**Федеральное государственное образовательное**

**бюджетное учреждение**

**высшего образования**

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ**

**ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ**

**ФЕДЕРАЦИИ»**

**(Финансовый университет)**

**Факультет**

**информационных технологий и анализа больших данных**

**Кафедра «Бизнес-информатика»**

**Домашнее задание № 4**

«Игры с природой в условиях риска и неопределенности.»

Студенты группы БИ20-8:

Луканина Полина

Аверкин Никита

Филимонова Арина

Совин Владимир

Горшков Георгий

Киселева Евгения

Руководитель:

Аксенов Дмитрий Андреевич

**Москва 2022**

### **Оглавление**

[Оглавление 2](#_heading=h.30j0zll)

[1.](#_heading=h.1fob9te) Постановка задачи (физическая модель) 5

[2.](#_heading=h.3znysh7) Математическая модель 5

[2.1. Решение задачи о принятии оптимального решения в условиях риска 6](#_heading=h.2et92p0)

[2.2. Решение задачи о принятии оптимального решения в условиях неопределенности 8](#_heading=h.tyjcwt)

[3.](#_heading=h.3dy6vkm) Алгоритмы решения задачи 10

[3.1.](#_heading=h.1t3h5sf) Решение задачи о нахождении выигрышной стратегии в условиях риска 10

[3.1.1.](#_heading=h.4d34og8) Описание входных данных. 10

[3.1.2.](#_heading=h.2s8eyo1) Методы вывода информации 12

[3.1.3.](#_heading=h.17dp8vu) Описание входных данных 14

[3.2. Решение задачи о нахождении выигрышной стратегии в условиях неопределенности. 15](#_heading=h.3rdcrjn)

[3.2.1. Описание входных данных 15](#_heading=h.26in1rg)

[3.2.2. Описание алгоритма решения 16](#_heading=h.lnxbz9)

[3.2.3. Описание входных данных 16](#_heading=h.35nkun2)

[4.](#_heading=h.1ksv4uv) Варианты использования системы 18

[4.1.1. ВИ 1 18](#_heading=h.44sinio)

[4.1.2. ВИ 2 20](#_heading=h.2jxsxqh)

[4.1.3. ВИ 3 21](#_heading=h.z337ya)

[4.2. Варианты использования в условиях неопределенности 22](#_heading=h.3j2qqm3)

[4.2.1. ВИ 1 22](#_heading=h.1y810tw)

[4.2.2. ВИ 2 24](#_heading=h.4i7ojhp)

[4.2.3. ВИ 3 26](#_heading=h.2xcytpi)

[5.](#_heading=h.1ci93xb) Архитектура решения 27

[5.1](#_heading=h.3whwml4) Алгоритм 1 27

[5.1.1. Функции считывания информации 27](#_heading=h.2bn6wsx)

[5.1.2. Функции обработки информации 32](#_heading=h.qsh70q)

[5.1.3 Функции вывода информации 40](#_heading=h.2p2csry)

[5.2](#_heading=h.147n2zr) Алгоритм 2 42

[5.2.1. Функции считывания информации 42](#_heading=h.3o7alnk)

[5.2.2. Функции обработки информации 45](#_heading=h.ihv636)

[5.2.3. Функции вывода информации 53](#_heading=h.vx1227)

[6](#_heading=h.3fwokq0) Тестирование 54

[6.1. Тестирование задачи на нахождение выигрышной стратегии в условиях риска 54](#_heading=h.1v1yuxt)

[6.1.1. Проверка №1 матрица 3x3: 54](#_heading=h.4f1mdlm)

[6.1.2. Проверка №2 матрица 3x3: 57](#_heading=h.2u6wntf)

[6.1.3. Проверка №3 матрица 3x3: 60](#_heading=h.19c6y18)

[6.1.4. Проверка №4 матрица 4x4: 63](#_heading=h.3tbugp1)

[6.1.5. Проверка №5 матрица 4x4: 66](#_heading=h.28h4qwu)

[6.2. Тестирование задачи на нахождение выигрышной стратегии в условиях неопределенности 70](#_heading=h.nmf14n)

[6.2.1. Проверка №1 матрица 3x3 70](#_heading=h.37m2jsg)

[6.2.2. Проверка №2 матрица 3x3: 72](#_heading=h.1mrcu09)

[6.2.3. Проверка №3 матрица 3x3: 74](#_heading=h.46r0co2)

[6.2.4. Проверка №4 матрица 4x4: 76](#_heading=h.2lwamvv)

[6.2.5. Проверка №5 матрица 4x4: 78](#_heading=h.111kx3o)

[7](#_heading=h.3l18frh) Заключение 81

[7.1. Заключение по задачи на нахождение выигрышной стратегии в условиях риска 81](#_heading=h.206ipza)

[7.2. Заключение по задачи на нахождение выигрышной стратегии в условиях неопределенности 83](#_heading=h.4k668n3)

# Постановка задачи (физическая модель)

Одна Российская IT-компания производит игры. у нее есть 4 основных продукта (игры), которые приносят большую часть прибыли. Величина прибыли определяется состоянием спроса на рынке, который может находится в одном из 4 рабочих состояний: повышенный спрос, умеренный спрос, пассивная покупательская способность и депрессивный рынок.

Для наглядности зависимости компания предоставила нам таблицу зависимости величины прибыли от вида игры и состояния рынка:

*Таблица 1. Зависимость величины прибыли от вида игры*

| Виды выпускаемых игр | Возможные состояния рынка спроса | | | |
| --- | --- | --- | --- | --- |
| Повышенный | Стагнирующий | Сокращающийся | Депрессивный |
| стратегия | 50 | 30 | 35 | 45 |
| шутер | 40 | 40 | 55 | 40 |
| аркада | 35 | 40 | 45 | 55 |
| гонки | 10 | 20 | 25 | 35 |

Необходимо определить оптимальные пропорции вкладываемых средств и сил компании в виды игр, продажа которых обеспечила бы ему максимально возможную выручку вне зависимости от того, какое состояние спроса будет реализовано.

# Математическая модель

*Таблица 2. Исходные данные*

| A1 | 50 | 30 | 35 | 45 |
| --- | --- | --- | --- | --- |
| A2 | 40 | 40 | 55 | 40 |
| A3 | 35 | 40 | 45 | 55 |
| A4 | 10 | 20 | 25 | 35 |
| pj | 0.15 | 0.25 | 0.5 | 0.1 |

## 2.1. Решение задачи о принятии оптимального решения в условиях риска

1) Критерий Байеса:

По критерию Байеса за оптимальные принимается та стратегия (чистая) Ai, при которой максимизируется средний выигрыш a или минимизируется средний риск r.

Считаем значения ∑(aijpj):

∑(a1,jpj) = 50\*0.15 + 30\*0.25 + 35\*0.5 + 45\*0.1 = 37

∑(a2,jpj) = 40\*0.15 + 40\*0.25 + 55\*0.5 + 40\*0.1 = 47.5

∑(a3,jpj) = 35\*0.15 + 40\*0.25 + 45\*0.5 + 55\*0.1 = 43.25

∑(a4,jpj) = 10\*0.15 + 20\*0.25 + 25\*0.5 + 35\*0.1 = 22.5

*Таблица 3. Значения ∑(aijpj)*

| Ai | П1 | П2 | П3 | П4 | ∑(aijpj) |
| --- | --- | --- | --- | --- | --- |
| A1 | 7.5 | 7.5 | 17.5 | 4.5 | 37 |
| A2 | 6 | 10 | 27.5 | 4 | 47.5 |
| A3 | 5.25 | 10 | 22.5 | 5.5 | 43.25 |
| A4 | 1.5 | 5 | 12.5 | 3.5 | 22.5 |
| pj | 0.15 | 0.25 | 0.5 | 0.1 |  |

Выбираем из (37; 47.5; 43.25; 22.5) максимальный элемент max=47.5

Вывод: выбираем стратегию N=2.

2) Критерий Лапласа:

Если вероятности состояний природы правдоподобны, для их оценки используют принцип недостаточного основания Лапласа, согласно которого все состояния природы полагаются равновероятными, т.е.:

q1 = q2 = ... = qn = 1/n.

qi = ¼

*Таблица 4. Критерий Лапласа*

| Ai | П1 | П2 | П3 | П4 | ∑(aij) |
| --- | --- | --- | --- | --- | --- |
| A1 | 12.5 | 7.5 | 8.75 | 11.25 | 40 |
| A2 | 10 | 10 | 13.75 | 10 | 43.75 |
| A3 | 8.75 | 10 | 11.25 | 13.75 | 43.75 |
| A4 | 2.5 | 5 | 6.25 | 8.75 | 22.5 |
| pj | 0.25 | 0.25 | 0.25 | 0.25 |  |

Выбираем из (40; 43.75; 43.75; 22.5) максимальный элемент max=43.75

Вывод: выбираем стратегию N=2.

