Авторы работы: Мырзагали Айым, Кулешова Екатерина, Юневич Дарья (312 группа)

Задание 1

Описание задачи

Задача состоит в численном решении антагонистической матричной игры. В рамках задачи мы :

- написали код, решающий матричную игру путем сведения ее к паре двойственных задач линейного программирования;
- проиллюстрировали работу данного кода путем визуализации спектров оптимальных стратегий;
- познакомились с языком программирования Python, библиотекой SciPy.

• Определение. Система Γ = (X, Y, K), где X и Y непустые множества, и функция K : X×Y*→R, называется антагонистическоӥ игроӥ в нормальноӥ форме.

• Определение. Антагонистические игры, в которых оба игрока имеют конечные множества стратегиӥ, называются матричными.

Описание алгоритма

В ходе программы рассматривается произвольная матрица выигрыша. На первом этапе путем прибавления положительного числа приводим все элементы матрицы к положительными значениям. Далее для нахождения оптимальных стратегий и значения игры данная задача сводится к двойственной задаче линейного программирования: введенная матрица преобразуется в две системы линейных неравенств относительно переменных, являющихся координатами вектора оптимальных стратегий игроков, а также учитываются ограничения на не отрицательность переменных. На данном этапе преобразованная задача уже состоит в том, чтобы минимизировать сумму элементов вектора оптимальной стратегии первого игрока и максимизировать сумму элементов вектора оптимальной стратегии второго игрока. В решении данной задачи используется симплекс-метод, который вычисляет два требуемых вектора. Далее полученный результат домножается на коэффициент, пропорциональный сумме элементов полученных векторов и результатом являются смешенные стратегии двух игроков. Для вычисления значения игры высчитывается разность между единицей, деленной на сумму элементов вектора, который является решением для минимизирующей задачи, и положительным числом, на которое была смещена изначальная матрица.

Описание программы

На вход программы требуется ввести матрицу выигрыша путем обозначения количества строк в матрице, а также сами строки. Из введенных данных в программе строится двумерный массив mas, который в последствии передается в функцию **Nash_Equilibrium(a)** для вычисления оптимальных стратегий и значения игры и в функцию **Print_answer(a)** для визуализации результата.

-- Описание функции Nash_Equilibrium(a)

На вход функция получает матрицу и первым действием приводит все ее элементы к положительным числам путем добавления модуля минимального элемента и единицы:

```
[b = a + abs(int(a.min())) + 1]
```

Далее строится двойственная задача для первого игрока. Для построения системы неравенств вводится транспонированная матрица mas1:

```
[mas1 = b.transpose()]
```

Элементы которой используются в качестве коэффициентов в линейных неравенствах. Далее с помощью функции linprog из библиотеки scipy.optimize решаем поставленную задачу линейного программирования симплекс-методом и записываем результат в переменную opt:

```
[opt = linprog(c = np.ones(len(b)), A_ub = mas1 * -1, b_ub = np.ones(len(b[0])) * -1, method = "revised simplex")]
```

Полями данной функции являются:

- c – единичный вектор длины, равной количеству столбцов матрицы b, характеризующий коэффициенты суммы элементов, которую следует минимизировать;
- A_ub – массив коэффициентов системы, взятых с отрицательными знаками (для изменения знака неравенства на противоположный ≥ → ≤);
- b_ub - единичный вектор длины, равной количеству строк матрицы b, характеризующий правую сторону неравенств;
- method – метод оптимизации, который применяет вызванная функция.

Далее в переменную out1 записывается вектор-решение оптимизационной задачи, домноженный на коэффициент.

```
[out1 = opt.x * (1/np.sum(opt.x))]
```

Затем функция аналогично высчитывает оптимальную стратегию для второго игрока и в переменную out3 записывает значение игры.

```
[out3 = 1/np.sum(opt.x) - (abs(int(a.min())) + 1)]
```

Возвращает функция список, состоящий из трех элементов – ответов на поставленную задачу.

-- Описание функции Print_answer(a)

Данная функция получает список из трех элементов: двух векторов и числа – значения игры. Функция выводит на печать эти данные с помощью методов из библиотеки matplotlib и реализует графики.

Этап тестирование

Для тестирования были выбраны три разные игры:

- Игра с седловой точкой в чистых стратегиях.
- Игра, в которой спектр оптимальной стратегии неполон.
- Игра, в которой спектр оптимальной стратегии полон.

Инструкция к запуску

Необходимое ПО: Windows 7

Запустить программу можно в google colab, пройдя по ссылке:
https://colab.research.google.com/drive/1g-o2UkjOR6uptTFOAkpf5N0pIm5SO4Od?usp=sharing