Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет
по лабораторной работе №2 «Теория игр»
по дисциплине «**Прикладная математика**»

Авторы: Деревицкая Полина Кирилловна M33081
Борздун Анна Вадимовна M33051
Фирсова Дарья Алексеевна M33081
Факультет: ИТиП
Преподаватель: Свинцов Михаил Викторович

ИТМО

Санкт-Петербург

2023

ЦЕЛИ И ЗАДАЧИ

Цель работы: знакомство с теорией игр

Задачи, решаемые во время выполнения работы:

1. Реализация возможности ввода данных из файла в формате JSON, содержащего матрицу игры.
2. Упрощение платежной матрицы путем анализа доминирующих стратегий.
3. Поиск решения игры в чистых стратегиях. Определение оптимальных стратегий и соответствующей цены игры.
4. Применение симплекс метода для поиска седловой точки в смешанных стратегиях, в случае если решения в чистых стратегиях не существует. Определение смешанных стратегий и соответствующей цены игры.

   каждого игрока максимален. Если такие распределения существуют, то цена игры равна ожидаемому выигрышу от этих стратегий.

ТЕОРИЯ

Теория игр – это раздел прикладной математики, в котором рассматриваются методы принятия оптимальных решений (стратегий) в конфликтных ситуациях. Игрой называется упрощенная модель таких ситуаций, которая отличается от реального конфликта тем, что ведется по определенным правилам. Методы и рекомендации теории игр разрабатываются применительно к таким специфическим конфликтным ситуациям, которые обладают свойством многократной повторяемости.

Задачей теории игр является выработка рекомендаций для игроков, т.е. определение для них *оптимальных* стратегий. *Стратегией* игрока называется система правил, однозначно определяющих поведение игрока на каждом шаге в зависимости от ситуации, сложившейся в процессе игры.

Игру можно представить в виде матрицы, в которой строки — стратегии первого игрока, столбцы — стратегии второго игрока, а элементы матрицы — выигрыши первого игрока, Такую матрицу называют платежной. В матричных играх двух лиц с нулевой суммой используются такие понятия, как доминирующая стратегия, оптимальная стратегия, чистая и смешанная стратегии, цена игры и седловая точка.

1. *Доминирующая* стратегия - это такая стратегия игрока, которая приносит ему больший выигрыш по сравнению с другими его стратегиями независимо от выбора стратегии другого игрока.

2. *Оптимальной* называется стратегия, которая при многократном повторении игры обеспечивает данному игроку максимально

возможный средний выигрыш.

3. *Чистая* стратегия даёт полную определённость, каким образом игрок продолжит игру. В частности, она определяет результат для каждого возможного выбора, который игроку придётся сделать. Пространством стратегий называют множество всех чистых стратегий, доступных данному игроку.

4. Если платежная матрица не имеет седловой точки, то поиск решения игры приводит к применению сложной стратегии, состоящей в случайном применении двух и более стратегий с определенными частотами. Такая сложная стратегия называется *смешанной*. Другими словами, *смешанные* стратегии - это стратегии, при которых игрок выбирает разные действия с определенными вероятностями.

5. *Цена* игры - это ожидаемый выигрыш игрока при использовании оптимальных стратегий. Величина α — гарантированный выигрыш, который может обеспечить себе первый игрок, называется *нижней* ценой игры (maxmin). Величина ß— гарантированный проигрыш второго игрока — называется *верхней* ценой игры (minmax). Для матричных игр справедливо неравенство $\alpha \leq$ ß.

6. Если $\alpha =$ ß , то такая игра называется игрой с седловой точкой. Элемент a_{ij} платежной матрицы, соответствующий паре оптимальных стратегий, называется *седловой* точкой матрицы. Иными словами, *седловая* точка в матрице выигрышей - это такая точка, в которой минимальный выигрыш в строке равен максимальному проигрышу в столбце. Если такая точка существует, то цена игры равна значению в этой точке, и игроки выбирают соответствующие стратегии.