3) Критерий Гермейера:

Преобразуем матрицу в соответствии с методом Гермейера:

eij=aij·qj, если aij < 0

eij=aij/qj, если aij > 0

Далее к этой матрице применяется принцип максимина.

Таким образом, новую матрицу необходимо дополнить справа еще одним столбцом, в который нужно внести наименьшие значения элементов каждой строки. Затем из элементов добавленного столбца нужно выбрать наибольший. Строка, в которой он стоит и будет оптимальной стратегией.

| Ai | П1 | П2 | П3 | П4 | min(ei) |
| --- | --- | --- | --- | --- | --- |
| A1 | 333.33333333333 | 120 | 70 | 450 | 70 |
| A2 | 266.66666666667 | 160 | 110 | 400 | 110 |
| A3 | 233.33333333333 | 160 | 90 | 550 | 90 |
| A4 | 66.666666666667 | 80 | 50 | 350 | 50 |
| pj | 0.15 | 0.25 | 0.5 | 0.1 |  |

Выбираем из (70; 110; 90; 50) максимальный элемент max=110

Вывод: выбираем стратегию N=2.

Таким образом, в результате решения статистической игры по различным критериям чаще других рекомендовалась стратегия A2.

## 2.2. Решение задачи о принятии оптимального решения в условиях неопределенности

1) Критерий «пессимизма»

*Таблица 5. Min - min*

| 30 |
| --- |
| 40 |
| 35 |
| **10** |

2) Критерий «оптимизма»

*Таблица 6. Max – max*

| 50 |
| --- |
| **55** |
| **55** |
| 35 |

3) Критерий Вальда

Min - max

**Правило Вальда** (правило крайнего пессимизма).

Рассматривая *i*-e решение будем полагать, что на самом деле ситуация складывается самая плохая, т.е. приносящая самый малый доход ai, но теперь уж выберем решение i0 с наибольшим ai0.

Итак, правило Вальда рекомендует принять решение i0, такое что ai0 = max (ai) = max (min qij)

Так, в нижеуказанной матрице имеем a1 = 30, a2 = 40, a3 = 35 a4 = 10. Из этих чисел максимальным является число 40. Значит, правило Вальда рекомендует принять 2-е решение:

*Таблица 7. Рекомендованное число 40*

| 30 |
| --- |
| **40** |
| 35 |
| 10 |

4) Правило Гурвица (взвешивающее пессимистический и оптимистический подходы к ситуации).:

Принимается решение i, на котором достигается максимум,

где 0 ≤ λ ≤ 1.

Значение λ выбирается из субъективных соображений. Если λ приближается к 1, то правило Гурвица приближается к правилу Вальда, при приближении λ к 0, правило Гурвица приближается к правилу "розового оптимизма":

*Таблица 8. В нашем случае правило Гурвица рекомендует 3-е решение.*

| Alpha | 0 | 0,1 | 0,2 | 0,3 | 0,4 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 | 1 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| А1 | 30 | 32 | 34 | 36 | 38 | 40 | 42 | 44 | 46 | 48 | 50 |
| А2 | **40** | **41,5** | 43 | 44,5 | 46 | 47,5 | 49 | 50,5 | 52 | 53,5 | 55 |
| А3 | 35 | 37 | **39** | **41** | **43** | **45** | **47** | **49** | **51** | **53** | **55** |
| А4 | 10 | 12,5 | 15 | 17,5 | 20 | 22,5 | 25 | 27,5 | 30 | 32,5 | 35 |

5) **Правило Сэвиджа** (правило минимального риска).

При применении этого правила анализируется матрица рисков R = (rij).

Рассматривая *i*-e решение будем полагать, что на самом деле складывается ситуация максимального риска bi = max [rij]

Но теперь уж выберем решение i0 с наименьшим bi0.

Итак, правило Сэвиджа рекомендует принять решение i0, такое что bi0 = min (bi) = min (max rij)

В рассматриваемом примере получим:

*Таблица 9. выбираем решение i0 с наименьшим bi0.*

| 0 | 10 | **20** | 10 |
| --- | --- | --- | --- |
| 10 | 0 | 0 | **15** |
| **15** | 0 | 10 | 0 |
| **40** | 20 | 30 | 20 |

Минимальным из этих чисел является число 15. Т.е. правило Сэвиджа рекомендует принять 2-е или 3-е решение.

# Алгоритмы решения задачи

Алгоритмы решения реализованы с помощью программного кода в Python.

# Решение задачи о нахождении выигрышной стратегии в условиях риска

### Описание входных данных.

Вид входных данных зависит от способа, которым будут вводится данными. Данный алгоритм позволяет самим ввести данные, или ввести с помощью CSV файла.

Для ручного ввода входными данными являются:

* Количество стратегий для компании А, название стратегий для компании А, количество состояний природы, название состояния природы, значения весовой матрицы, значения вектора вероятности.

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 1. Терминал в питоне*

* Количество стратегий игрока А, количество состояний природы, значения вектора вероятности.

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 2. 2 способ ввода данных*

Импорт входных данных из csv файла:

* Весовая матрица с названиями столбцов и строк.



*Рисунок 3. Способ ввода данных 3*

### Методы вывода информации

После того как данные введены, программе необходимо преобразовать данные для дальнейшего использования.

Шаг 1: необходимо рассчитать оценку по критерию Лапласа, для этого нужно рассчитать вероятности: единицу разделить на количество стратегий компании А. Затем с помощью цикла перебираются значения, находится оптимальное.

Шаг 2: необходимо рассчитать оценку Байеса. Для этого необходимо перемножить вероятности, и значения по строке платежной матрицы, а после сложить. Затем с помощью цикла перебрать значения и найти оптимальное.

Шаг 3: необходимо рассчитать оценку по критерию Гермейера, а именно находятся минимальное значение по строке и округляется до трех знаков после запятой.

Шаг 4: берем весовую матрицу, находим минимум каждой строки, а

далее - максимум — это получилась цена игры для игрока А.

Шаг 5: находим максимум для каждого столбца весовой матрицы, и

затем находим минимальное из них — это цена игры для игрока B.

Шаг 6: Переход к канонической форме задачи линейного

программирования путем введения неотрицательных дополнительных

балансовых (базисных) переменных. Запись задачи в симплекс-таблицу.

Между системой ограничений задачи и симплекс-таблицей взаимно-

однозначное соответствие. Строчек в таблице столько, сколько равенств в

системе ограничений, а столбцов - столько, сколько свободных переменных.

Базисные переменные заполняют первый столбец, свободные - верхнюю

строку таблицы. Нижняя строка называется индексной, в ней записываются

коэффициенты при переменных в целевой функции.

Шаг 7: Проверка опорного плана на оптимальность. Для этого

необходимо анализировать строку целевой функции F. Если найдется хотя

бы один коэффициент индексной строки меньше нуля, то план не

оптимальный, и его необходимо улучшить.

Шаг 8: Улучшение опорного плана. Из отрицательных

коэффициентов индексной строки выбирается наибольший по абсолютной

величине. Затем элементы столбца свободных членов симплексной таблицы

делит на элементы того же знака ведущего столбца. Далее идет построение

нового опорного плана.

Шаг 9: Выписывание оптимального решения.

### Описание входных данных

В конце программа рассчитает весовую матрицу, состояние природы, оптимальную стратегию игрока А по критерию Лапласа, цену игры оптимальной стратегией по критерию Лапласа, оптимальную стратегию игрока А по критерию Байеса, цену игры оптимальной стратегией по критерию Байеса, оптимальную чистую стратегию игрока А по критерию Гермейера, цену игры оптимальной чистой стратегией по критерию Гермейера, цену игры для игрока А при выборе смешанной оптимальной стратегии по критерию Гермейера и таблицу оптимальных смешанных стратегий для игрока А по критерию Гермейера.

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 4. Предоставление программой оптимальной стратегии*

## 3.2. Решение задачи о нахождении выигрышной стратегии в условиях неопределенности.

### 3.2.1. Описание входных данных

Вид входных данных зависит от способа, которым будут вводиться данными. Данный алгоритм позволяет самим ввести данные, или ввести с помощью CSV файла.

Для ручного ввода входными данными являются:

* Количество стратегий для компании А, название стратегий для компании А, количество состояний природы, название состояния природы, значения весовой матрицы.

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 5. Терминал питона при 1 способе ввода*

* Количество стратегий игрока А, количество состояний природы, минимальное и максимальное значение.

