**Министерство образования Российской Федерации**

**МОСКВОСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. Н.Э.БАУМАНА**

Факультет: Информатика и системы управления (ИУ)  
Кафедра: Информационная безопасность (ИУ8)

**МЕТОДЫ ОПТИМИЗАЦИИ**

**Лабораторная работа №5 на тему:**   
««Игры с природой». Критерии принятия решений.».

Вариант – 1

**Преподаватель:**   
Коннова Н.С.

**Студент:**   
Александров А. Н.

**Группа:**   
ИУ8-34

Москва, 2020

**Цель работы:**

Ознакомиться с основными понятиями «игр с природой»; получить навыки применения различных критериев (Бернулли, Вальда, максимума, Сэвиджа) для выбора стратегии в условиях полной неопределённости.

**Постановка задачи:**

Найти стратегии игрока при реализации гипотез недостаточного основания (Бернулли), пессимизма (Вальда), оптимизма, смешанной (Гурвица) при α = 0,5, рисков (Сэвиджа). Рекомендовать выбор стратегии согласно принципу большинства. (строки матриц соответствуют стратегиям игрока, столбцы – состояниям природы):

**Ход решения:**

Для решения задачи, воспользовался собственной программой, написанной на языке программирования **Python**(см. Приложение А):

Файл ***best\_strategy.py***, содержащий функции для нахождения оптимальных стратегий согласно критериям:

* функция ***BernulliCriteria(matrix: np.array)****:* критерий недостаточного основания (Бернулли);
* функция ***ValdCriteria(matrix: np.array)****:* критерий пессимизма (Вальда);
* функция ***OptimismCriteria(matrix: np.array)****:* критерий авантюры (максимума, оптимизма);
* функция ***GurvitzCriteria(matrix: np.array)****:* критерий Гурвица (смешанный);
* функция ***SevigeCriteria(matrix: np.array)****:* критерий рисков (Севиджа).

Каждая из функций принимает на вход матрицу стратегий и возвращает оптимальную стратегию, а функция ***ChooseBestStrategy(matrix: np.array)*** отбирает по принципу большинства стратегию, оказавшуюся наилучшей в большем числе критериев.

Итак, матрица стратегий, описывающая «игру с природой», имеет вид:

Таблица 1 Матрица стратегий.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Стратегии |  |  |  |  |
|  | 1 | 11 | 12 | 11 |
|  | 7 | 5 | 7 | 7 |
|  | 16 | 6 | 13 | 2 |
|  | 9 | 9 | 16 | 13 |
|  | 17 | 18 | 15 | 7 |

***Критерий Бернулли (принцип недостаточного основания).***

Все состояния природы предполагаем равновероятными. Тогда стоит руководствоваться стратегией с максимальной величиной математического ожидания. Для каждой стратегии она определяется как

