегия 4  0.45

Цена игры: 2.5
```

# Метод Брауна-Робинсона



- Шаг 1: выбрать начальные стратегии по критериям максимизации выигрыша (для 1 игрока) и минимизации проигрыша (для 2 игрока).
  - Создать смешанные стратегии P и Q из выбранных чистых
  - Посчитать критерии эффективности полученных стратегий α и β
- Шаг 2 (итерационный):
  - для каждого игрока найти оптимальную чистую стратегию с учетом предыдущей стратегии противника $P_{prev}$ и $Q_{prev}$.
  - Обновить смешанные стратегии игроков, с учетом выбранных на этом шаге оптимальных стратегий:

$$p_i(k+1) = \begin{cases} \frac{k}{k+1} p_i(k) + \frac{1}{k+1}, \text{если } i = i_{k+1} \\ \frac{k}{k+1} p_i(k), \quad \text{если } i \neq i_{k+1} \end{cases}$$

  - Рассчитать критерии эффективности новых стратегий.
    - Если критерий улучшился то обновить критерий игры
    - Обновить лучшую стратегию
  - Если разница между критериями игры меньше удвоенной заданной точности, то остановиться, иначе повторить шаг 2.

  - Вывести оптимальные стратегии и цену игры, рассчитанную как среднее критериев α и β

# Сравнение методов



## Проверим работу методов на небольшом примере:

```
matrix = np.array([[33, 10, 20, 26.5],
                   [50, 67, 11.5, 25],
                   [23.5, 35, 40, 58.5]])

eps = 0.1
```

## Метод Брауна-Робинсона:

```
Игрок 1:
   Сратегия 1  Сратегия 2  Сратегия 3
0         0.0    0.294118    0.705882
Игрок 2:
   Сратегия 1  Сратегия 2  Сратегия 3  Сратегия 4
0    0.518519         0.0    0.481481         0.0
Цена игры: 31.3785
```

## Метод сведения к ЗЛП:

```
Игрок 1:

Сратегия 1  0.0
Сратегия 2  0.3
Сратегия 3  0.7

Игрок 2:

Сратегия 1  0.52
Сратегия 2  0.00
Сратегия 3  0.48
Сратегия 4  0.00

Цена игры: 31.45
```

Решения совпадают с учетом допустимой погрешности eps = 0.1

# Время работы

В целях сравнения времени работы двух методов, сгенерируем большую 100 на 100 случайную платежную матрицу.

Таблица 1. Время работы при разном диапазоне значений

| Метод\диап. | От -10 до 10 | От -20 до 20 | От -30 до 30 |
|---|---|---|---|
| МСЗЛП | 0.612 с | 0.601 с | 0.597 с |
| МБР | 1.130 с | 6.210 с | 12.50 с |

Таблица 2. Время работы и k при разном eps

| Хар-ка\eps | 0.5 | 0.3 | 0.1 |
|---|---|---|---|
| Время | 0.43 с | 1.16 с | 10.10 с |
| Кол-во итераций | 1352 | 3675 | 31514 |

Заметно, что метод Брауна-Робинсона чувствителен к различным входным характеристикам, а метод сведения к ЗЛП работает более стабильно.

# Вывод



- В ходе выполнения лабораторной работы был реализован алгоритм решения матричных игр путем сведения к двум ЗЛП и поиска оптимальных стратегий.
- При возможности используются чистые стратегии.

# Спасибо за внимание!

Осадчий Дмитрий, группа J42113

ITsMOre than a UNIVERSITY