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 6. Терминал питона при 2 варианте ввода*

Импорт входных данных из csv файла:

* Файл с весовая матрица с названиями столбцов и строк.



*Рисунок 7. Терминал питона при 3 способе ввода*

### 3.2.2. Описание алгоритма решения

После того как данные введены, программе необходимо преобразовать данные для дальнейшего использования.

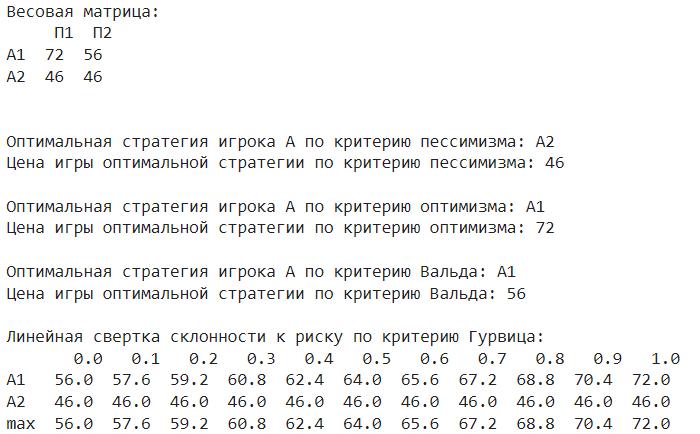
Шаг 1: необходимо транспонировать матрицу.

Шаг 2: необходимо рассчитать стратегию по критерию оптимизма и пессимизма, а также Вальда.

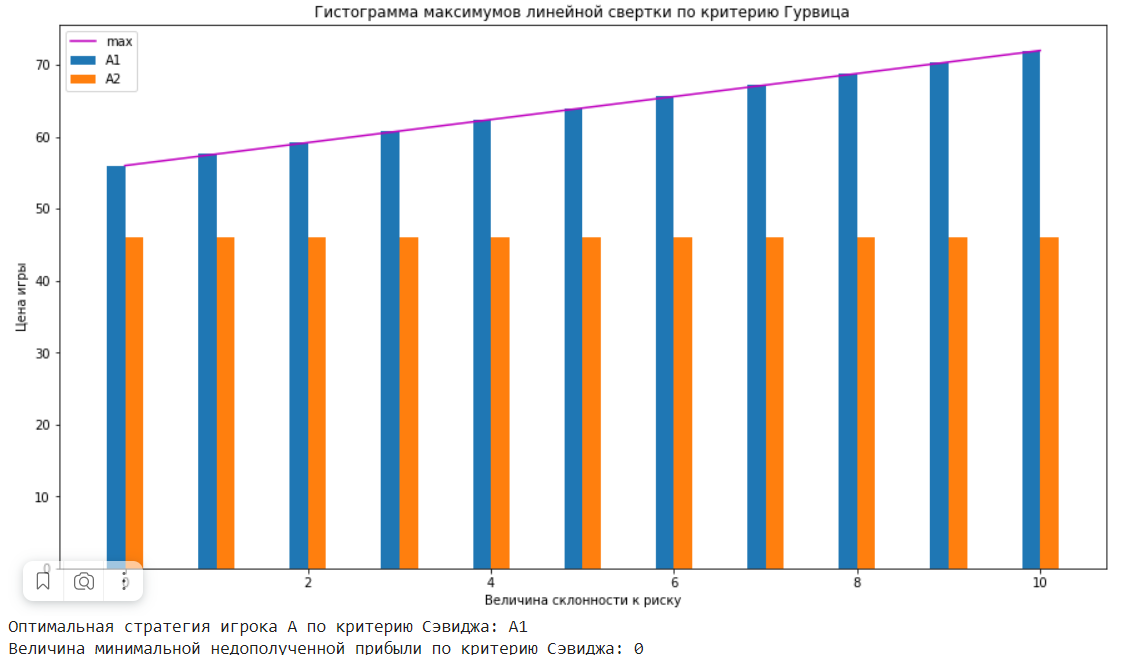
Шаг 3: необходимо рассчитать линейную свертку склонности к риску по критерию Гурвица, а также разработать гистограмму максимумов.

Шаг 4: также создать некую защиту от неверного ввода данных.

### 3.2.3. Описание входных данных

В конце программа выдаст весовую матрицу; оптимальную стратегию игрока А по критерию пессимизма, а также цена игры оптимальной стратегии этому критерию; оптимальную стратегию игрока А по критерию оптимизма, а также цена игры оптимальной стратегии этому критерию; оптимальную стратегию игрока А по критерию Вальда, а также цена игры оптимальной стратегии этому критерию; оптимальная стратегия игрока А по критерию Сэвиджа, а также величину минимальной недополученной прибыли по этому критерию.

*Рисунок 8. Вывод оптимальной стратегии и линейная сверстка*



*Рисунок 9. Гистограмма максимумов*

# Варианты использования системы

В нашей системе есть три варианта использования.

## 4.1.1. ВИ 1

Данный вариант использования включает в себя ручной ввод данных с клавиатуры. Для того, чтобы его активировать в графу «Каким способом вы хотите ввести значения?» надо ввести цифру «1».

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 10. Терминал кода при ВИ 1*

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 11. Оптимальный путь при ВИ 1*

После этого необходимо ввести количество стратегий для компании А цифрой. (Например, 2).

Затем нужно ввести названия стратегий, их количество зависит от того, сколько стратегий вы указали ранее. (Например, Косметика).

Далее переходим к заполнению данных для состояния природы. Мы также вводим количество состояний, а также их названия.

После этого вы должны ввести данные для заполнения весовой матрицы. Количество значений зависит от того, сколько стратегий вы указали ранее. (Например, 45).

Затем вводиться данные вектора вероятности поэлементно. (Например: 0.15). Важно вводить не целые значения через точку.

После этого от пользователя требуется лишь нажатие клавиши «Enter» и на экране выведется весовая матрица, состояние природы, оптимальная стратегия игрока А по критерию Лапласа, цену игры оптимальной стратегией по критерию Лапласа, оптимальную стратегию игрока А по критерию Байеса, цену игры оптимальной стратегией по критерию Байеса, оптимальную чистую стратегию игрока А по критерию Гермейера, цену игры оптимальной чистой стратегией по критерию Гермейера, цену игры для игрока А при выборе смешанной оптимальной стратегии по критерию Гермейера и таблицу оптимальных смешанных стратегий для игрока А по критерию Гермейера.

## 4.1.2. ВИ 2

Данный вариант позволяет сгенерировать весовую матрицу, для его выбора в первой строке необходимо ввести «2».

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 12. Терминал и ответ при 2 ВИ*

Далее Вам необходимо ввести количество стратегий игрока А, а также количество состояний природы.

Затем нужно ввести вектор вероятности поэлементно.

После этого от пользователя требуется лишь нажатие клавиши «Enter» и на экране выведется весовая матрица, состояние природы, оптимальная стратегия игрока А по критерию Лапласа, цену игры оптимальной стратегией по критерию Лапласа, оптимальную стратегию игрока А по критерию Байеса, цену игры оптимальной стратегией по критерию Байеса, оптимальную чистую стратегию игрока А по критерию Гермейера, цену игры оптимальной чистой стратегией по критерию Гермейера, цену игры для игрока А при выборе смешанной оптимальной стратегии по критерию Гермейера и таблицу оптимальных смешанных стратегий для игрока А по критерию Гермейера .

## 4.1.3. ВИ 3

Данный вариант использования включает в себя ввод данных с помощью файла csv. Для того, чтобы его активировать в графу «Каким способом вы хотите ввести значения?» надо ввести цифру «3».

После этого появляется окно, в котором вводим путь к csv файлу. Например:



*Рисунок 13. Терминал при 3 ВИ*

## 4.2. Варианты использования в условиях неопределенности

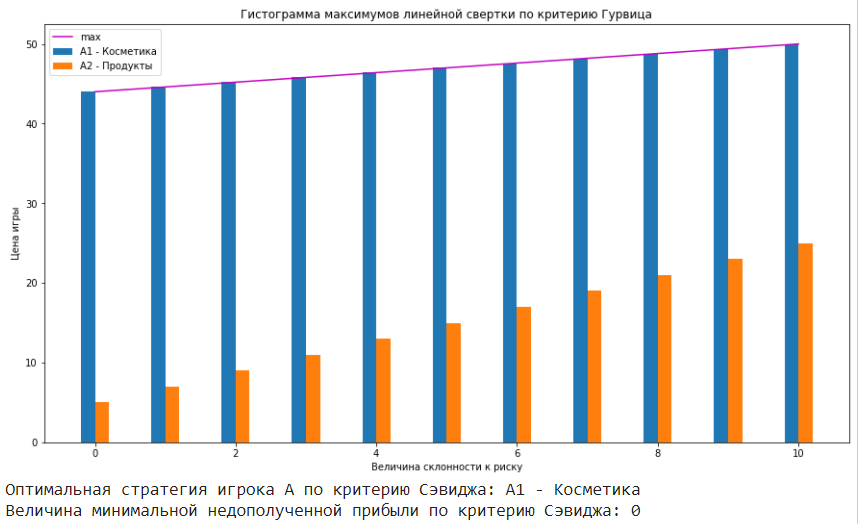
### 4.2.1. ВИ 1

Данный вариант использования включает в себя ручной ввод данных с клавиатуры. Для того, чтобы его активировать в графу «Каким способом вы хотите ввести значения?» надо ввести цифру «1».

