Отчет по практическому заданию2

Задание состоит в численном решении антагонистической матричной игры

Команда:

Раева Анастасия - 312 группа

Богданова Елена – 312 группа

Чистяков Иван - 312 группа

Постановка Задачи.

Рассматривается G - антагонистическая игра двух лиц A и B. Игроки имеют противоположные интересы: выигрыш одного равен проигрышу другого. Иными словами, в численном описании выигрышей игроков, выигрыш игрока A равен выигрышу игрока B, взятым с противоположным знаком. Поэтому нам достаточно рассмотреть матрицу, описывающую выигрыш, только одного из игроков. Ввиду сказанного, естественно, игрок A хочет максимизировать, а B - минимизировать свой выигрыш.

Пусть игрок A имеет I, а B – J стратегий. Тогда составляется прямоугольная таблица размера I x J, в которой столбцы соответствуют стратегиям игрока B (их J штук), а строки – A (их I штук). Элементы таблицы – значения выигрыша в ситуации(ситуацией называется пара (i,j), где i – некоторая стратегия из I, j – из J).

Если составлена такая таблица, то говорят что игра G приведена к матричной форме.

Предполагается что игрок может выбрать только одну стратегию из конечного множества его стратегий и придерживаться ее в течении игры.

Наша задача состоит в поиске цены игры и оптимальных стратегий для игроков A и B. Оптимальными называем стратегии, от которых игрокам отклоняться не выгодно, а ценой игры их средний выигрыш при условии что они придерживаются своих оптимальных стратегий.

Поставленная задача может иметь решение в чистых стратегиях или в смешанных (зависит от особенностей игры). В первом случае существует пара стратегий игроков A и B (то есть ситуация равновесная по Нэшу, которую называют седловой точкой), которая обеспечивает выигрыш обоим игрока, эта ситуация является оптимальной для них. Во втором случае стратегии игроков описываются векторами вероятности выбора отдельных стратегий.

Подход к решению задачи.

1 Проверяем, имеет ли матрица выигрыша седловую точку. Для этого нужно сравнить максимальное значение из минимумов элементов, взятых по каждой строке с минимумом из максимальных значений среди столбцов. Если они совпадают - то полученное значение равняется выигрышу(или как его еще называют, значению игры), а номера соответствующих строки и столбца – номера стратегий, которые являются чистыми стратегииями(мы используем нумерацию стратегий начиная с нуля). Для поиска седловой точки реализована функция saddle(a), принимающая на вход матрицу игры.

2 Если седловой точки нет - находим решение игры в смешанных стратегиях. Решаем задачу линейного программирования, используя симплекс – метод.

3 Полагаем, что игроку с большим числом стратегий соответствуют столбцы матрицы игры, а с меньшим, соответственно, строки. Если количество стратегий игроков одинаковое, то особых предположений по поводу матрицы игры не делаем(это предположение связано с особенностью реализации симплекс-метода). Иными словами, рассматриваем матрицы квадратные или такие, в которых строк больше, чем столбцов.

4 Реализация симплекс-метода. Рассмотрим на примере, чтобы сделать объяснение более наглядным.

Симплекс – метод:

Имеем матрицу игры (взяли в качестве примера некоторую матрицу)

Игроки	B_1	B_2	B_3	B_4	B_5	B_6
A_1	4	0	6	2	2	1
A_2	3	8	4	10	4	4
A_3	1	2	6	5	0	0
A_4	6	6	4	4	10	3
A_5	10	4	6	4	0	9
A_6	10	7	0	7	9	8

Решим прямую задачу линейного программирования симплексным методом, с использованием симплексной таблицы.

Определим максимальное значение целевой функции $Z(Y) = y_1+y_2+y_3+y_4+y_5+y_6$ при следующих условиях-ограничений.

$4y_1+6y_3+2y_4+2y_5+y_6 \le 1$

$3y_1+8y_2+4y_3+10y_4+4y_5+4y_6 \le 1$

$y_1+2y_2+6y_3+5y_4 \le 1$

$6y_1+6y_2+4y_3+4y_4+10y_5+3y_6 \le 1$

$10y_1+4y_2+6y_3+4y_4+9y_6 \le 1$

$10y_1+7y_2+7y_4+9y_5+8y_6 \le 1$

Для построения первого опорного плана систему неравенств приведем к системе уравнений путем введения дополнительных переменных (переход к канонической форме).

