Московский государственный университет
имени М. В. Ломоносова
Факультет вычислительной математики и кибернетики

Отчет по заданию №1

«Численное решение антагонистической матричной игры»

Выполнили:
Шумилин Я. Т. 312гр.
Рзянина А. Т. 312гр.
Долгая Л. В. 311гр.

Москва
2019

Содержание

Постановка задачи

Антагонистическая игра — некооперативная игра, в которой участвуют два или более игроков, выигрыши которых противоположны.
Система $\Gamma = \langle X, Y, F(x, y)\rangle$, где X и Y непустые множества стратегий первого и второго игрока, а функция $F : X \times Y \to R$, называется антагонистической игрой двух лиц в нормальной форме.
Элементы $x \in X$ и $y \in Y$ называются стратегиями игроков 1 и 2 соответственно в игре Γ, элементы произведения множеств $(x, y) \in X \times Y$ – ситуациями, а функция F - функцией выигрыша игрока 1. Выигрыш игрока 2 в ситуации (x, y) полагается равным $(-F(x, y))$, поэтому функция F называется функцией выигрыша самой игры Γ.
Пара $(x^0, y^0) \in X \times Y$ называется седловой точкой функции $F(x, y)$ на $X \times Y$, если $\forall x \in X, \forall y \in Y \ F(x, y^0) \leq F(x^0, y^0) \leq F(x^0, y)$.
Говорят, что антагонистическая игра Γ имеет решение, если функция $F(x, y)$ имеет на $X \times Y$ седловую точку.
Пусть (x^0, y^0) - седловая точка. Тройка $(x^0, y^0, v = F(x^0, y^0))$ называется решением игры, а x^0, y^0 - оптимальными стратегиями игроков 1 и 2, а v - значением игры.
Антагонистическая игра называется матричной, если множества стратегий игроков конечны: $X = 1, \ldots, m$, $Y = 1, \ldots, n$. При этом стратегии первого игрока принято обозначать через i, а стратегии второго через j, а выигрыш первого $F(i, j)$ через a_{ij}
Матрица $A = (a_{ij})_{m \times n}$ называется матрицей игры, в которой первый игрок выбирает номер строки, а второй - номер столбца.
Антагонистическая игра $\overline{\Gamma} = \langle\{\varphi\}, \{\psi\}, F(\varphi, \psi)\rangle$ называется смешанным расширением игры Γ.
Смешанной стратегией первого игрока в игре Γ называется вероятностное распределение φ на множестве стратегий X. Смешанной стратегией второго игрока в игре Γ называется вероятностное распределение ψ на множестве стратегий Y.
В матричной игре для обозначения смешанной стратегии будем использовать вероятностный вектор $p = (p_1, \ldots, p_m), (q = (q_1, \ldots, q_n))$, удовлетворяющий ограничениям $\sum_{i=1}^{m} p_i = 1, p_i \geq 0, i = 1, \ldots, m$. Если применяется вектор p, то стратегия i выбирается с вероятностью p_i.
$\overline{\Gamma} = \langle\{\varphi\}, \{\psi\}, F(\varphi, \psi)\rangle = \langle P, Q, A(p, q)\rangle$ - смешанное расширение матричной игры Γ, где P и Q - множества всевозможных смешанных стратегий первого и второго игроков, а $A(p, q) = \sum_{i=1}^{m} \sum_{j=1}^{n} p_i a_{ij} q_j$ - математическое ожидание выигрыша первого игрока.

Теорема[1]
Для того чтобы ситуация (p^0, q^0) была ситуацией равновесия в смешанных стратегиях игры $\overline{\Gamma}$, необходимо и достаточно, чтобы нашлись множества $X^0 \subseteq X$, $Y^0 \subseteq Y$ и числа v_1, v_2, для которых выполнены условия

$$\begin{cases} \sum_{j \in Y^0} a_{ij} q_j^0 = v_1, i \in X^0, \\ \sum_{j \in Y^0} a_{ij} q_j^0 \leq v_1, i \notin X^0, \\ \sum_{j \in Y^0} q_j^0 = 1, q_j^0 \geq 0, j \in Y^0 \end{cases}$$

$$\begin{cases} \sum_{i \in X^0} p_i^0 b_{ij} = v_2, j \in Y^0, \\ \sum_{i \in X^0} p_i^0 b_{ij} \leq v_2, j \notin Y^0, \\ \sum_{i \in X^0} p_i^0 = 1, p_i^0 \geq 0, i \in X^0 \end{cases}$$

Причем $b_{ij} = -a_{ij}$.
Выполнение условий данной теоремы обуславливает переход к задаче линейного программирования.