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 14. Терминал питона с решением*



*Рисунок 15. Гистограмма максимумов*

После этого необходимо ввести количество стратегий для компании А цифрой. (Например, 2).

Затем нужно ввести названия стратегий, их количество зависит от того, сколько стратегий вы указали ранее. (Например, Косметика).

Далее переходим к заполнению данных для состояния природы. Мы также вводим количество состояний, а также их названия.

После этого вы должны ввести данные для заполнения весовой матрицы. Количество значений зависит от того, сколько стратегий вы указали ранее. (Например, 45).

После этого от пользователя требуется лишь нажатие клавиши «Enter» и на экране выведется весовая матрица; оптимальная стратегия игрока А по критерию пессимизма, а также цена игры оптимальной стратегии этому критерию; оптимальную стратегию игрока А по критерию оптимизма, а также цена игры оптимальной стратегии этому критерию; оптимальную стратегию игрока А по критерию Вальда, а также цена игры оптимальной стратегии этому критерию; оптимальная стратегия игрока А по критерию Сэвиджа, а также величину минимальной недополученной прибыли по этому критерию.

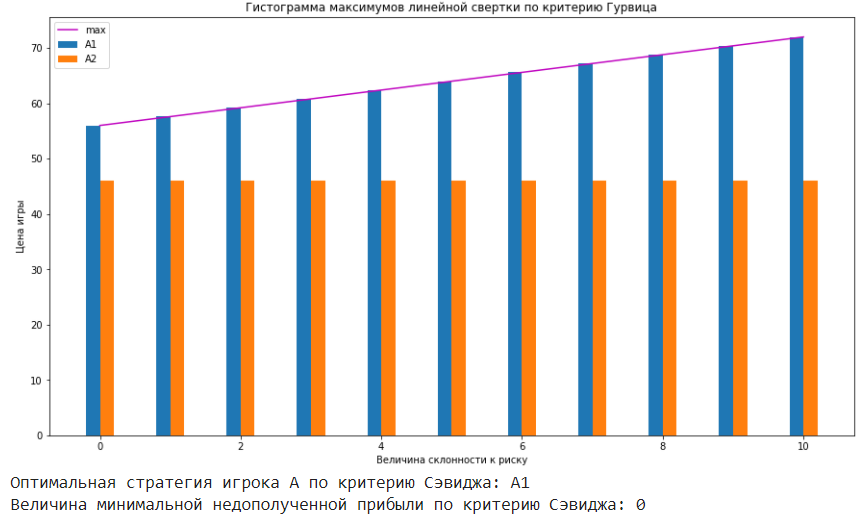
### 4.2.2. ВИ 2

Данный вариант позволяет сгенерировать весовую матрицу, для его выбора в первой строке необходимо ввести «2».

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 16. ВИ с генерацией весовой матрицы*



*Рисунок 17. Гистограмма при данном способе ввода информации*

Далее Вам необходимо ввести количество стратегий игрока А, а также количество состояний природы.

Затем нужно ввести минимальное и максимальное значение.

После этого от пользователя требуется лишь нажатие клавиши «Enter» и на экране выведется весовая матрица; оптимальная стратегия игрока А по критерию пессимизма, а также цена игры оптимальной стратегии этому критерию; оптимальную стратегию игрока А по критерию оптимизма, а также цена игры оптимальной стратегии этому критерию; оптимальную стратегию игрока А по критерию Вальда, а также цена игры оптимальной стратегии этому критерию; оптимальная стратегия игрока А по критерию Сэвиджа, а также величину минимальной недополученной прибыли по этому критерию.

### 4.2.3. ВИ 3

Данный вариант использования включает в себя ввод данных с помощью файла csv. Для того, чтобы его активировать в графу «Каким способом вы хотите ввести значения?» надо ввести цифру «3».

После этого появляется окно, в котором вводим путь к csv файлу. Например:



*Рисунок 18. Способ ввода информации 3*

# Архитектура решения

Для решения задачи использовались методы (функции), которые можно разделить на 3 принципиальных кода.

## Алгоритм 1

### 5.1.1. Функции считывания информации

После запуска программы необходимо определиться каким способом будет происходить ввод данных: 1 – ручной ввод; 2 – случайные числа; 3 – файл csv.

1. **Если введено «1»:**

**Входные данные:**

* Нет входных данных

**Выходные данные:**

* name\_rows – список названий стратегий компании А (тип данных: list);
* name\_columns – список названий стратегий компании В (тип данных: list);
* matrix\_A – список значений весовой платёжной матрицы (тип данных: list);
* matrix\_P – список значений весовой матрицы вероятностей (тип данных: list);

**Переменные, затрагиваемые в ходе работы:**

* sum\_strategy\_A – количество стратегий для компании А (тип данных: int);
* sum\_strategy\_В – количество стратегий для компании В (тип данных: int);
* name\_rows – список названий стратегий компании А (тип данных: list);
* name\_columns – список названий стратегий компании В (тип данных: list);
* matrix\_A – список значений весовой платёжной матрицы (тип данных: list);
* matrix\_P – список значений весовой name – название стратегий (тип данных: str);
* value – элементы весовой матрицы (тип данных: int);

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 19. Часть кода, в которой упоминаются эти функции*

1. **Если введено «2»:**

**Входные данные:**

* Нет входных данных

**Выходные данные:**

* name\_rows – список названий стратегий компании А (тип данных: list);
* name\_columns – список названий стратегий компании В (тип данных: list);
* matrix\_A – список значений весовой платёжной матрицы (тип данных: list);
* matrix\_P – список значений весовой матрицы вероятностей (тип данных: list);

**Переменные, затрагиваемые в ходе работы:**

* name\_rows – список названий стратегий компании А (тип данных: list);
* name\_columns – список названий стратегий компании В (тип данных: list);
* matrix\_A – список значений весовой платёжной матрицы (тип данных: list);
* matrix\_P – список значений весовой матрицы вероятностей (тип данных: list);
* size\_rows – количество стратегий игрока А (тип данных: int);

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 20. Часть кода, в которой были затронуты эти переменные*

1. **Если ведено «3»:**

**Входные данные:**

* Нет входных данных

**Выходные данные:**

* matrix\_A – список значений весовой платёжной матрицы (тип данных: list);
* matrix\_P – список значений весовой матрицы вероятностей (тип данных: list);
* name\_rows – список названий стратегий компании А (тип данных: list);
* name\_columns – список названий стратегий компании В (тип данных: list);

**Переменные, затрагиваемые в ходе работы:**

* name\_rows – список названий стратегий компании А (тип данных: list);
* name\_columns – список названий стратегий компании В (тип данных: list);
* file\_way – путь к файлу (тип данных: str).
* matrix\_A – список значений весовой платёжной матрицы (тип данных: list);
* matrix\_P – список значений весовой матрицы вероятностей (тип данных: list);

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 21. Часть кода, в которой были использованы данные переменные*

### 5.1.2. Функции обработки информации

После того, как вы введете все необходимые данные, программа их получит и начнет первичную обработку.

При выборе ручного ввода все необходимые данные заносятся в словари, с помощью метода append ().

В случае случайной генерации, нужные данные генерируются с помощью random.randint(), при этом программа получает случайное целое число в заданном диапазоне. После этого данные также заносятся в пустые словари.

Если ввод данных осуществляется с помощью файла csv, то для начала файл необходимо открыть и прочитать программе, а затем разделить данные и также занести их в пустой словарь. Более того, программа образует список с названиями колонок матрицы, а также список с названиями строк. Не мало важен и перевод коэффициентов из типа данных str в тип int.

После всех необходимых операций с данными реализуются важные функции, рассчитывающие оптимальную чистую стратегию и оптимальную смешанную стратегию.