ТЕСТОВЫЕ ДАННЫЕ

```json
{
 "matrix": [
      [0, 1, -1],
      [0, -1, 0],
      [1, 0, -1]
 ]
}
```

```json
{
 "matrix": [
      [10, 4, 11, 7],
      [7, 6, 8, 20],
      [6, 2, 1, 11]
 ]
}
```

```json
{
 "matrix": [
      [4, 5, 6, 7],
      [3, 4, 6, 5],
      [7, 6, 10, 8],
      [8, 5, 4, 3]
 ]
}
```

```json
{
 "matrix": [
      [8, 9, 9, 4],
      [6, 5, 8, 7],
      [3, 4, 8, 6],
      [8, 9, 9, 4]
 ]
}
```

```json
{
 "matrix": [
      [7, 6, 5, 4, 2],
      [5, 4, 3, 2, 3],
      [5, 6, 6, 3, 5],
      [2, 3, 3, 2, 4]
 ]
}
```

РЕШЕНИЕ ЗАДАЧ №1 И №2

Данный скрипт осуществляет ввод данных из файла в формате JSON, который содержит матрицу игры. После ввода данных, скрипт упрощает платежную матрицу путем анализа доминирующих стратегий.

```
file_path = '/content/drive/MyDrive/Primat/2023/game_matrix.json'

with open(file_path, 'r') as file:
    data = json.load(file)

matrix = data['matrix']
print_matrix(matrix, "Исходная матрица")
modified_matrix, gamma = add_absolute_min(matrix)
print_matrix(modified_matrix, f"Матрица с добавлением модуля минимального отрицательного числа (gamma={gamma})")
simplified_matrix = dominance_reduction(modified_matrix)
print_matrix(simplified_matrix, "Упрощенная матрица")
```

В данном скрипте реализован алгоритм поиска доминирующих стратегий, который последовательно сравнивает каждую пару строк (или столбцов) и удаляет доминируемые строки (или столбцы). Этот процесс повторяется до тех пор, пока не будет найдена упрощенная матрица, в которой нет доминируемых строк или столбцов.

```
def dominance_reduction(matrix, recursive_iteration=1):
    matrix = np.array(matrix)

    dominant_rows = []
    dominant_cols = []

    for i in range(matrix.shape[1]):
        for j in range(i + 1, matrix.shape[1]):
            if all(matrix[:, i] <= matrix[:, j]):
                dominant_cols.append(j)
            elif all(matrix[:, j] <= matrix[:, i]):
                dominant_cols.append(i)

    for i in range(matrix.shape[0]):
        for j in range(i + 1, matrix.shape[0]):
            if all(matrix[i, :] >= matrix[j, :]):
                dominant_rows.append(j)
            elif all(matrix[j, :] >= matrix[i, :]):
                dominant_rows.append(i)

    result_matrix = np.delete(np.delete(matrix, dominant_rows, axis=0), dominant_cols, axis=1)

    print_matrix(result_matrix, f"Итерация {recursive_iteration}")

    if not np.array_equal(matrix, result_matrix):
        result_matrix = dominance_reduction(result_matrix, recursive_iteration + 1)

    return result_matrix
```

Исходная матрица:

0	1	−1
0	−1	0
1	0	−1

Матрица с добавлением модуля минимального отрицательного числа (gamma=1):

1	2	0
1	0	1
2	1	0

Итерация 1:

2	0
0	1
1	0

Итерация 2:

2	0
0	1

Итерация 3:

2	0
0	1

Упрощенная матрица:

2	0
0	1

Преимущества и недостатки

Основным преимуществом данного подхода является его простота и эффективность. Алгоритм способен быстро находить доминирующие стратегии и упрощать матрицу игры, что упрощает анализ игры и принятие решений.

Однако, этот подход также имеет и недостатки. Во-первых, он основан на предположении, что игроки рациональны и всегда выбирают доминирующую стратегию. На практике это может быть не так, и игроки могут выбирать стратегии по другим причинам, не связанным с максимизацией выигрыша.

Во-вторых, алгоритм не учитывает возможность смешанных стратегий, когда игроки могут выбирать разные стратегии с определенной вероятностью. Это может привести к тому, что некоторые потенциально выгодные стратегии будут пропущены.