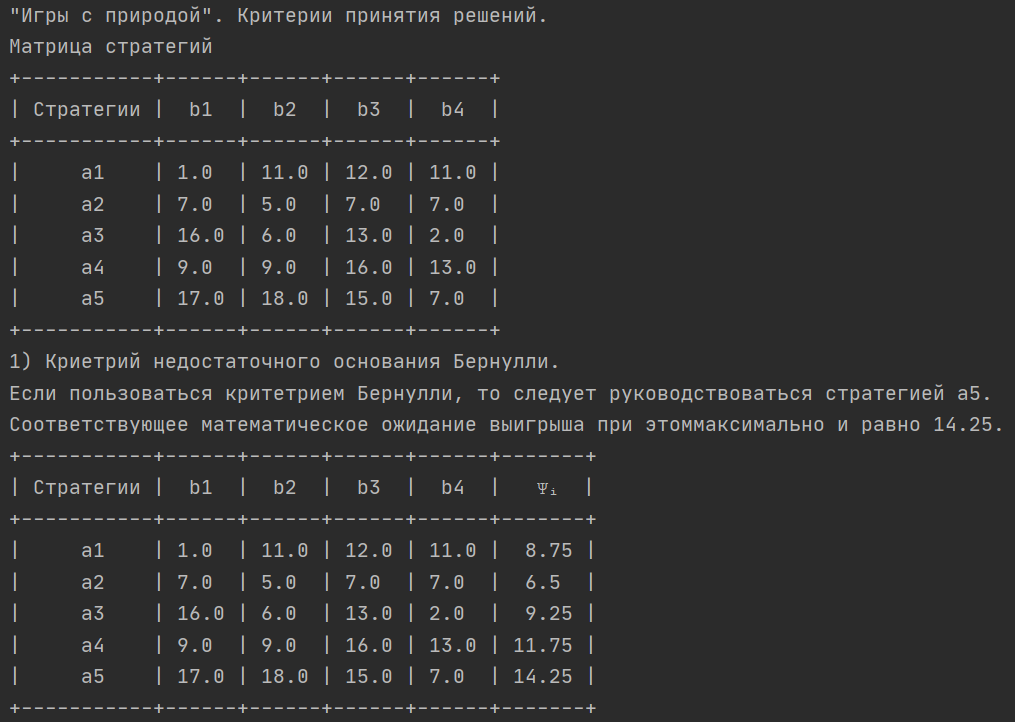


Рисунок 1 Критерий Бернулли.

Оптимальной является стратегия с соответствующим максимальным математическим ожиданием

***Критерий Вальда (пессимистический).***

В соответствии с этим критерием стоит выбрать стратегию с наименьшей вероятностью проигрыша и доставит максимальную прибыль. То есть критерий есть ни что иное, как нижняя цена игры:

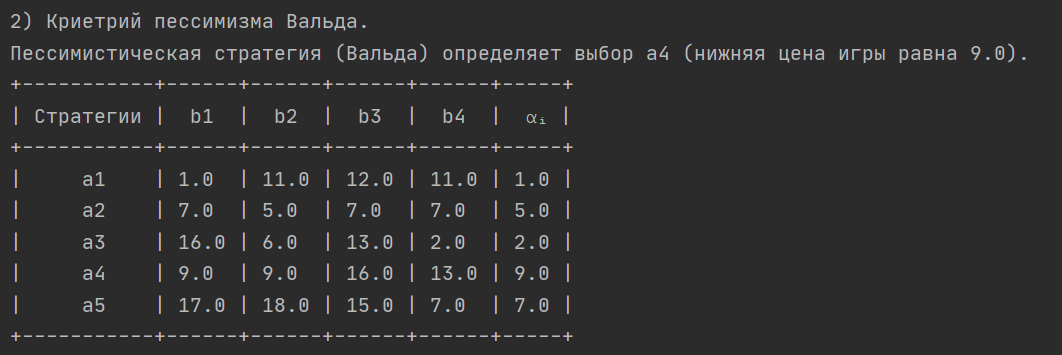


Рисунок 2 Критерий Вальда.

Оптимальной является стратегия Нижняя цена игры равна 9.

***Критерий максимума (оптимистический).***

Выбираем наиболее авантюристическую стратегию с максимальным выигрышем:

.

Максимальный выигрыш 18 достигается при стратегии .

***Критерий Гурвица.***

Определим вероятность своего «везения» как . Тогда критерий определяется из условия равновероятной реализации пессимистической и оптимистической гипотез:

.

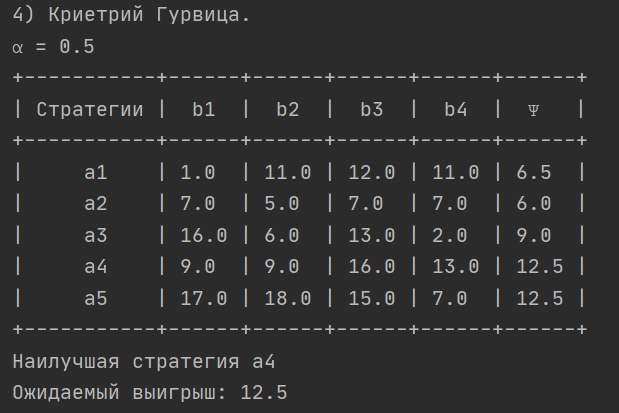


Рисунок 3 Критерий Гурвица

Наилучшая стратегия , ожидаемый выигрыш равен 12,5.