В функции pure\_strategy реализуется оптимальная чистая стратегия. Она представляет собой нахождение минимума в каждой строке, а также нахождения максимума в каждом столбце. Это находится с помощью функции min (), max ().

В функции nash\_equilibrium реализуется оптимальная смешанная стратегия. С помощью функции linprog () находится оптимальное решение переменных с помощью, которых мы находим процентное соотношение по стратегиям. Более того, здесь также составляются таблицы смешанных стратегий.

Также в коде присутствует вычисление по критерию Лапласа, Байеса и Граймейра.

Критерий Лапласа реализуется с помощью вычисления вероятностей, а также с помощью цикла перебираются значения, находится оптимальное и записывается в переменную strategy\_Laplace строкового типа.

Критерий Байеса реализуется с помощью цикл, который перебираются значения, находится оптимальное значение и записываются в переменную strategy\_Bayes строкового типа.

Оценка по критерию Гермейера

Далее следует подробное описание входных, выходных и переменных, затрагивается в ходе работы.

**Критерий Лапласа:**

**Входные данные:**

* value – элементы весовой матрицы (тип данных: int);

**Выходные данные:**

* strategy\_Laplace – оптимальная стратегия (тип данных: str);
* average\_Laplace – список средневзвешенных (тип данных: list);

**Переменные, затрагиваемые в ходе работы:**

* strategy\_Laplace – оптимальная стратегия (тип данных: str);
* average\_Laplace – список средневзвешенных (тип данных: str);
* value – элементы весовой матрицы (тип данных: int);
* average – средневзвешенное значение (тип данных: int);
* probability\_Laplace – вероятность состояния природы (тип данных: float).

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 22. Оценка по критерию Лапласа*

**Критерий Байеса:**

**Входные данные:**

* value – элементы весовой матрицы (тип данных: int);
* strategy\_Bayes – матрица Байеса (тип данных: str);
* average – средневзвешенное значение (тип данных: int);
* average\_Bayes – список средневзвешенных значений (тип данных: str);
* matrix\_A\_DF - список значений весовой платёжной матрицы (тип данных: list);

**Выходные данные:**

* strategy\_Bayes – матрица Байеса (тип данных: str);
* average\_Bayes– список средневзвешенных значений (тип данных: str);

**Переменные, затрагиваемые в ходе работы:**

* value – элементы весовой матрицы (тип данных: int);
* strategy\_Bayes – оптимальная стратегия
* average – средневзвешенное значение (тип данных: int);
* average\_Bayes – список средневзвешенных значений (тип данных: str);
* probability\_Bayes – значения вероятностей природы (тип данных: float).

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 23. Оценка по критерию Байеса*

**Критерий Гермейера:**

**Входные данные:**

* matrix\_P\_DF – список значений весовой name – название стратегий (тип данных: str);
* value – элементы весовой матрицы (тип данных: int);
* matrix\_A\_DF - список значений весовой платёжной матрицы (тип данных: list);

**Выходные данные:**

* matrix\_Germeier - матрица Гермейера (тип данных: list);

**Переменные, затрагиваемые в ходе работы:**

* probability\_Germeier – значения вероятностей природы (тип данных: float);
* matrix\_Germeier - матрица Гермейера (тип данных: list);
* value – элементы весовой матрицы (тип данных: int);
* matrix\_P\_DF – список значений весовой name – название стратегий (тип данных: str);
* matrix\_A\_DF - список значений весовой платёжной матрицы (тип данных: list);Изображение выглядит как текст

  Автоматически созданное описание

**Функция pure\_strategy:**

**Входные данные:**

* name\_rows – список названий стратегий компании А (тип данных: list);
* name\_columns – список названий стратегий компании В (тип данных: list);
* probability\_Germeier – значения вероятностей природы (тип данных: float);
* matrix\_Germeier - матрица Гермейера (тип данных: list);

**Выходные данные:**

* probability\_Germeier – значения вероятностей природы (тип данных: float);
* strategy\_Germeier – оптимальное стратегия (тип данных: str);
* average\_Germeier - список с минимальные значениями по стратегии (тип данных: str);
* max\_strategy - значение исходной матрицы (тип данных: int);

**Переменные, затрагиваемые в ходе работы:**

* name\_rows – список названий стратегий компании А (тип данных: list);
* name\_columns – список названий стратегий компании В (тип данных: list);
* probability\_Germeier – значения вероятностей природы (тип данных: float);
* matrix\_Germeier - матрица Гермейера (тип данных: list);
* strategy\_Germeier – оптимальное стратегия (тип данных: str);
* average – минимальные значения по стратегии (тип данных: int);
* average\_Germeier - список с минимальные значениями по стратегии (тип данных: str);
* max\_strategy - значение исходной матрицы (тип данных: int);
* probability\_Germeier - значения вероятностей природы (тип данных: float);
* matrix\_Germeier - матрица Гермейера (тип данных: list);

Изображение выглядит как текст

Автоматически созданное описание

**Функция nash\_equilibrium:**

**Входные данные:**

* name\_rows – список названий стратегий компании А (тип данных: list);
* name\_columns – список названий стратегий компании В (тип данных: list);
* probability\_Germeier - значения вероятностей природы (тип данных: float);
* matrix\_A\_DF - список значений весовой платёжной матрицы (тип данных: list);

**Выходные данные:**

* a – матрица Гермейера в формате numpy.array
* name\_rows – список названий стратегий компании А (тип данных: list);
* name\_columns – список названий стратегий компании В (тип данных: list);
* probability\_Germeier - значения вероятностей природы (тип данных: float);
* matrix\_A\_DF - список значений весовой платёжной матрицы (тип данных: list);
* average\_Germier – список с мин значениями по стратегии матрица Гермейера

**Переменные, затрагиваемые в ходе работы:**

* name\_rows – список названий стратегий компании А (тип данных: list);
* name\_columns – список названий стратегий компании В (тип данных: list);
* probability\_Germeier - значения вероятностей природы (тип данных: float);
* matrix\_A\_DF - список значений весовой платёжной матрицы (тип данных: list);
* res — это массив с вероятностями стратегий игроков (тип данных:
* scipy.optimize.optimize.OptimizeResult );
* mytable - – таблицы, которые выводят стратегии и проценты (тип
* данных: prettytable.prettytable.PrettyTable);
* mix\_strategy\_A – таблица смешанной стратегии для игрока А (тип данных: list);
* a\_T – транспонированная матрица Гермейера (тип данных: int);
* matrix\_Germeier - матрица Гермейера (тип данных: list);
* a – матрица Гермейера в формате numpy.array
* sum\_procent – суммарная вероятность (тип данных: list).

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 26. Часть кода, в которой используются данные переменные*

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 27. Вывод таблицы PrettyTable*

### 5.1.3 Функции вывода информации

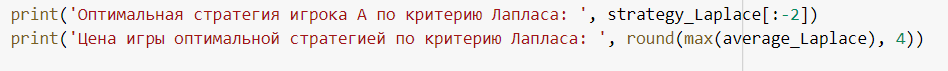
Метод вывода информации (он заключен внутри каждой функции)

Что делает: осуществляет вывод необходимой информации

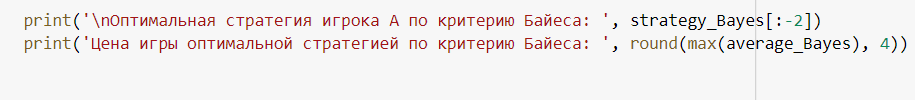
Вывод информации осуществляется с помощью функции print ()

Затрагиваемые переменные:

* matrix\_P – список значений весовой name – название стратегий (тип данных: str);
* strategy\_Laplace – оптимальная стратегия (тип данных: str);
* average\_Laplace – список средневзвешенных (тип данных: list);
* strategy\_Bayes – матрица Байеса (тип данных: str);
* average\_Bayes– список средневзвешенных значений (тип данных: str);
* strategy\_Germeier - оптимальное стратегия (тип данных: str);
* average\_Germeier - список с минимальные значениями по стратегии (тип данных: str);
* max\_strategy - значение исходной матрицы (тип данных: int);
* sum\_procent
* res — это массив с вероятностями стратегий игроков (тип данных:
* scipy.optimize.optimize.OptimizeResult );
* mytable - – таблицы, которые выводят стратегии и проценты (тип
* данных: prettytable.prettytable.PrettyTable);



*Рисунок 28.Вывод оптимальной стратегии по критерию Лапласа*



*Рисунок 29.Вывод оптимальной стратегии по критерию Байеса*



*Рисунок 30. Вывод оптимальной стратегии по критерию Гермейера*



*Рисунок 31. Вывод оптимальной стратегии по критерию Гермейера*



*Рисунок 32. Таблица оптимальных смешанных стратегий для игрока Ф по критерию Гермейера*

## Алгоритм 2

### 5.2.1. Функции считывания информации

После запуска программы необходимо определиться каким способом будет происходить ввод данных: 1 – ручной ввод; 2 – случайные числа; 3 – файл csv.