$4y_1+6y_3+2y_4+2y_5+y_6+y_7 = 1$

$3y_1+8y_2+4y_3+10y_4+4y_5+4y_6+y_8 = 1$

$y_1+2y_2+6y_3+5y_4+y_9 = 1$

$6y_1+6y_2+4y_3+4y_4+10y_5+3y_6+y_{10} = 1$

$10y_1+4y_2+6y_3+4y_4+9y_6+y_{11} = 1$

$10y_1+7y_2+7y_4+9y_5+8y_6+y_{12} = 1$

Решим систему уравнений относительно базисных переменных: y_7, y_8, y_9, y_{10}, y_{11}, y_{12}

Число дополнительных переменных равняется числу строк в матрицы игры.

Полагая, что свободные переменные(те что не являются дополнительными) равны 0, получим первый опорный план:

Y0 = (0,0,0,0,0,0,1,1,1,1,1,1)

Симплекс таблица построенная для нишей игры имеет вид:

Базис	B	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}	y_{12}
y_7	1	4	0	6	2	2	1	1	0	0	0	0	0
y_8	1	3	8	4	10	4	4	0	1	0	0	0	0
y_9	1	1	2	6	5	0	0	0	0	1	0	0	0
y_{10}	1	6	6	4	4	10	3	0	0	0	1	0	0
y_{11}	1	10	4	6	4	0	9	0	0	0	0	1	0
y_{12}	1	10	7	0	7	9	8	0	0	0	0	0	1
Z(Y0)	0	-1	-1	-1	-1	-1	-1	0	0	0	0	0	0

Далее переходим к основному алгоритму симплекс-метода:

Наша задача: итерационно преобразовать симплекс-таблицу к виду, когда опорный план не содержит отрицательных элементов(иными словами, индексная

строка (последняя строка в нашей симплекс-таблице) не содержит отрицательных элементов). Если содержит, то опорный план называем неоптимальным.
Итерационный процесс:

Просматриваем последнюю (индексную) строку, если среди значений индексной строки нет отрицательных, то эта таблица определяет оптимальный план задачи, если отрицательный элемент найден, то:

1. Выполняется функция find_min_in_last_string(a), принимающая на вход симплекс-таблицу и возвращающая номер столбца, в котором находится минимальный элемент.
2. Стоим базисный вектор коэффицентов. Размерность базисного вектора равна числу добавленных переменных. Изначально в нем содержатся индексы базисных переменных, по мере изменения симплекс – таблицы, значения базисного вектора меняются.
3. В столбце, номер которого найден в п.1, ищем минимальное частное от деления элемента первого столбца(столбец B в обозначениях симплекс-таблицы) на элемент найденного столбца, расположенный в той же строк. Выполняем для каждой строки кроме последней. Это реализует функция minD_string(res, column), принимающая симплекс-таблицу и номер столбца из п.1 и возвращающая номер строки искомого элемента
4. Запускаем алгоритм изменения симплекс-таблицы - функция changing(res), принимающая симплекс-таблицу, результаты п.1 и п.3

1) Меняем все элементы кроме столбца и строки из п.1 и п.3 по правилу прямоугольника - формула a[i][j] = (a[i][j])-((a[r][j])*(a[i][s])/(a[r][s])) (r - строка из п.3, s - столбец из п.1) с помощью функции change_others(res), принимающей симплекс-таблицу, r, s и возвращающей измененную таблицу
2) Изменяем столбец(r) из п.1 - все элементы, кроме соответствующего строке(s) из п.3, обнуляются с помощью функции change_col, принимающей симплекс-таблицу, r, s и возвращающей измененную таблицу
3) Изменяем строку (s) из п.3 - все элементы делим на элемент этой же строки, соответствующий столбцу(r) из п.1, с помощью функции change_string, принимающей симплекс-таблицу, r, s и возвращающей измененную таблицу

5. Изменяем базисный вектор - элементу, соответствующему номеру строки из п.3 присваиваем значение столбца из п.1
6. Итерационный процесс останавливается, когда индексная строка не содержит отрицательных элементов.

Окончательный вариант симплекс-таблицы(для взятого примера):

