**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра математического обеспечения и применения ЭВМ**

отчет

**по практической работе №1**

**по дисциплине «Теория принятия решений»**

Тема: Принятие решений в матричных играх

Вариант 13

|  |  |  |
| --- | --- | --- |
| Студент гр. 8383 |  | Мололкин К.А. |
| Преподаватель |  | Попова Е.В. |

Санкт-Петербург

2022

**Цель работы**

Изучение различных инструментальных средств для решения задач поддержки принятия решения, а также овладение навыками принятия решения на основе матричных игр.

**Основные теоретические положения**

Рассмотрим простейшую математическую модель конечной конфликтной ситуации, в которой имеется два участника и выигрыш одного равен проигрышу другого. Такая модель называется антагонистической игрой двух лиц с нулевой суммой. Игра состоит из двух ходов: игрок А выбирает одну из возможных стратегий, а игрок Б выбирает одну из возможных стратегий . Каждый выбор производится при полном незнании выбора соперника. В результате выигрыш игроков составит соответственно  и . Цель игрока А – максимизировать величину , а игрока Б – минимизировать эту величину. Критерием принятия решения является функция, выражающая предпочтение лица, принимающего решение, и определяющая правило, по которому выбирается приемлемый или оптимальный вариант решения.

Матрица, составленная из величин .

|  |  |  |
| --- | --- | --- |
|  | . |  |

Матрица выше называется платежной матрицей, или матрицей игры. Каждый элемент платежной матрицы , равен выигрышу А (проигрышу Б), если он выбрал стратегию , а игрок Б выбирал стратегию .

Задача каждого из игроков – найти наилучшую стратегию игры, при этом предполагается, что противники одинаково разумны и каждый из них делает все, чтобы получить наибольший доход.

Найдем наилучшую стратегию первого игрока. Если игрок А выбрал стратегию , то в худшем случае (например, если его ход известен Б) он получит выигрыш . Предвидя такую возможность, игрок А должен выбрать такую стратегию, чтобы максимизировать свой минимальный выигрыш.

|  |  |  |
| --- | --- | --- |
|  |  |  |

Представленная выше величина  – гарантированный выигрыш игрока А называется нижней ценой игры. Стратегия , обеспечивающая получение выигрыша , называется максиминной. Если первый игрок будет придерживаться своей максиминной стратегии, то у него есть гарантия, что он в любом случае выиграет не меньше . Аналогично определяется наилучшая стратегия второго игрока. Игрок Б при выборе стратегии , в худшем случае получит проигрыш . Он выбирает стратегию  при которой его проигрыш будет минимальным и составит

|  |  |
| --- | --- |
|  |  |

Представленная выше величина  – гарантированный проигрыш игрока Б называется верхней ценой игры. Стратегия , обеспечивающая получение проигрыша , называется минимаксной. Если второй игрок будет придерживаться своей минимаксной стратегии, то у него есть гарантия, что он в любом случае проиграет не больше . Фактический выигрыш игрока А (проигрыш игрока Б) при разумных действиях партнеров ограничен верхней и нижней ценой игры. Для матричной игры справедливо неравенство . Если , т.е.

|  |  |  |
| --- | --- | --- |
|  |  |  |

то выигрыш игрока А (проигрыш игрока Б) определяется числом . Оно называется ценой игры.

В соответствии с этим, если , то такая игра называется игрой с седловой точкой. Элемент матрицы , соответствующий паре оптимальных стратегий , называется седловой точкой матрицы. Этот элемент является ценой игры.

Седловой точке соответствуют оптимальные стратегии игроков. Их совокупность – решение игры, которое обладает свойством: если один из игроков придерживается оптимальной стратегии, то второму отклонение от своей оптимальной стратегии не может быть выгодным. Если игра имеет седловую точку, то говорят, что она решается в чистых стратегиях.

Если платежная матрица не имеет седловой точки, т.е.  и то поиск решения игры приводит к применению сложной стратегии, состоящей в случайном применении двух и более стратегий с определенными частотами.

Сложная стратегия, состоящая в случайном применении всех стратегий с определенными частотами, называется смешанной.