1. **Если введено «1»:**

**Входные данные:**

* Нет входных данных

**Выходные данные:**

* num - список количества стратегий компании А и количество состояния природы (тип данных: list);

**Переменные, затрагиваемые в ходе работы:**

* sum\_strategy\_company – количество стратегий для компании А (тип данных: int);
* name\_strategy\_company – список названий стратегий компании А (тип данных: list);
* name\_strategy\_nature - cписок названий природы (тип данных: list);
* sum\_strategy\_nature – количество состояния природы (тип данных: int);
* num – список количества стратегий компании А и количество состояния природы (тип данных: list);
* name – название состояния природы (тип данных: str);

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 33. Часть кода с данными переменными*

1. **Если введено «2»:**

**Входные данные:**

* Нет входных данных

**Выходные данные:**

**•** num – список количества стратегий компании А и количество состояния природы (тип данных: list);

**Переменные, затрагиваемые в ходе работы:**

* min\_value – минимальное значение (тип данных: int);
* max\_value – максимальное значение (тип данных: int);
* sum\_strategy\_company – количество стратегий для компании А (тип данных: int);
* sum\_strategy\_nature – количество состояния природы (тип данных: int);
* num – список количества стратегий компании А и количество состояния природы (тип данных: list);

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 34. Часть кода с данными переменными*

1. **Если ведено «3»:**

**Входные данные:**

* Нет входных данных

**Выходные данные:**

* num - список количества стратегий компании А и количество состояния природы (тип данных: list);

**Переменные, затрагиваемые в ходе работы:**

* file\_way – путь к файлу (тип данных: str).
* sum\_strategy\_company – количество стратегий для компании А (тип данных: int);
* name\_strategy\_company – список названий стратегий компании А (тип данных: list);
* name\_strategy\_nature - cписок названий природы (тип данных: list);
* sum\_strategy\_nature – количество состояния природы (тип данных: int);
* num – список количества стратегий компании А и количество состояния природы (тип данных: list);
* name – название состояния природы (тип данных: str);

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 35. Часть кода с данными переменными*

### 5.2.2. Функции обработки информации

После того, как вы введете все необходимые данные, программа их получит и начнет первичную обработку.

При выборе ручного ввода все необходимые данные заносятся в словари, с помощью метода append ().

В случае случайной генерации, нужные данные генерируются с помощью random.randint(), при этом программа получает случайное целое число в заданном диапазоне. После этого данные также заносятся в пустые словари.

Если ввод данных осуществляется с помощью файла csv, то для начала файл необходимо открыть и прочитать программе, а затем разделить данные и также занести их в пустой словарь. Более того, программа образует список с названиями колонок матрицы, а также список с названиями строк. Не мало важен и перевод коэффициентов из типа данных str в тип int.

После всех необходимых операций с данными реализуются важные функции, рассчитывающие оптимальную чистую стратегию и оптимальную смешанную стратегию.

В функции transpose\_mtrx транспонируется матрица с помощью метода transpose.

В функции input\_matrix осуществляется создание матрицы вручную. Это обеспечивается за счёт ввода элементов с помощью int (input ().

В функции PesOpVal просчитывается стратегия по критерию пессимизма, оптимизма, Вальда. Здесь необходимые значения подсчитываются с помощью цикла for i in range, подсчитываются количество стратегий компании А.

В функции Gurvits осуществляется линейная свертка склонности к риску по критерию Гурвица, а также построение гистограммы. Здесь необходимые значения подсчитываются с помощью цикла for i in range, подсчитываются количество стратегий компании А. График же формируется с помощью метода plt.

В функции Sevige подсчитывается оптимальная стратегия по критерию Сэвиджа. Здесь необходимые значения подсчитываются с помощью цикла for i in range, подсчитываются количество стратегий компании А.

В функции sum\_strategy реализуется защита от неверного ввода данных с помощью конструкции try и except, а также с помощью конструкции if.

**Функция transpose\_mtrx:**

**Входные данные:**

* а – матрица (тип данных: int);

**Выходные данные:**

* trns - транспонированная матрица (тип данных: int);

**Переменные, затрагиваемые в ходе работы:**

* trns - транспонированная матрица (тип данных: int);
* а – матрица (тип данных: int);

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 36. Транспонирование матрицы*

**Функция input\_matrix:**

**Входные данные:**

* x – количество стратегий для компании А
* y – количество стратегий состояния природы

**Выходные данные:**

* num – элемент весовой матрицы (тип данных: int);

**Переменные, затрагиваемые в ходе работы:**

* num – элемент весовой матрицы (тип данных: int);

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 37. Функция создания матрицы вручную*

**Функция PesOpVal:**

**Входные данные:**

* num – элемент весовой матрицы (тип данных: int);

**Выходные данные:**

* answr – список с оптимальными значениями по критериям (тип данных: list);
* pes – список значений по стратегии пессимизма (тип данных: list);
* opt - список значений по стратегии оптимизма (тип данных: list);

**Переменные, затрагиваемые в ходе работы:**

* min\_and\_max – список минимальных и максимальных значений весовой матрицы (тип данных: list);
* row – строки матрицы (тип данных: list);
* Vald – список минимальных значений по критерию Вальда (тип данных: int)
* transpose\_min\_and\_max – транспонированная матрица (тип данных: int)
* answr – цена игры оптимальной стратегии по критериям (тип данных: list);
* num – элемент весовой матрицы (тип данных: int);
* pes – список значений по стратегии пессимизма (тип данных: list);
* opt - список значений по стратегии оптимизма (тип данных: list);
* strategy\_Vald - список значений по стратегии оптимизма (тип данных: list);Изображение выглядит как текст

  Автоматически созданное описаниеИзображение выглядит как текст

  Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

**Функция Gurvits:**

**Входные данные:**

* num – элемент весовой матрицы (тип данных: int);

**Выходные данные:**

* DF\_matrix\_Gurvits - список матрицы Гурвица (тип данных: list);

**Переменные, затрагиваемые в ходе работы:**

* DF\_matrix\_Gurvits – список матрицы Гурвица (тип данных: list);
* num – элемент весовой матрицы (тип данных: int);
* min\_and\_max – список минимальных значений весовой матрицы (тип данных: list);
* matrix\_Gurvits – матрица Гурвица (тип данных: int);
* row – строки матрицы (тип данных: list);
* transpose\_matrix\_Gurvits – транспонированная матрица Гурвица (тип данных: int);
* answr - – цена игры оптимальной стратегии по критериям (тип данных: list);

Изображение выглядит как текст

Автоматически созданное описание

**Функция Sevige:**

**Входные данные:**

* num – элемент весовой матрицы (тип данных: int);

**Выходные данные:**

* answr – цена игры оптимальной стратегии по критериям (тип данных: list);
* strategy\_Sevige – список оптимальных значений по стратегии Сэвиджа (тип данных: list);

**Переменные, затрагиваемые в ходе работы:**

* transpose\_num – транспонированная весовая матрица (тип данных: int);
* max\_value – максимальные значения весовой матрицы (тип данных: int);
* matrix Sevige – матрица Сэвиджа (тип данных: int);
* row – строки матрицы (тип данных: list);
* answr – цена игры оптимальной стратегии по критериям (тип данных: list);
* strategy\_Sevige - список оптимальных значений по стратегии Сэвиджа (тип данных: list);

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 40. Оптимальная стратегия по критерию Сэвиджа*

**Функция sum\_strategy:**

**Входные данные:**

* нет входных данных

**Выходные данные:**

* x – количество стратегий для компании А
* y – количество стратегий состояния природы

**Переменные, затрагиваемые в ходе работы:**

* x – количество стратегий для компании А
* y – количество стратегий состояния природы

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 41. Защита от неверного ввода данных*

### 5.2.3. Функции вывода информации

Метод вывода информации (он заключен внутри каждой функции)

Что делает: осуществляет вывод необходимой информации

Вывод информации осуществляется с помощью функции print ()

**Затрагиваемые переменные:**

* answr – цена игры оптимальной стратегии по критериям (тип данных: list);
* pes – список значений по стратегии пессимизма (тип данных: list);
* opt - список значений по стратегии оптимизма (тип данных: list);
* strategy\_Vald -
* DF\_matrix\_Gurvits - список матрицы Гурвица (тип данных: list);
* strategy\_Sevige – список оптимальных значений по стратегии Сэвиджа (тип данных: list);



*Рисунок 42. Оптимальная стратегия по критерию пессимизма*



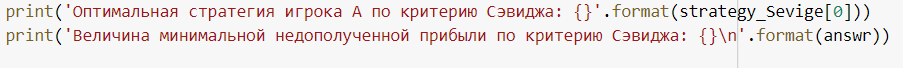
*Рисунок 43. Оптимальная стратегия по критерию оптимизма*



*Рисунок 44.. Оптимальная стратегия по критерию Вальда*



*Рисунок 45. Оптимальная стратегия по критерию Гурвица*



*Рисунок 46. Оптимальная стратегия по критерию Сэвиджа*

# Тестирование

Проведём тестирование нашей программы и сравним полученные показатели, чтобы сделать вывод о предпочтительном варианте использования нашей программы или excel-заготовки под условия заказчика.