Базис	B	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}	y_{12}
y_7	$^{23}/_{906}$	$^{305}/_{906}$	$-^{166}/_{151}$	0	0	0	0	1	$^{139}/_{302}$	$-^{391}/_{453}$	$^{-58}/_{151}$	$^{-85}/_{453}$	0
y_4	$^{9}/_{302}$	$^{-117}/_{302}$	$^{94}/_{151}$	0	1	0	0	0	$^{45}/_{302}$	$^{-2}/_{151}$	$^{-9}/_{151}$	$^{-7}/_{151}$	0
y_{12}	$^{297}/_{604}$	$^{1575}/_{604}$	$-^{412}/_{151}$	0	0	0	0	0	$-^{629}/_{604}$	$^{387}/_{302}$	$^{-73}/_{151}$	$^{-40}/_{151}$	1
y_3	$^{257}/_{1812}$	$^{887}/_{1812}$	$^{-28}/_{151}$	1	0	0	0	0	$^{-75}/_{604}$	$^{161}/_{906}$	$^{15}/_{302}$	$^{35}/_{906}$	0
y_6	$^{1}/_{302}$	$^{289}/_{302}$	$^{44}/_{151}$	0	0	0	1	0	$^{5}/_{302}$	$^{-17}/_{151}$	$^{-1}/_{151}$	$^{16}/_{151}$	0
y_5	$^{55}/_{1812}$	$^{493}/_{1812}$	$^{51}/_{151}$	0	0	1	0	0	$^{-9}/_{604}$	$^{-29}/_{906}$	$^{16}/_{151}$	$^{-13}/_{453}$	0
Z(Y6)	$^{31}/_{151}$	$^{50}/_{151}$	$^{10}/_{151}$	0	0	0	0	0	$^{4}/_{151}$	$^{3}/_{151}$	$^{27}/_{302}$	$^{21}/_{302}$	0

Оптимальный план прямой задачи составляется на основе значений в базисном векторе. Переменная входит в оптимальный план если она не являлась дополнительной. Для таких переменных, если их индексы присутствуют в базисном векторе, значением является число из столбца B, находящееся с переменной в одной "строке".

Если не дополнительной переменной нет в базисе, ее значение принимается равным нулю.

Для нашего примера оптимальный план прямой задачи можно записать так:

$y_1 = 0$, $y_2 = 0$, $y_3 = {}^{257}/_{1812}$, $y_4 = {}^{9}/_{302}$, $y_5 = {}^{55}/_{1812}$, $y_6 = {}^{1}/_{302}$

Оптимальный план двойственной задачи составляется из добавочных переменных. Значения добавочных переменных определятся их окончательного варианта симплекс таблицы. Им соответствуют значения последней строки таблицы. Таким образом, оптимальный план двойственной задачи:

$x_1=0$, $x_2={}^{4}/_{151}$, $x_3={}^{3}/_{151}$, $x_4={}^{27}/_{302}$, $x_5={}^{21}/_{302}$, $x_6=0$

Ценой игры является значение обратное к значению в последней строке в столбце B. То есть, для нашего примера цена игры $g = 1/(31/151) = 151/31$

Найдем оптимальные смешанные стратегии игроков.

Смешанная стратегия игрока – вектор вероятностей применения стратегий. Он получается из вектора оптимального плана игрока умножением каждой компоненты на цену игры. Таким образом, для первого игрока имеем:

$p_1 = {}^{151}/_{31} * 0 = 0$

$p_2 = {}^{151}/_{31} * {}^{4}/_{151} = {}^{4}/_{31}$

$p_3 = {}^{151}/_{31} * {}^{3}/_{151} = {}^{3}/_{31}$

$p_4 = {}^{151}/_{31} * {}^{27}/_{302} = {}^{27}/_{62}$

$p_5 = {}^{151}/_{31} * {}^{21}/_{302} = {}^{21}/_{62}$

$p_6 = {}^{151}/_{31} * 0 = 0$

Оптимальная смешанная стратегия игрока I: $P = (0; {}^{4}/_{31}; {}^{3}/_{31}; {}^{27}/_{62}; {}^{21}/_{62}; 0)$

Для второго :

$q_1 = {}^{151}/_{31} * 0 = 0$

$q_2 = {}^{151}/_{31} * 0 = 0$

$q_3 = {}^{151}/_{31} * {}^{257}/_{1812} = {}^{257}/_{372}$

$q_4 = {}^{151}/_{31} * {}^{9}/_{302} = {}^{9}/_{62}$

$q_5 = {}^{151}/_{31} * {}^{55}/_{1812} = {}^{55}/_{372}$

$q_6 = {}^{151}/_{31} * {}^{1}/_{302} = {}^{1}/_{62}$

Оптимальная смешанная стратегия игрока II: $Q = (0; 0; {}^{257}/_{372}; {}^{9}/_{62}; {}^{55}/_{372}; {}^{1}/_{62})$

Цена игры $g = 151/31$

Техническое описание:
Модуль start_nash
Используется для запуска функции nash_equilibrium(a) с параметром a – матрица выигрыша.

