Задача численного решения антагонистической матричной игры

Описание:

1) Написана функция nash_equilibrium(a), которая принимает матрицу выигрыша и возвращает значение игры и оптимальные стратегии первого и второго игроков.

2) Проиллюстрированна работа кода путем решения нескольких игр и визуализации спектров оптимальных стратегий игроков в Jupyter. В частности, приведены игры, в которых:

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

Для выполнения задачи необходимо установить библиотеки : numpy, scipy, matplotlib а так же jupiter notebook.

Алгоритм решения задачи построен на сведении матричной игры, решение которое представимо в виде v=$\max\limits_{p\epsilon P}\min\limits_{1\le j\le n}\sum_{i=1}^{m}p_i a_{ij}$, к задаче линейного программирования. С помощью замен переменных: v=$\max\limits_{(u,p)\in B} u$, где $B=\left\{(u,p)\middle|\sum_{i=1}^{m}p_i a_{ij}, j=1,\dots,n, \sum_{i=1}^{m}p_i=1,\ p_i\ge 0, i=1,\dots,m\right\}$

и $z=p_i/u$, получаем решение задачи в виде $v=1/\sum_{i=1}^{m}z_i^0$, где z^0-оптимальное решение задачи линейного программирования:
$\sum_{i=1}^{m}z_i \to min, \sum_{i=1}^{m}a_{ij}z_i\ge 1,\ j=1,\dots,n,\ \ z_i\ge 0,\ i=1,\dots,m.$

Для избавления проблем со знаками в начале матрица проверяется на неположительность и преобразуется к положительному виде в случае необходимости. Для решения задачи в программе используется функция linprog из scipy.optimize, которая принимает следующие параметры: коэффициенты функции(в виде массива), которую необходимо минимизировать для первого игрока, максимизировать для второго (для первого игрока-единицы, для второго игрока-единицы со знаком минус), матрицу коэффициентов для неравенств (вида $a*p_1+b*p_2\le c$, поэтому для первого игрока берется транспонированная матрица, умноженная на -1, для второго-исходная) и массив, представляющий собой правую часть неравенств (опять же, для первого игрока это единицы, умноженные на -1, для второго- единицы). Далее с помощью linprog().fun получаем значение исследуемой функции, а по формуле $p^0=vz^0$ находим массив равный оптимальным стратегиям игроков. Анализируем полученные ответы и выводим информацию о значении игры, полноте/неполноте спектра стратегии и существовании решения в чистых стратегиях.

Далее с помощью функций из matplotlib.pyplot создаются 2 фигуры, каждая из которых соответствует распределению значений оптимальных стратегий для каждого из игроков, далее для наглядности проводятся вертикальные линии в точках, соответствующих каждой стратегии, и добавляется сетка.

Задание выполняли: Ефарова Д, Михайлов Д., Стрелецкий Н. 312 группа

Написанием кода занимались совместно.