## 6.1. Тестирование задачи на нахождение выигрышной стратегии в условиях риска

### 6.1.1. Проверка №1 матрица 3x3:

#### 6.1.1.1. Тест №1 кодом Python:

Выбираем размер матрицы, задаём названия столбцов и строк, вводим вероятности и данные матрицы и получаем оптимальную стратегию по критерию Байеса, Лапласа и Гермейера.

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 47. Решение в питон*

#### 6.1.1.2. Тест №1 Excel:

Вводим необходимые данные в матрицу, проверяем корректность введённых данных и получаем искомые данные по критерию Байеса, Лапласа и Гермейера.

Изображение выглядит как стол

Автоматически созданное описание

*Рисунок 48. Платежная матрица*

Изображение выглядит как стол

Автоматически созданное описание

*Рисунок 49. Оценка по критерию Байеса*

Изображение выглядит как стол

Автоматически созданное описание

*Рисунок 50. Оценка по критерию Лапласа*

Изображение выглядит как стол

Автоматически созданное описание

*Рисунок 51. Матрица Гермейера*

### 6.1.2. Проверка №2 матрица 3x3:

#### 6.1.2.1. Тест №2 кодом Python:

Выбираем размер матрицы, задаём названия столбцов и строк, вводим вероятности и данные матрицы и получаем оптимальную стратегию по критерию Байеса, Лапласа и Гермейера.

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 52. Решение в питоне 2*

#### 6.1.2.2. Тест №2 Excel:

Вводим необходимые данные в матрицу, проверяем корректность введённых данных и получаем искомые данные по критерию Байеса, Лапласа и Гермейера.

Изображение выглядит как стол

Автоматически созданное описание

*Рисунок 53. Платежная матрица 2*

Изображение выглядит как стол

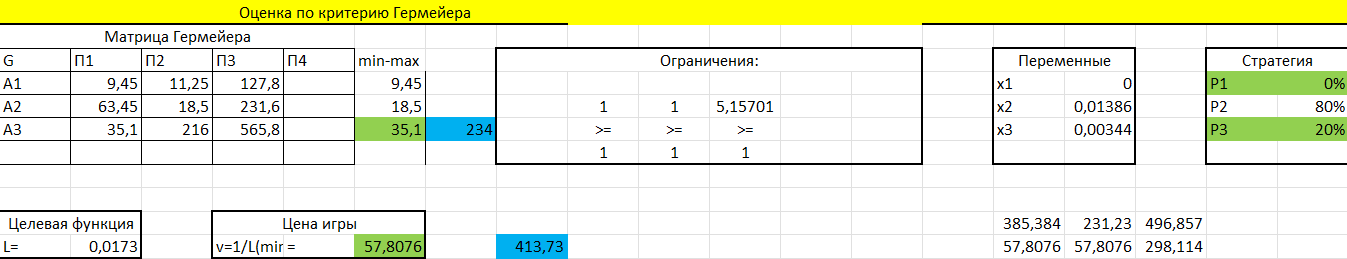
Автоматически созданное описание

*Рисунок 54. Оценка по критерию Байеса 2*

Изображение выглядит как стол

Автоматически созданное описание

*Рисунок 55. Оценка по критерию Лапласа*



*Рисунок 56. Матрица Гермейера 2*

### 6.1.3. Проверка №3 матрица 3x3:

#### 6.1.3.1. Тест №3 кодом Python:

Выбираем размер матрицы, задаём названия столбцов и строк, вводим вероятности и данные матрицы и получаем оптимальную стратегию по критерию Байеса, Лапласа и Гермейера.

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 57. Решение в питоне 3*

#### 6.1.3.2. Тест №3 Excel:

Вводим необходимые данные в матрицу, проверяем корректность введённых данных и получаем искомые данные по критерию Байеса, Лапласа и Гермейера.

Изображение выглядит как стол

Автоматически созданное описание

*Рисунок 58. Платежная матрица 3*

Изображение выглядит как стол

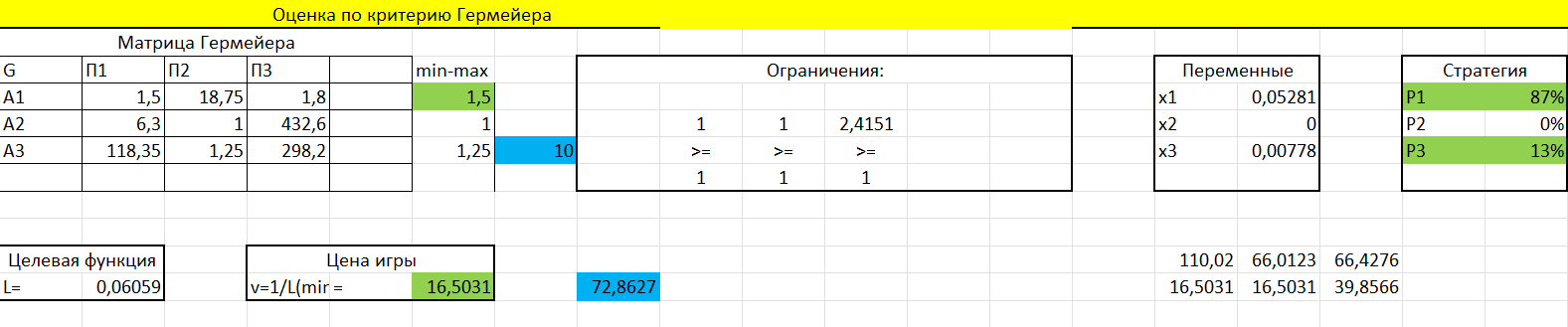
Автоматически созданное описание

*Рисунок 59. Оценка по критерию Байеса 3*

Изображение выглядит как стол

Автоматически созданное описание

*Рисунок 60. Оценка по критерию Лапласа 3*



*Рисунок 61.Матрица Гермейера 3*

### 6.1.4. Проверка №4 матрица 4x4:

#### 6.1.4.1. Тест №4 кодом Python:

Выбираем размер матрицы, задаём названия столбцов и строк, вводим вероятности и данные матрицы и получаем оптимальную стратегию по критерию Байеса, Лапласа и Гермейера.

Изображение выглядит как стол

Автоматически созданное описание

*Рисунок 62.Решение в питоне 4*

#### 6.1.4.2. Тест №4 Excel:

Вводим необходимые данные в матрицу, проверяем корректность введённых данных и получаем искомые данные по критерию Байеса, Лапласа и Гермейера.

Изображение выглядит как стол

Автоматически созданное описание

*Рисунок 63. Платежная матрица 4*

Изображение выглядит как стол

Автоматически созданное описание

*Рисунок 64. Оценка по критерию Байеса 4*

Изображение выглядит как стол

Автоматически созданное описание

*Рисунок 65. Оценка по критерию Лапласа 4*

Изображение выглядит как стол

Автоматически созданное описание

*Рисунок 66. Матрица Гермейера 4*

### 6.1.5. Проверка №5 матрица 4x4:

#### 6.1.5.1. Тест №5 кодом Python:

Выбираем размер матрицы, задаём названия столбцов и строк, вводим вероятности и данные матрицы и получаем оптимальную стратегию по критерию Байеса, Лапласа и Гермейера.

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 67. Результат тестирования в питоне 5*

#### 6.1.5.2. Тест №5 Excel:

Вводим необходимые данные в матрицу, проверяем корректность введённых данных и получаем искомые данные по критерию Байеса, Лапласа и Гермейера.