Модуль: nash
В нем реализована функция nash_equilibrium(a), которая принимает на вход матрицу выигрыша и возвращает значение игры и оптимальные стратегии играков

Модуль plot
Иллюстрирует работу кода путем визуализации спектров оптимальных стратегий играков

Модуль unit_tests
Набор тестов, покрывающий все множество возможных матриц выигрыша в наших предположениях. Разработан для тестирования функции nash_equilibrium(a) из модуля nash. Проверяет работу этой функции для игр:

1) спектр оптимальной стратегии состоит из одной точки (т.е. существует равновесие Нэша в чистых стратегиях)
2) спектр оптимальной стратегии неполон (т.е. некоторые чистые стратегии не используются)
3) спектр оптимальной стратегии полон

Решение оформлено в виде пакета: nash_equilibrium_pkg
Для установки пакета прочитать README.md

Инструкции по запуску: выполнить start_nash.py
(по умолчанию в качестве параметра в функцию nash_equiliblium(a) педерается матрица
```
a = ([
    [4,0,6,2,2,1],
    [3,8,4,10,4,4],
    [1,2,6,5,0,0],
    [6,6,4,4,10,3],
    [10,4,6,4,0,9],
    [10,7,0,7,9,8]])
)
```

чтобы изменить матрицу a:
в любом текстовом редакторе открыть start_nash.py
вместо a ввести другую матрицу в формате
```
a = np.array( [[…],
              […],
              …
              […]])
```

Важно, что если матрица неквадратная, то строк должно быть больше, чем столбцов. Для квадратных матриц особых указаний нет.

Состав команды и вклад:

Богданова Елена - функция поиска седловой точки, поиск оптимальных смешанных стратегий по окончательной симплекс-таблице, работа с базисным вектором.
(функции:
saddle(…)
second_gamer(…)
first_gamer(…)
get_vector(…)
nash_equilibrium(…)
). Создание пакета nash_equilibrium_pkg, составление отчета.

Раева Анастасия - итерации симплекс-метода
(функции:
minD_string(…);
change_string (…);
change_others (…);
change_col (...);
changing (…);
simptab (...);
find_min_in_last_string(a)), составление отчета.

Чистяков Иван – графическое изображение оптимальных стратегий играков, создание unit тестов, составление отчета.
(модули:
plot
unit_testes
)

Иллюстрация работы:

Рассмотрим работу программы на примере различных матриц выигрыша a;

1) Решение в смешанных стратегиях

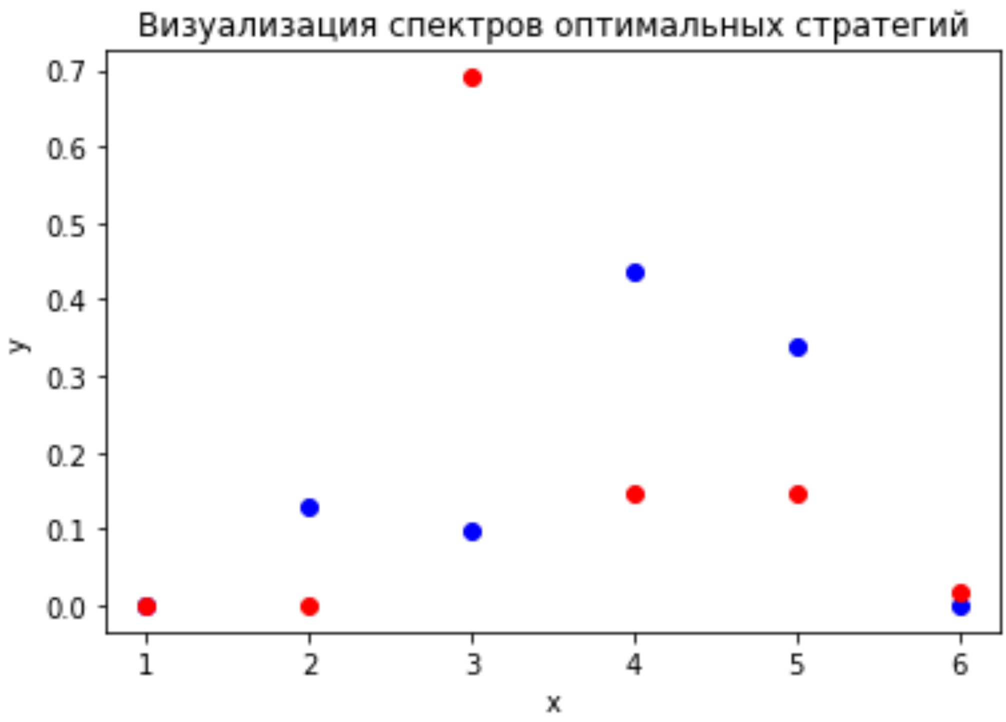
```
a = ([
    [4,0,6,2,2,1],
    [3,8,4,10,4,4],
    [1,2,6,5,0,0],
    [6,6,4,4,10,3],
    [10,4,6,4,0,9],
    [10,7,0,7,9,8]])
```