Метод решения

Описание программы

- nash_equilibrium принимает матрицу выигрышей, возвращает решения и цену игры
- clean_str принимает матрицу выигрышей, возвращает номера стратегий соответсвующих седловой точке или (-1, -1) в случае отсутствия седловой точки
- simplex_matrix принимает матрицу выигрышей, возвращает начальную симплексную таблицу, вектор базисных переменных и вектор опорного плана
- check_z принимает вектор опорного плана и проверяет план на оптимальность, возвращает номер ведущего столбца либо -1 в случае оптимальности плана
- get_pivot принимает симплексную таблицу и номер опорного столбца, производит поиск ведущего элемента в ведущем столбце, возвращает номер строки
- change_base принимает вектор базисных переменных и номера стратегий соответствующих ведущему элементу, произвводит смену базисных переменных
- change_table принимает симплексную таблицу, вектор базисных переменных, вектор опорного плана, номера стратегий соответствующих ведущему элементу, выполняет перестроение симплексной таблицы по методу Жордана-Гаусса
- get_res принимает симплексную таблицу, вектор базисных переменных, вектор опорного плана, векторы p и q, которые заполняет для получения спектров оптимальных смешанных стратегий
- print_res принимает векторы спектров смешанных стратегий p и q и строит их графики

Ход решения

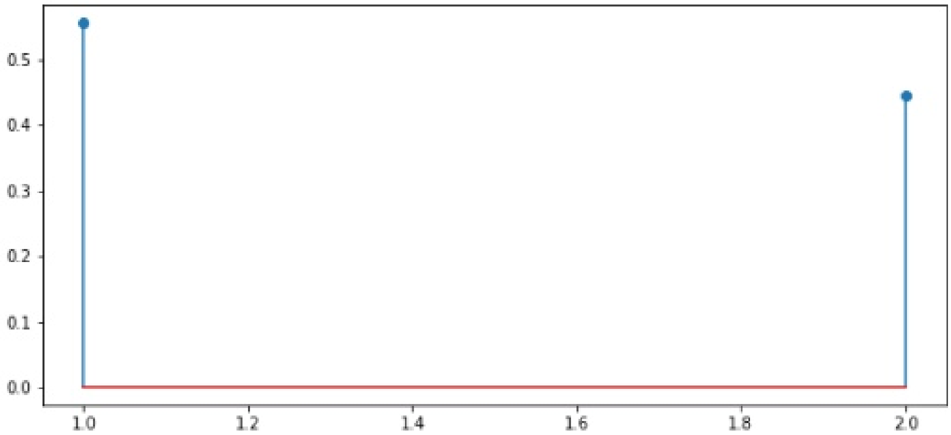
Функция nash_equilibrium получает на вход исходную матрицу выигрышей. Выполняется проверка на наличие чистых стратегий (поиск седловой точки) с помощью функции clean_str.
Если решение в чистых стратегиях не существует, то выполняется поиск решения в смешанных стратегиях симплекс методом, иначе вывод решения в чистых стратегиях и графиков спектров.
На основе исходной матрицы выигрышей строится симплексная таблица и запускается итерационный симплекс-метод.

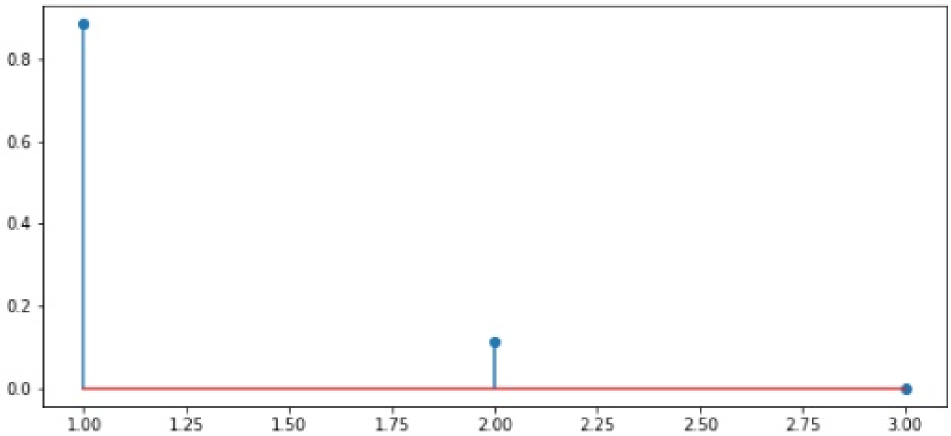
1. Выполняется проверка плана на оптимальность
2. Улучшение опорного плана
3. Пересчет симплексной таблицы

Когда достигнут оптимальный план, производится вывод решений и графиков спектров.