***Критерий Сэвиджа (критерий рисков).***

Заполним таблицу рисков по формуле (столбец ):

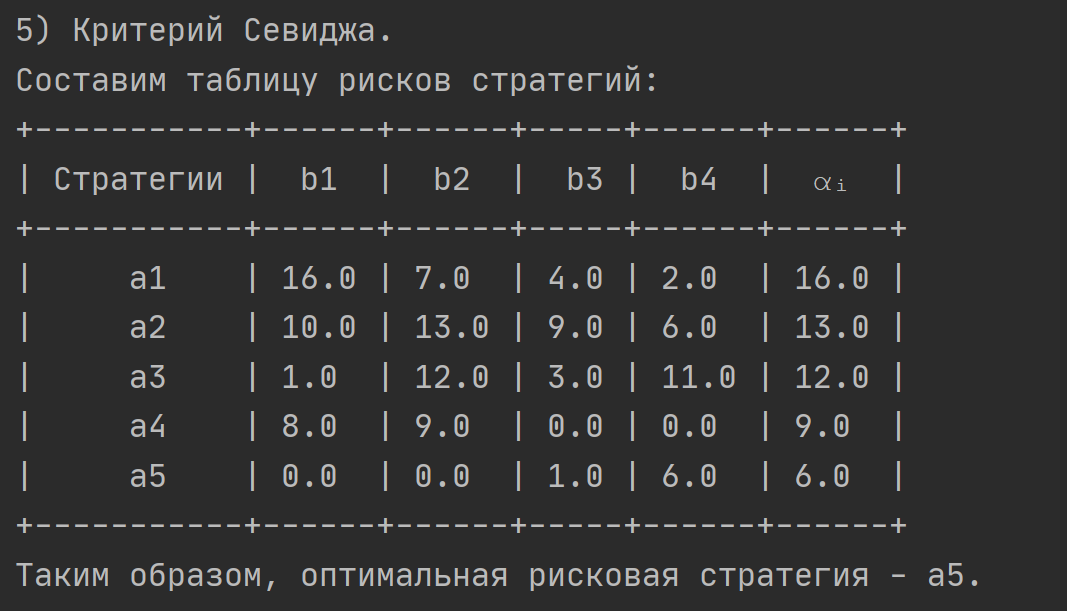


Рисунок Критерий Сэвиджа.

Тогда оптимальная стратегия соответствует минимальному значению в столбце . Оптимальная рисковая стратегия .

В заключение, по принципу большинства, следует рекомендовать стратегию .