Значение игры:
game price = 4.870967741935484

Векторы вероятностей второго и первого игроков

p = [0.0,
0.12903225806451613,
0.09677419354838711,
0.435483870967742,
0.33870967741935487,
0.0]

q = [0.0,
0.0,
0.6908602150537634,
0.14516129032258068,
0.14784946236559143,
0.01612903225806453]



2) Смешанные стратегии

```
a = np.array([
        [10, 30],
        [40, 20]
```

Значение игры:

```
game price =  25.0
```

Векторы вероятностей:

```
p =  [0.5, 0.5]
q =  [0.25, 0.75]
```

3) Смешанные стратегии

```
a = np.array([
    [5,3,4],
    [1,2,5],
    [0,6,10],
    [0,4,10]])
```

Значение игры

```
game price =  3.75
```

Векторы вероятности

```
 p =  [0.75, 0.0, 0.25, 0.0]
q =  [0.375, 0.625, 0.0]
```

4)Есть седловая точка

```
a = np.array([
        [3,9,2,1],
        [7,8,5,6],
        [4,7,3,5],
        [5,6,1,7]])
```

Значение игры

Game price: 5 ;

Стратегии(нумерация стратегий начинается с нуля)

Strategies: 1 2

5) Есть седовая точка

```
a = np.array([
        [2,7,2],
        [2,3,2],
        [2,1,8]]
```

Значение игры

```
Game price:  2;
```

Стратегии

```
Strategies: 1   0
```

6) Смешанные стратегии

```
a = np.array([
    [9,3,4],
    [1,7,0],
    [0,0,3],
    [0,4,9]])
```

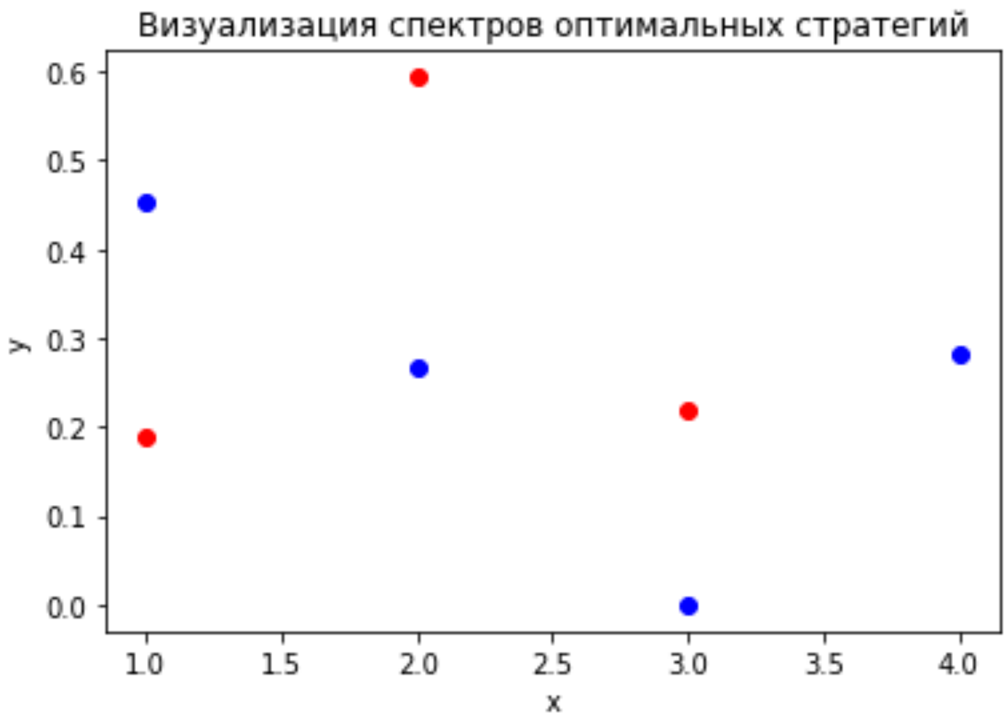
Значение игры

game price = 4.34375

Векторы вероятности выбора стратегии

p = [0.45312499999999994,
0.265625,
0.0,
0.28125]

q = [0.1875,
0.5937499999999999,
0.21875000000000003]



Литература:

См. Васин А.А., Краснощеков П.С., Морозов В.В. *Исследование операций* - М.: Издательский центр “Академия”.