**Постановка задачи**

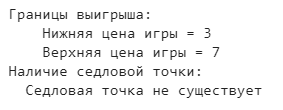
Используя инструментальные средства компьютерной алгебры решить матричные задачи.

**Выполнение работы**

1. С помощью инструментального средства определить границы выигрыша и наличие седловой точки для матрицы . Матрица представлена ниже:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Результат выполнения программы показан на рис. 1. Исходный код программы можно увидеть в приложении А.

****

*Рисунок 1 – Результат выполнения программы для матрицы*

1. Графически и аналитически решить матричную игру 2×2 для матрицы . Матрица представлена ниже:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Решим данную задачу аналитически. Надо проверить, есть ли седловая точка для исходной платежной матрицы. В случае если платежная матрица имеет седловую точку, необходимо выписать решение игры в чистых стратегиях.

Найдем нижнюю и верхнюю цену игры.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

Получается, что , а значит нет седловой точки и решение в чистых стратегиях не существует.

Найдем цену игры . Известно, что . В таком случае игрок A должен выбрать свои смешанные стратегии так, чтобы получить максимальный средний выигрыш. Аналогично, игрок Б должен выбрать свои смешанные стратегии так, чтобы минимизировать средний выигрыш игрока А. Для этого запишем две системы уравнений и решим их.

Для игрока А:

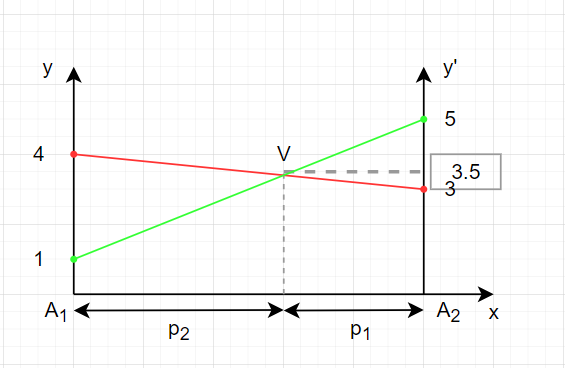
|  |  |  |
| --- | --- | --- |
|  |  |  |

Для игрока Б:

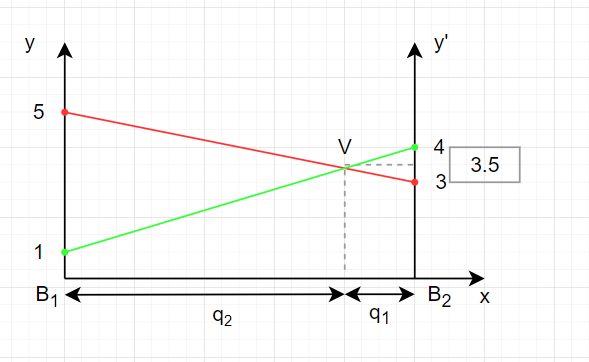
|  |  |  |
| --- | --- | --- |
|  |  |  |

Оптимальная стратегия игрока А: Оптимальная стратегия игрока Б: Цена игры:

Решим данную задачу графически. Результаты представлены на рис. 2 и 3.



*Рисунок 2 – Гарантированный выигрыш первого игрока в матрице платежей*



*Рисунок 3 – Гарантированный проигрыш второго игрока в матрице платежей*

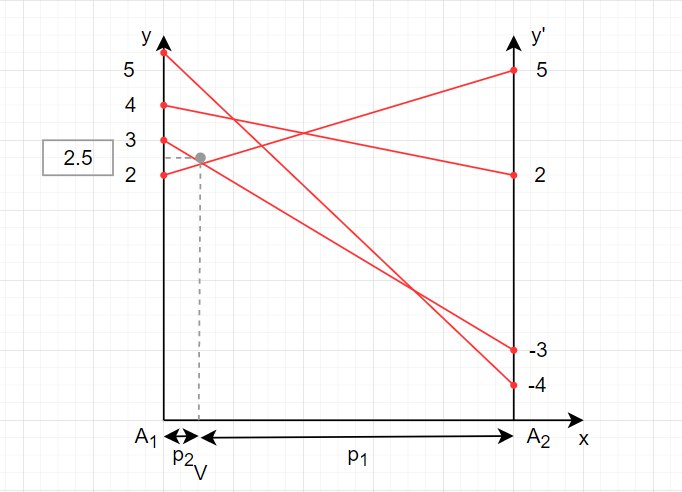
По рис. 2 и 3 можно определить, что стратегии игроков А и Б равны , цена игры – , Так как можем сделать вывод о том, что седловой точки не существует.

