МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ

по практической работе №3
по дисциплине «Теория принятия решений»
Тема: Игры с природой. Использование вероятностных характеристик в задачах принятия решений.

Студентка гр. 8383	____________________	Гречко В.Д.
Преподаватель	____________________	Попова Е.В.

Санкт-Петербург
2021

Цель работы.

Познакомиться с оптимизационными критериями, используемыми при игре с одним игроком. Применить вероятностные характеристики для решения задач принятия решений с наличием неопределенности.

Основные теоретические положения.

Существует неопределенность, не связанная с осознанным противодействием противника, а возникающая в связи с недостаточной информированностью ЛПР об объективных условиях, в которых будет приниматься решение. В математической модели присутствует «природа», и заранее неизвестно её состояние во время принятия решения, игра при этом называется игрой с природой. Осознанно действует только один игрок. Природа не противник, принимает какое-то состояние, не преследует цели, безразлична к результату игры.

В платежной матрице по строкам реализуются стратегии игрока, а по столбцам – множество состояний противоположной стороны. Элементы столбцов не являются проигрышами природы при соответствующих её состояниях. Задача выбора игроком чистой или смешанной стратегии проще так как отсутствует противодействие, но сложнее так как существует наличие неопределенности, связанное с дефицитом осведомленности игрока о характере проявления состояний природы.

Вариант.

Вариант 5. Критерий Гурвица.

Выполнение работы.

1. <u>Написать программу, формирующую входную матрицу, размера 10 на 10 с рандомными значениями из заданного диапазона (от 1/(N+1) до N+1, где N – номер варианта). Для критерия Байеса, формируется ещё одна строка с 10</u>

значениями вероятностей, рандомных, в сумме дающих 1. Для критерия Гурвица склонность к риску принять за 1/(N+2).

С помощью языка Python была написана программа, формирующая матрицу 10×10 со значениями из диапазона $\left[\frac{1}{6}; 6\right)$. Так как заданный оптимизационный критерий – это критерий Гурвица, то склонность к риску равна $\frac{1}{(N+2)} = \frac{1}{7}$. Полученная матрица представлена на рис. 1.

```
Платежная матрица:
 [[4.8693 3.4944 4.4715 5.3897 4.4463 3.0079 1.6795 0.1936 4.5362 5.5493]
 [0.262  2.9378 1.0348 1.151  3.3011 5.3366 3.6852 2.6601 1.6794 5.5564]
 [3.4035 4.091  4.0714 4.6571 1.7127 1.3913 5.0726 2.1021 2.176  2.6261]
 [4.6401 2.212  1.5353 0.8    4.5182 5.0279 4.2324 2.5532 5.0132 4.6054]
 [4.9245 1.232  0.5807 1.2564 5.3704 4.0114 0.2452 0.8587 4.03   3.4945]
 [5.9047 1.537  4.4614 3.3906 4.2327 1.3729 0.8758 4.7754 3.0836 4.849 ]
 [4.0878 5.735  5.245  3.2794 2.8958 1.9688 1.0129 3.0666 5.4468 0.4552]
 [0.8579 3.4711 4.4631 4.0352 5.4244 4.123  5.7502 3.2598 4.469  2.2547]
 [5.7386 1.7453 1.2313 5.1259 4.5558 1.1201 2.3628 5.153  5.7238 4.2798]
 [5.4806 5.2715 4.0268 3.4419 2.9942 4.6148 4.5287 0.2068 5.6784 1.6691]]
```

Рисунок 1 – Платежная матрица

2. Применить заданный оптимизационный критерий к платежной матрице и выбрать оптимальную стратегию (с помощью инструментальных средств).

Вычислим коэффициент Гурвица для платежной матрицы по формуле:

$$\max_i a_{ji}\gamma + (1-\gamma)\min_i a_{ji}$$

Полученные коэффициенты представлены на рис. 2.

```
Коэффициент Гурвица:
 [[0.9587]
 [1.0183]
 [1.9172]
 [1.404 ]
 [0.9774]
 [1.5942]
 [1.2094]
 [1.5568]
 [1.7799]
 [0.9884]]
```

Рисунок 2 - Коэффициенты Гурвица для платежной матрицы

Для того, чтобы принять решение, какую из стратегий использовать рассчитаем Z:

$$Z = \max_j(\max_i a_{ji}\gamma + (1-\gamma)\min_i a_{ji})$$

```
Z = 1.9171674090873942
Оптимальная стратегия находится в 3 строке
```

Z соответствует стратегии a_3.

3. Получить вероятностные характеристики предложенных задач в папке Варианты, для принятия решений в задачах со случайными характеристиками (без инструментальных средств).

Задача 1

Имеются 3 одинаковые на вид урны. В первой урне 3 белых и 8 черных шаров, во второй – 5 белых и 6 черных шаров, в третьей – 7 белых и 4 черных шаров. Некто выбирает наугад урну и вынимает один шар. Найти вероятность того, что шар извлечен из третьей урны, если известно, что этот шар белый (и принять решение об извлечении из других урн).