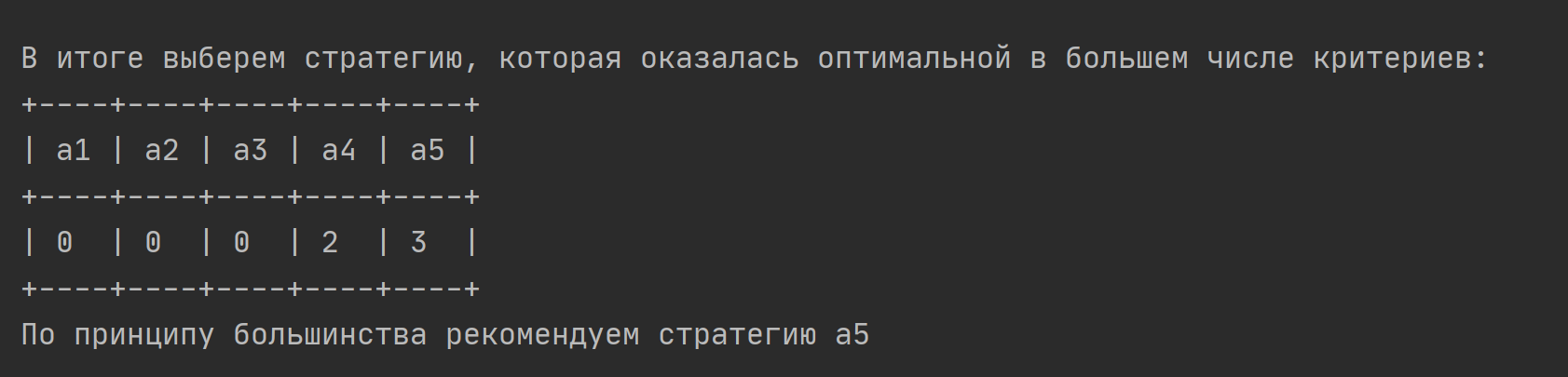


Рисунок 5 Выбор рекомендуемой стратегии.

**Вывод:**

В данной работе я познакомился с основными понятиями «игр с природой»; получил навыки применения критериев стратегии в условиях неопределённости.

По результатам оптимальными стратегиями стали:

* по критерию Бернулли: ;
* по критерию Вальда: ;
* по критерию максимума: ;
* по критерию Гурвица: ;
* по критерию Сэвиджа: .

По принципу большинства была рекомендована стратегия .

**Приложение А**

*Код программы*

***Файл «main.py»***

*# Copyright 2020 Alexey Alexandrov <sks2311211@yandex.ru>*

*"""*

*Лабораторная работа № 6*

*"Игры с природой". Критерии принятия решений*

*Цель работы: ознакомление с основными понятиями "игр с природой"; получение навыков применения различных критериев*

*(Бернулли, Вальда, максимума, Гурвица, Сэвиджа) для выбора стратегии в условиях полной неопроеделённости.*

*Вариант 1.*

*"""*

from best\_strategy import \*

STRATEGY\_MATRIX = np.array([

[1, 11, 12, 11],

[7, 5, 7, 7],

[16, 6, 13, 2],

[9, 9, 16, 13],

[17, 18, 15, 7]

], dtype=float)

if \_\_name\_\_ == **'\_\_main\_\_'**:

print(**f'"Игры с природой". Критерии принятия решений.**\n**Матрица стратегий**\n{OutMatrix(STRATEGY\_MATRIX)}**'**)

ChooseBestStrategy(STRATEGY\_MATRIX)

***Файл «best\_strategy.py»***

***# Copyright 2020 Alexey Alexandrov <sks2311211@yandex.ru>***

import numpy as np

from prettytable import PrettyTable

*# Вероятность "везения" для критерия Гурвица.*

ALPHA = 0.5

def OutMatrix(matrix: np.array):

table = PrettyTable()

table.field\_names = [**"Стратегии"**] + [**f"b**{j}**"** for j in range(1, matrix.shape[1] + 1)]

for i in range(matrix.shape[0]):

table.add\_row([**f"a**{i + 1}**"**] + list(matrix[i]))

return table

def BernulliCriteria(matrix: np.array):

*"""*

*Криетрий недостаточного основания Бернулли.*

*:param matrix: матрица стратегий.*

*:return полученная оптимальная стратегия.*

*"""*

table = OutMatrix(matrix)

col\_sum = np.sum(matrix, axis=1)

table.add\_column(**"Ψᵢ"**, list(col\_sum / matrix.shape[1]))

optimal\_strategy = **f"a**{col\_sum.argmax(axis=0) + 1}**"**

print(**f"1) Криетрий недостаточного основания Бернулли.**\n**"**

**f"Если пользоваться критетрием Бернулли, то следует руководствоваться стратегией"**

**f"**{optimal\_strategy}**. Соответствующее математическое ожидание выигрыша при этом"**

**f"максимально и равно** {col\_sum.max(axis=0) / matrix.shape[1]}**.**\n{table}**"**)

return optimal\_strategy

def ValdCriteria(matrix: np.array):