Относительная погрешность равна

1. Графически и аналитически решить матричную игру 2×N для матрицы . Матрица представлена ниже:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Решим данную задачу графически. Результат представлен на рис. 5.



*Рисунок 5 – Гарантированный выигрыш первого игрока при различном выборе смешанной стратегии*

На рис. 5 можно определить цену игры , Так как можем сделать вывод о том, что седловая точка отсутствует.

Решим данную задачу аналитически. Для аналитического решения выберем прямые, на пересечении которых находится седловая точка и построим из них матрицу 2 × 2:

Найдем нижнюю и верхнюю цену игры.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

Получаем, что , а значит нет седловой точки и решение в чистых стратегиях не существует.

Найдем цену игры . Известно, что . Запишем две системы и уравнений и решим их.

Для игрока А:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Для игрока Б:

|  |  |  |
| --- | --- | --- |
|  |  |  |

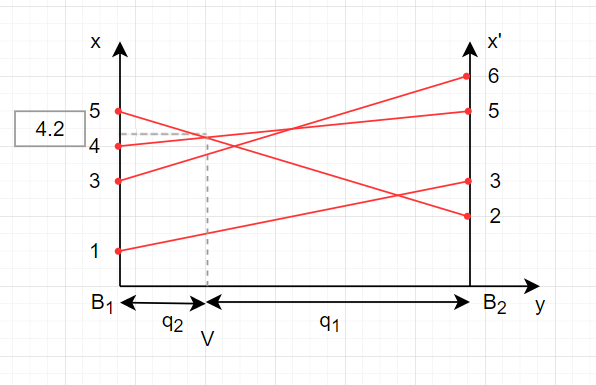
Оптимальная стратегия игрока А: Оптимальная стратегия игрока Б: Цена игры:

Относительная погрешность равна:

1. Графически и аналитически решить матричную игру *M×*2 для матрицы . Матрица представлена ниже:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Решим данную задачу графически. Результат представлен на рис. 6.



*Рисунок 6 – Гарантированный проигрыш второго игрока при различном выборе смешанной стратегии*

Исходя из рис. 6 можно определить, что цена игры – Так как можем сделать вывод о том, что седловой точки не существует.

Решим данную задачу аналитически.

Для аналитического решения выберем прямые, на пересечении которых находится седловая точка и построим из них матрицу 2 × 2:

Найдем нижнюю и верхнюю цену игры.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

Получаем, что , а значит седловой точки нет и решение в чистых стратегиях не существует.

Найдем цену игры . Известно, что .

Запишем две системы уравнений.

Для игрока А:

|  |  |
| --- | --- |
|  |  |
|  |  |

Для игрока Б:

|  |  |
| --- | --- |
|  |  |
|  |

Оптимальная стратегия игрока А: . Оптимальная стратегия игрока Б: . Цена игры: .

Относительная погрешность равна

1. С помощью симплекс-метода решить матричную игру *M×N* для матрицы . Матрица представлена ниже:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Найдем нижнюю и верхнюю цену игры.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

Получаем, что , а значит нет седловой точки и решение в чистых стратегиях не существует.

Найдем цену игры . Известно, что . Запишем две системы уравнений.

Для игрока А:

|  |  |  |
| --- | --- | --- |
|  | |  |
|  |  | |

Чтобы найти решение данной задачи относительно первого игрока А надо найти минимум функции при следующих ограничениях:

|  |  |
| --- | --- |
|  |  |

С помощью библиотеки SciPy симплекс-методом вычислен вектор *X*. Результаты представлены на рис. 7.



*Рисунок 7 – Решение вектора симплекс-методом матрицы*

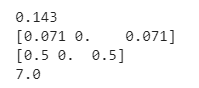
Получаем при

Цена игры при этом что соотносится с первоначальной оценкой .

|  |  |
| --- | --- |
| Для игрока Б: |  |

Чтобы найти решение данной задачи относительно второго игрока Б надо найти максимум функции при следующих ограничениях:

Симплекс-методом вычислен вектор *Y*. Результаты вычислений представлены на рис. 8.