Решение

Вероятность, что шар будет извлечён из B_i-той урны:

$$P(B_1) = P(B_2) = P(B_3) = \frac{1}{3}$$

Вероятность вытащить белый шар (A) из B_i-той урны:

$$P_{B_1}(A) = \frac{3}{11};$$

$$P_{B_2}(A) = \frac{5}{11}$$

$$P_{B_3}(A) = \frac{7}{11}$$

Полная вероятность наступления события A:

$$P(A) = P(B_1)P_{B1}(A) + P(B_2)P_{B2}(A) + \cdots + P(B_n)P_{Bn}(A)$$

$$P(A) = \frac{1}{3} \cdot \frac{3}{11} + \frac{1}{3} \cdot \frac{5}{11} + \frac{1}{3} \cdot \frac{7}{11} = \frac{5}{11}$$

Вероятность того, что белый шар извлечен из третьей урны, можно найти по формуле Байеса:

$$P_A(B_i) = \frac{P(B_i)P_{Bi}(A)}{P(A)}$$

$$P_A(B_3) = \frac{\frac{1}{3} \cdot \frac{7}{11}}{\frac{5}{11}} = \mathbf{\frac{7}{15} \approx 0.47}$$

Найдем также вероятность извлечения из других урн:

$$P_A(B_1) = \frac{\frac{1}{3} \cdot \frac{3}{11}}{\frac{5}{11}} = \frac{1}{5} \approx 0.2$$

$$P_A(B_2) = \frac{\frac{1}{3} \cdot \frac{5}{11}}{\frac{5}{11}} = \frac{1}{3} \approx 0.33$$

Задача 2

На станциях отправления поездов находится 1000 автоматов для продажи билетов. Вероятность выхода из строя одного автомата в течение часа равна 0,002. Какова вероятность того, что в течение часа выйдут из строя менее двух автоматов (принять на основании этого решение о покупке автоматов).

Решение

Событие X – в течение часа выйдут из строя менее двух автоматов можно представить, как сумму несовместных событий: X_1 – в течение часа выйдет из строя 1 автомат и X_0 – в течение часа ни один автомат не выйдет из строя.

$$n = 1000$$

$$p = 0.002$$

$$P_n(k) = \frac{\lambda^k e^{-\lambda}}{k!}, \lambda = np$$

$$\lambda = np = 1000 \cdot 0.002 = 2$$

Найдем вероятность того, что из строя выйдет 0 автоматов:

$$k = 0$$

$$P_{1000}(0) = \frac{1 * e^{-2}}{0!} \approx 0.1353$$

Найдем вероятность того, что из строя выйдет 1 автомат:

$$k = 1$$

$$P_{1000}(1) = \frac{2^1 * e^{-2}}{1!} \approx 0.2707$$

Тогда:

$$P(X) = P(X_0) + P(X_1) = 0.1353 + 0.2707 = 0.406$$

Задача 3

Два стрелка стреляют по мишени. Вероятность попадания в цель для первого стрелка равна 0,6, для второго – 0,7. Составить закон распределения случайной величины X - числа попаданий. Найти математическое ожидание, дисперсию и среднее квадратическое отклонение этой случайной величины.

Решение

Пусть $q_1 = 1 - 0.6 = 0.4$ – вероятность промаха первого стрелка.

Пусть $q_2 = 1 - 0.7 = 0.3$ – вероятность промаха второго стрелка.

Вероятность того, что оба стрелка промахнулись:

$$P(0) = q_1 \cdot q_2 = 0.4 \cdot 0.3 = 0.12$$

Вероятность того, что один из стрелков попал в мишень:

$$P(1) = p_1 \cdot q_2 + q_1 \cdot p_2 = 0.6 \cdot 0.3 + 0.4 \cdot 0.7 = 0.46$$

Вероятность того, что оба стрелка попали в мишень:

$$P(2) = p_1 \cdot p_2 = 0.6 \cdot 0.7 = 0.42$$

Закон распределения случайной величины X – числа попаданий:

X	0	1	2
p	0.12	0.46	0.42

Математическое ожидание X:

$$M(X) = 0 \cdot 0.12 + 1 \cdot 0.46 + 2 \cdot 0.42 = 1.3$$

Дисперсия:

$$D(x) = M(X^2) - [M(X)]^2$$

$$D(x) = (1 \cdot 0.46 + 4 \cdot 0.42) - 1.3^2 = 2.14 - 1.69 = 0.45$$

Среднее квадратическое отклонение случайной величины:

$$\sigma(X) = \sqrt{D(X)}$$

$$\sigma(X) = \sqrt{0.45} \approx 0.671$$

Выводы.

В ходе лабораторной работы были изучены игры с «природой» и оптимизационные критерии для выбора стратегии при игре с одним игроком. Для решения поставленной задачи было написано инструментальное средство для вычисления критерия Гурвица и принятия решения относительно полученных результатов.

Кроме того, были получены навыки решения задач о принятии решений в задачах со случайными характеристиками.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

```
import numpy as np
a = 1/6
b = 6
inputMatrix = (b - a) * np.random.random_sample((10, 10)) + a

np.set_printoptions(precision = 4)
print("Платежная матрица:\n", inputMatrix)
gamma = 1/7
hurwitz = inputMatrix.max(axis=1)*gamma + (1 - gamma)*inputMatrix.min(axis=1)
hurwitz = hurwitz.reshape(-1,1)
print("Коэффициент Гурвица:\n", hurwitz)
print("Z =", hurwitz.max())
print("Оптимальная стратегия находится в", np.argmax(hurwitz)+1, "строке")
```