РЕШЕНИЕ ЗАДАЧИ №3

Была произведена модификация исходной матрицы путем добавления модуля минимального отрицательного числа. Это делается для того, чтобы все элементы матрицы стали не отрицательными, что облегчает дальнейший анализ.

```
def add_absolute_min(matrix):
    min_negative = None
    for row in matrix:
        for val in row:
            if val < 0 and (min_negative is None or val < min_negative):
                min_negative = val
    if min_negative is not None:
        abs_min = abs(min_negative)
        gamma = abs_min
        modified_matrix = [[element + abs_min for element in row] for row in matrix]
        return modified_matrix, gamma
    else:
        return matrix, 0
```

```
Матрица с добавлением модуля минимального отрицательного числа (gamma=1):
```

1	2	0
1	0	1
2	1	0

```
Минимальные значения строк: (0, 0, 0)
Максимальные значения столбцов: (2, 2, 1)
Нижняя цена (максимин): 0
Верхняя цена (минимакс): 1

Решение в чистых стратегиях невозможно.
```

Затем были найдены минимальные значения по строкам и максимальные значения по столбцам. Эти значения используются для определения нижней и верхней цены игры.

РЕШЕНИЕ ЗАДАЧИ №4

Для решения задачи о нахождении седловой точки в смешанных стратегиях можно использовать симплекс-метод. В этом случае стратегии игроков представляются в виде вероятностных распределений на множестве их чистых стратегий, и задача сводится к поиску таких распределений, при которых ожидаемый выигрыш каждого игрока максимален. Если такие распределения существуют, то цена игры равна ожидаемому выигрышу от этих стратегий.

В ходе выполнения задания мы столкнулись с ситуацией, когда решение в чистых стратегиях было невозможно. Это означает, что игрокам пришлось использовать смешанные стратегии, то есть изменять свою стратегию в ходе игры. Для определения этих стратегий и соответствующей цены игры применяется симплекс-метод.

В контексте теории игр седловая точка обозначает равновесие, в котором ни один из игроков не может улучшить свой результат, изменяя свою стратегию при условии, что стратегия другого игрока остается неизменной.

В ходе выполнения задания мы применили симплекс-метод для поиска оптимальных значений переменных для каждого из игроков и соответствующей цены игры. Цена игры в данном контексте - это значение функции полезности (выигрыша) в равновесии Нэша.

Исходная матрица:

0	1	−1
0	−1	0
1	0	−1

Матрица с добавлением модуля минимального отрицательного числа (gamma=1):

1	2	0
1	0	1
2	1	0

Математическая модель для игрока A -> min:

1	1	2	1
2	0	1	1
0	1	0	1

Математическая модель для игрока B -> max:

1	2	0	1
1	0	1	1
2	1	0	1

```
Оптимальные значения переменных игрока A: [0.3333333333333333, 0.6666666666666666, 0]
Цена игры: −0.33333333333333337
```

```
Оптимальные значения переменных игрока B: [0, 0.3333333333333333, 0.6666666666666666]
Цена игры: −0.33333333333333337
```

ВЫВОД

В ходе выполнения данной лабораторной работы мы познакомились с теорией игр и ее применением на практике. Мы реализовали ввод данных из файла в формате JSON, содержащего матрицу игры, что позволило нам работать с различными сценариями игр.

Одной из ключевых задач было упрощение платежной матрицы путем анализа доминирующих стратегий. Это позволило нам определить стратегии, которые являются наилучшими для каждого из игроков и могут быть использованы ими для достижения максимального выигрыша.

Также мы искали решение игры в чистых стратегиях. В случае, когда решение в чистых стратегиях невозможно, мы применяли симплекс-метод для поиска седловой точки в смешанных стратегиях. Этот метод позволил нам определить оптимальные значения переменных для каждого из игроков и соответствующую цену игры.

В результате выполнения работы мы получили практические навыки применения теории игр и алгоритмов оптимизации в задачах принятия решений. Эти навыки могут быть полезны в различных областях, где присутствует конкуренция и необходимо принимать оптимальные решения.