*Рисунок 8 – Решение вектора симплекс-методом матрицы*

Получаем при

Цена игры что соотносится с первоначальной оценкой .

Опираясь на полученные данные, можно найти цену игры и оптимальные смешанные стратегии игроков А и Б:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

**Выводы**

В ходе выполнения практической работы было реализовано инструментальное средство для нахождения нижней и верхней цен игры и для нахождения координат седловой точки на языке программирования Python, а также были получены навыки работы с библиотекой SciPy для линейного программирования.

При решении матричных игр были применены графический и аналитический методы нахождения оптимальных смешанных стратегий, а также был применён симплекс-метод. Решения матричных игр обоими способами дали одинаковый результат.

Приложение А

**ПРОГРАММА ПОИСКА СТАЦИОНАРНОЙ ТОЧКИ**

# -\*- coding: utf-8 -\*-

"""Untitled0.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/1xinIRuRO3vw0D-bgm2ldxfZLRxLaBFCM

"""

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from IPython.core.interactiveshell import InteractiveShell

InteractiveShell.ast\_node\_interactivity = "all"

from scipy.optimize import linprog

def default(c):

amin = c.min(axis=1)

bmax = c.max(axis=0)

alpha = max(amin)

beta = min(bmax)

print('Границы выигрыша:')

print(f' Нижняя цена игры = {alpha}')

print(f' Верхняя цена игры = {beta}')

print('Наличие седловой точки:')

print(f' Седловая точка существует') if alpha == beta else print(f' Седловая точка не существует')

if alpha == beta:

print(f'Координаты седловой точки равны ({np.argmax(amin)+1}, {np.argmin(bmax)+1})')

p1 = (c[1,1]-c[1,0])/(c[0,0]+c[1,1]-(c[0,1]+c[1,0]))

p2 = 1-p1

q1 = (c[1,1]-c[0,1])/(c[0,0]+c[1,1]-(c[0,1]+c[1,0]))

q2 = 1-q1

v = c[0,0]\*p1+c[1,0]\*p2

print(np.array([p1,p2,q1,q2,v]).round(3))

c1 = np.array([[2,9,10,5], [3,4,8,7], [-4,3,-4,-2], [8,5,-3,-4]])

c2 = np.array([[1,4],[5,3]])

c3 = np.array([[2,4,3,5], [5,2,-3,-4]])

c31 = np.array([[2,3],[5,-3]])

c4 = np.array([[3,6],[5,2],[1,3],[4,5]])

c41 = np.array([[5,2],[4,5]])

c5 = np.array([[3,14,7], [8,9,6], [5,8,9]])

# c1,c2,c3,c4,c5

default(c41)

IST = 21/9

SCHIT = 2.5

(np.abs(IST-SCHIT)/IST)\*100

''

(np.abs((1-IST)-(1-SCHIT))/(1-IST))\*100

func\_coefs = [1,1,1]

lhs\_coefs = [[-3,-8,-5],[-14,-9,-8],[-7,-6,-9]]

rhs\_coefs = [-1,-1,-1]

edges = [(0, float("inf")),(0, float("inf")),(0, float("inf"))]

opt = linprog(c=func\_coefs, A\_ub=lhs\_coefs,

b\_ub=rhs\_coefs, bounds=edges,

method="simplex")

opt.fun.round(3)

opt.x.round(3)

(1/opt.fun) \* opt.x

(1/opt.fun).round(3)

obj = [-1,-1,-1]

lhs\_ineq = [[3,14,7],[8,9,6],[5,8,9]]

rhs\_ineq = [1,1,1]

bnd = [(0, float("inf")),(0, float("inf")),(0, float("inf"))]

opt = linprog(c=obj, A\_ub=lhs\_ineq, b\_ub=rhs\_ineq, bounds=bnd,method="simplex")

print(-opt.fun.round(3))

print(opt.x.round(3))

print((-1/opt.fun) \* opt.x)

print((-1/opt.fun).round(3))