*"""*

*Криетрий пессимизма Вальда.*

*:param matrix: матрица стратегий.*

*:return полученная оптимальная стратегия.*

*"""*

table = OutMatrix(matrix)

col\_min = np.min(matrix, axis=1)

table.add\_column(**"αᵢ"**, list(col\_min))

optimal\_strategy = **f"a**{col\_min.argmax(axis=0) + 1}**"**

print(**f"2) Криетрий пессимизма Вальда.**\n**"**

**f"Пессимистическая стратегия (Вальда) определяет выбор** {optimal\_strategy} **"**

**f"(нижняя цена игры равна** {col\_min.max(axis=0)}**).**\n{table}**"**)

return optimal\_strategy

def OptimismCriteria(matrix: np.array):

*"""*

*Криетрий авантюры (максимума, оптимизма).*

*:param matrix: матрица стратегий.*

*:return полученная оптимальная стратегия.*

*"""*

optimal\_strategy = **f"a**{np.max(matrix, axis=1).argmax() + 1}**"**

print(**f"3) Критерий авантюры(максимума, оптимизма).**\n **"**

**f"Оптимистическая стратегия соответствует выбору** {optimal\_strategy} **"**

**f"c максимальным выигрышем в матрице -** {matrix.max()}**."**)

return optimal\_strategy

def GurvitzCriteria(matrix: np.array):

*"""*

*Криетрий Гурвица.*

*:param matrix: матрица стратегий.*

*:return полученная оптимальная стратегия.*

*"""*

table = OutMatrix(matrix)

psi = ALPHA \* np.min(matrix, axis=1) + (1 - ALPHA) \* np.max(matrix, axis=1)

table.add\_column(**"Ψ"**, psi)

optimal\_strategy = **f"a**{psi.argmax() + 1}**"**

print(**f"4) Криетрий Гурвица.**\n**"**

**f"Критерий Гурвица определяется из условия равновероятной реализации пессимистической и"**

**f" оптимистической гипотез (α** **=** {ALPHA}**).**\n{table}\n**Наилучшая стратегия** {optimal\_strategy}\n**"**

**f"Ожидаемый выигрыш:** {psi.max()}**"**)

return optimal\_strategy

def SevigeCriteria(matrix: np.array):

*"""*

*Криетрий рисков Севиджа.*

*:param matrix: матрица стратегий.*

*:return полученная оптимальная стратегия.*

*"""*

risks = np.max(matrix, axis=0) - matrix

table = OutMatrix(risks)

max\_col = np.max(risks, axis=1)

table.add\_column(**"αᵢ"**, max\_col)

optimal\_strategy = **f"a**{max\_col.argmin() + 1}**"**

print(**f"5) Критерий Севиджа.**\n**"**

**f"Составим таблицу рисков стратегий:**\n{table}\n**"**

**f"Таким образом, оптимальная рисковая стратегия -** {optimal\_strategy}**."**)

return optimal\_strategy

def ChooseBestStrategy(matrix: np.array):

*"""*

*Отбирает наилучшую стратегию согласно принципу большинства на основе всех рассмотренных критериев.*

*:param matrix: матрица стратегий.*

*"""*

result\_map = {}

for k in [**f"a**{i + 1}**"** for i in range(matrix.shape[0])]:

result\_map[k] = 0

result\_map[BernulliCriteria(matrix)] += 1

result\_map[ValdCriteria(matrix)] += 1

result\_map[OptimismCriteria(matrix)] += 1

result\_map[GurvitzCriteria(matrix)] += 1

result\_map[SevigeCriteria(matrix)] += 1

strategies = list(result\_map.keys())

ranks = list(result\_map.values())

table = PrettyTable()

table.field\_names = strategies

table.add\_row(ranks)

print(**f"**\n**В итоге выберем стратегию, которая оказалась оптимальной в большем числе критериев:**\n{table}\n**"**

**f"По принципу большинства рекомендуем стратегию** {strategies[max(enumerate(ranks), key=lambda x: x[1])[0]]}**"**)