Тестирование

1) $\begin{pmatrix} 4 & 0 & 6 \\ 3 & 8 & 4 \end{pmatrix}$
value = 3.5(5)
p: [0.556, 0,444]
q: [0.889, 0.111, 0]



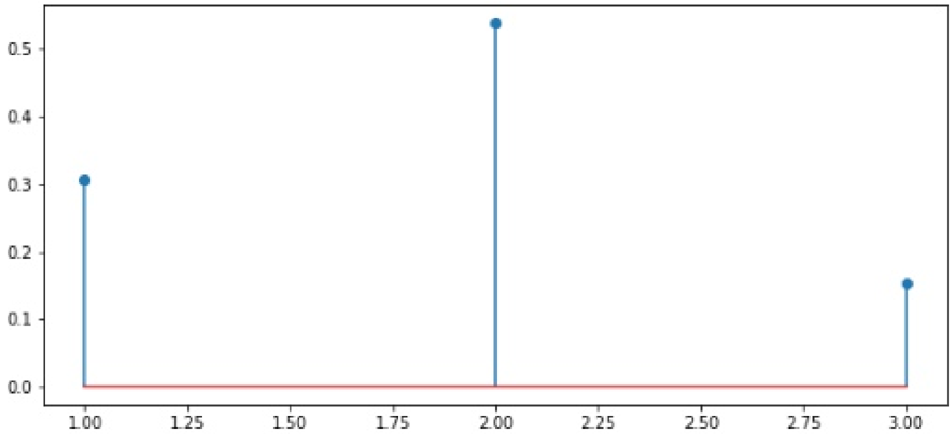


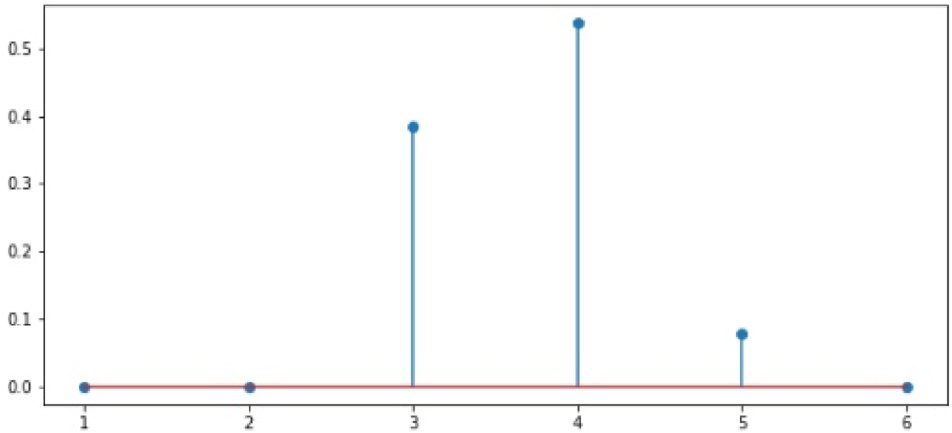
2) $\begin{pmatrix} 6 & 6 & 4 & 4 & 10 & 3 \\ 10 & 4 & 6 & 4 & 0 & 9 \\ 10 & 7 & 0 & 7 & 9 & 8 \end{pmatrix}$

value $= 4.4615$

p: [0.308, 0.538, 0.154]

q: [0, 0, 0.385, 0.538, 0.077, 0]



