1 Лабораторная работа №2

Цель работы - изучить аналитический (обратной матрицы) и численный (Брауна-Робинсона) методы нахождения смешанных стратегий в антагонистической игре двух лиц в нормальной форме.

1.1 Постановка задачи

Найдите цену игры и оптимальные стратегии обоих игроков методами обратной матрицы и Брауна-Робинсон. Сравните полученные результаты.

1.2 Ход работы

Вариант 5.

[16]:
```
pretty_print_default(False)
latex.add_to_preamble("\\usepackage[russian]{babel}")
from IPython.display import display, Math, Latex
```

Зададим матрицу игры:

[2]:
```
C = matrix(SR, 3, 3, [8, 12, 10, 1, 6, 19, 17, 11, 11])
C
```

[2]:
```
[ 8 12 10]
[ 1  6 19]
[17 11 11]
```

Стратегии игрока A:

[3]:
```
x = C.rows()
x
```

[3]:
```
[(8, 12, 10), (1, 6, 19), (17, 11, 11)]
```

Стратегии игрока B:

[4]:
```
y = C.columns()
y
```

[4]:
```
[(8, 1, 17), (12, 6, 11), (10, 19, 11)]
```

1.2.1 Аналитический метод

[5]:
```
u = matrix(SR, 1, C.nrows(), [1] * C.nrows())
v = (1/(u * C^(-1) * u.transpose()))
v
```

[5]:
```
[845/76]
```

Оптимальные стратегии игрока A:

[6]:
```
x_star = v * (u * C^(-1))
x_star
```

[6]:
```
[39/76  3/38 31/76]
```

Оптимальные стратегии игрока B:

[7]:
```
y_star = v * (C^(-1) * u.transpose()).transpose()
y_star
```

[7]:
```
[ 3/152  11/19 61/152]
```

1.2.2 Метод Брауна-Робинсон

[18]:
```
k = 12 # Число итераций

xk, yk = 0, 0

winA = [0] * C.nrows()
lossB = [0] * C.ncols()

vxk, vyk = dict(), dict()
for i in range(C.nrows()):
    vxk[i], vyk[i] = 0, 0

rows = [['$k$', 'Выбор игрока $A$', 'Выбор игрока $B$', 'Выигрыш $A$', 'Проигрыш $B$', '$\overline{v}^{k}$', '$\underline{v}^{k}$', '$\\varepsilon$']]
for ki in range(1, k + 1):
    winA = [xi + yi for xi, yi in zip(winA, y[yk])]
    lossB = [xi + yi for xi, yi in zip(lossB, x[xk])]

    maxX = max(winA)
    minY = min(lossB)

    vxk[xk] += 1
    vyk[yk] += 1

    rows.append([ki, xk + 1, yk + 1, winA, lossB, maxX / ki, minY / ki, maxX / ki - minY / ki])

    countX = winA.count(maxX)
    countY = lossB.count(minY)

    xk = winA.index(maxX)
    yk = lossB.index(minY)
```

```
    if countX > 1 and countY > 1:
        raise Exception("Should not be at all")
    elif countX == 1 and countY > 1:
        if lossB[xk] == minY:
            yk = xk
    elif countY == 1 and countX > 1:
        if winA[yk] == maxX:
            xk = yk

display(latex(table(rows)))
```

[18]:

k	Выбор игрока A	Выбор игрока B	Выигрыш A	Проигрыш B	$\overline{v}^k$	$\underline{v}^k$	ε
1	1	1	$[8, 1, 17]$	$[8, 12, 10]$	17	8	9
2	3	1	$[16, 2, 34]$	$[25, 23, 21]$	17	$\frac{21}{2}$	$\frac{13}{2}$
3	3	3	$[26, 21, 45]$	$[42, 34, 32]$	15	$\frac{32}{3}$	$\frac{13}{3}$
4	3	3	$[36, 40, 56]$	$[59, 45, 43]$	14	$\frac{43}{4}$	$\frac{13}{4}$
5	3	3	$[46, 59, 67]$	$[76, 56, 54]$	$\frac{67}{5}$	$\frac{54}{5}$	$\frac{13}{5}$
6	3	3	$[56, 78, 78]$	$[93, 67, 65]$	13	$\frac{65}{6}$	$\frac{13}{6}$
7	3	3	$[66, 97, 89]$	$[110, 78, 76]$	$\frac{97}{7}$	$\frac{76}{7}$	3
8	2	3	$[76, 116, 100]$	$[111, 84, 95]$	$\frac{29}{2}$	$\frac{21}{2}$	4
9	2	2	$[88, 122, 111]$	$[112, 90, 114]$	$\frac{122}{9}$	10	$\frac{32}{9}$
10	2	2	$[100, 128, 122]$	$[113, 96, 133]$	$\frac{64}{5}$	$\frac{48}{5}$	$\frac{16}{5}$
11	2	2	$[112, 134, 133]$	$[114, 102, 152]$	$\frac{134}{11}$	$\frac{102}{11}$	$\frac{32}{11}$
12	2	2	$[124, 140, 144]$	$[115, 108, 171]$	12	9	3

Тогда оптимальные стратегии для игрока A:

[9]:
```
x_strategies = matrix(1, 3, [i / k for i in vxk.values()])
x_strategies
```

[9]: `[1/12 5/12  1/2]`

Оптимальные стратегии для игрока B:

[10]:
```
y_strategies = matrix([i / k for i in vyk.values()])
y_strategies
```

[10]: `[1/6 1/3 1/2]`