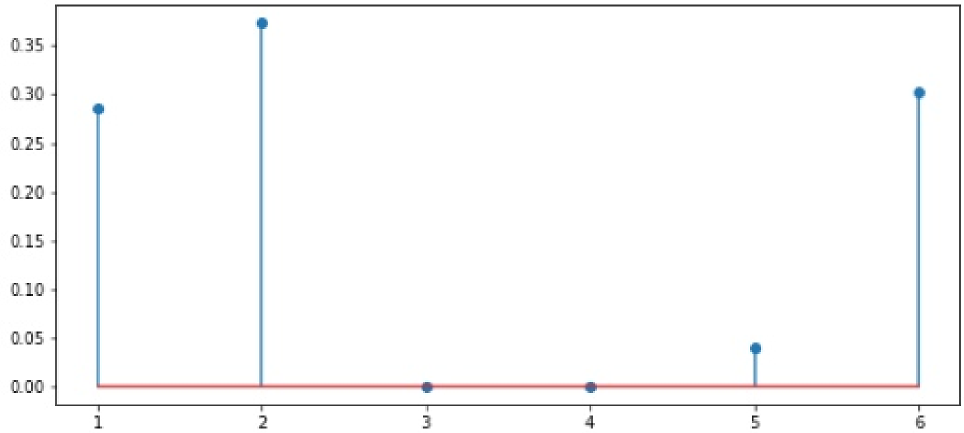


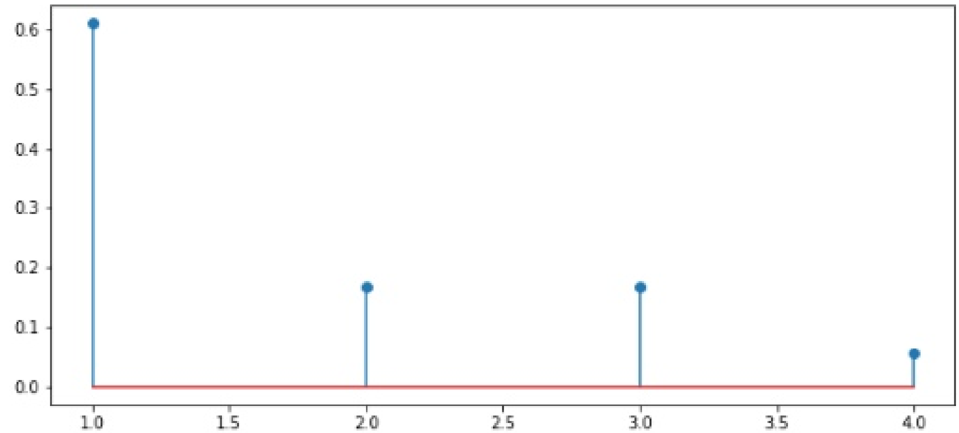
3) $\begin{pmatrix} 4 & 1 & 6 & 5 \\ 3 & 8 & 4 & 1 \\ 3 & 8 & 1 & 3 \\ 3 & 8 & 0 & 9 \\ 3 & 8 & 2 & 7 \\ 5 & 1 & 2 & 6 \end{pmatrix}$

value = 3.8(8)

p: [0.286, 0.373, 0, 0, 0.04, 0.302]

q: [0.611, 0.167, 0.167, 0.056]





Другие примеры рассмотрены в prak.ipynb и nose.py

Необходимое ПО

Пакет ПО Anaconda3
Библиотека nosetests
Библиотека numpy
Библиотека matplotlib

Инструкция по запуску

1. Открыть Jupyter Notebook
2. В веб-среде перейти к расположению файла prak.ipynb
3. Запустить prak.ipynb

Для проведения тестирования необходимо перейти в папку nosetest и выполнить команду в терминале

nosetests nose.py

Примечание: файл nose.py и prak.py должны находиться в одной репозитории

Вклад участников команды

- Изучение задачи с точки зрения теории игр: Рзянина
- Изучение алгоритма симплекс-метода: Долгая, Шумилин
- Реализация функции nash_equilibrium: Долгая, Шумилин
- Визуализация спектров: Рзянина
- Написание автоматических тестов: Шумилин
- Оформление в виде пакета: Шумилин
- Написание и оформление отчета: Шумилин
- Редакция отчета: Рзянина

Список литературы

[1] Васин А. А., Морозов В. В. 2005. Теория игр и модели математической экономики. (учебное пособие). - М.: МАКС Пресс, 272С.

[2] HEC Montreal. The steps of the simplex method algorithm // HEC Montreal, https://www.hec.ca/en/cams/help/topics/The_steps_of_the_simplex_algorithm.pdf

[3] ООО "Новый семестр". Пример решения матричной игры методом линейного программирования // ООО "Новый семестр https://math.semestr.ru/games/gamesimplex.php