Изображение выглядит как стол

Автоматически созданное описание

*Рисунок 68. Платежная матрица 5*

Изображение выглядит как стол

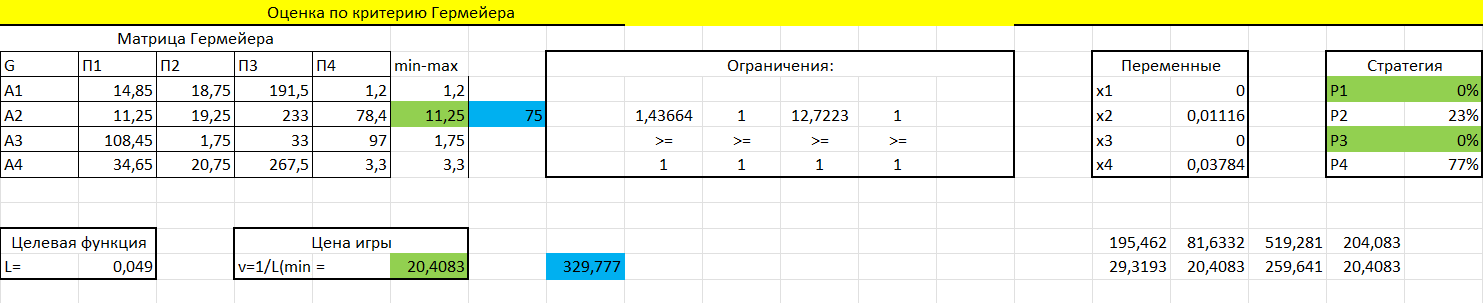
Автоматически созданное описание

*Рисунок 69. Оценка по критерию Байеса 5*

Изображение выглядит как стол

Автоматически созданное описание

*Рисунок 70. Оценка по критерию Лапласа 5*



*Рисунок 71. Матрица Гермейера 5*

Для подсчёта времени на выполнение алгоритмов будет использовано время, затраченное на ввод необходимых данных в предназначенные поля и расчёт данных самой программой с дальнейшим выводом их пользователю на экран.

*Таблица 10. Время, затраченное на выполнение программы*

| **Входные данные** | **Python** | **Excel** |
| --- | --- | --- |
| Тестирование №1 | ~25 секунд | ~50 секунд |
| Тестирование №2 | ~25 секунд | ~51 секунд |
| Тестирование №3 | ~23 секунд | ~49 секунд |
| Тестирование №4 | ~22 секунд | ~50 секунд |
| Тестирование №5 | ~22 секунд | ~48 секунд |

## 6.2. Тестирование задачи на нахождение выигрышной стратегии в условиях неопределенности

### 6.2.1. Проверка №1 матрица 3x3

#### 6.2.1.1. Тест №1 кодом Python:

Выбираем размер матрицы, для неё вводим диапазон допустимых значений и получаем оптимальную стратегию по критерию Вальда, гистограмму максимумов линейной свёртки по критерию Гурвица и оптимальную стратегию по критерию Сэвиджа.

Изображение выглядит как текст, канцелярские товары

Автоматически созданное описание

*Рисунок 72. Решение в питоне 1*

#### 6.2.1.2. Тест №1 онлайн-калькулятором:

Вводим необходимые данные в матрицу онлайн-калькулятора[[1]](#footnote-0), выбираем критерий «Максимально ожидаемый выигрыш», критерий Вальда, критерий Сэвиджа и обобщенный критерий Гурвица и запускам программу.

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 73. Критерий Вальда 1*



*Рисунок 74. Критерий Сэвиджа 1*



*Рисунок 75. Критерий Гурвица 1*

### 6.2.2. Проверка №2 матрица 3x3:

#### 6.2.2.1. Тест №2 кодом Python:

Выбираем размер матрицы, для неё вводим диапазон допустимых значений и получаем оптимальную стратегию по критерию Вальда, гистограмму максимумов линейной свёртки по критерию Гурвица и оптимальную стратегию по критерию Сэвиджа.

Изображение выглядит как текст, канцелярские товары, снимок экрана

Автоматически созданное описание

*Рисунок 76. Решение в питоне 2*

#### 6.2.2.2. Тест №2 онлайн-калькулятором:

Вводим необходимые данные в матрицу онлайн-калькулятора[[2]](#footnote-1), выбираем критерий «Максимально ожидаемый выигрыш», критерий Вальда, критерий Сэвиджа и обобщённый критерий Гурвица и запускам программу.



*Рисунок 77. Критерий Вальда 2*



*Рисунок 78. Критерий Сэвиджа 2*



*Рисунок 79. Критерий Гурвица 2*

### 6.2.3. Проверка №3 матрица 3x3:

#### 6.2.3.1. Тест №3 кодом Python:

Выбираем размер матрицы, для неё вводим диапазон допустимых значений и получаем оптимальную стратегию по критерию Вальда, гистограмму максимумов линейной свёртки по критерию Гурвица и оптимальную стратегию по критерию Сэвиджа.

Изображение выглядит как текст, канцелярские товары

Автоматически созданное описание

*Рисунок 80. Решение в питоне 3*

#### 6.2.3.2. Тест №3 онлайн-калькулятором:

Вводим необходимые данные в матрицу онлайн-калькулятора[[3]](#footnote-2), выбираем критерий «Максимально ожидаемый выигрыш», критерий Вальда, критерий Сэвиджа и обобщённый критерий Гурвица и запускам программу.



*Рисунок 81. Критерий Вальда 3*



*Рисунок 82. Критерий Сэвиджа 3*



*Рисунок 83. Критерий Гурвица 3*

### 6.2.4. Проверка №4 матрица 4x4:

#### 6.2.4.1. Тест №4 кодом Python:

Выбираем размер матрицы, для неё вводим диапазон допустимых значений и получаем оптимальную стратегию по критерию Вальда, гистограмму максимумов линейной свёртки по критерию Гурвица и оптимальную стратегию по критерию Сэвиджа.

Изображение выглядит как текст, канцелярские товары

Автоматически созданное описание

*Рисунок 84. Решение в питоне 4*

#### 6.2.4.2. Тест №4 онлайн-калькулятором:

Вводим необходимые данные в матрицу онлайн-калькулятора[[4]](#footnote-3), выбираем критерий «Максимально ожидаемый выигрыш», критерий Вальда, критерий Сэвиджа и обобщённый критерий Гурвица и запускам программу.



*Рисунок 85. Критерий Вальда 4*

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 86. Критерий Сэвиджа 4*



*Рисунок 87. Критерий Гурвица 4*

### 6.2.5. Проверка №5 матрица 4x4:

#### 6.2.5.1. Тест №5 кодом Python:

Выбираем размер матрицы, для неё вводим диапазон допустимых значений и получаем оптимальную стратегию по критерию Вальда, гистограмму максимумов линейной свёртки по критерию Гурвица и оптимальную стратегию по критерию Сэвиджа.

Изображение выглядит как текст, канцелярские товары, письменная принадлежность, карандаш

Автоматически созданное описание

*Рисунок 88. Решение в питоне 5*

#### 6.2.5.2. Тест №5 онлайн-калькулятором:

Вводим необходимые данные в матрицу онлайн-калькулятора[[5]](#footnote-4), выбираем критерий «Максимально ожидаемый выигрыш», критерий Вальда, критерий Сэвиджа и обобщённый критерий Гурвица и запускам программу.



*Рисунок 89. Критерий Вальда 5*



*Рисунок 90. Критерий Сэвиджа 5*



*Рисунок 91. Критерий Гурвица 5*

Для подсчёта времени на выполнение алгоритмов будет использовано время, затраченное на ввод необходимых данных в предназначенные поля и расчёт данных самой программой с дальнейшим выводом их пользователю на экран.

*Таблица 11. Сравнение решения в питоне и эксель*

| **Входные данные** | **Python** | **Онлайн-калькулятор** |
| --- | --- | --- |
| Тестирование №1 | ~15 секунд | ~18 секунд |
| Тестирование №2 | ~14 секунд | ~18 секунд |
| Тестирование №3 | ~14 секунд | ~17 секунд |
| Тестирование №4 | ~13 секунд | ~17 секунд |
| Тестирование №5 | ~13 секунд | ~17 секунд |

# Заключение

## 7.1. Заключение по задачи на нахождение выигрышной стратегии в условиях риска

Наш представленный код решает поставленную задачу. На основании тестирования данного алгоритма можно сделать вывод о том, что Python выводит самое оптимальное решение достаточно быстро. Ниже представлено решение поставленной задачи через Python:

Изображение выглядит как стол

Автоматически созданное описание

*Рисунок 92. Решение поставленной задачи через питон*

Теперь сравним два алгоритма по критериям: эффективности, скорости использования алгоритма, простоты использования, надёжности в разрезе человеческого фактора и точности предоставляемого решения.

*Таблица 12. Сравнение Питона и Эксель*

| **Критерий** | **Python** | **Excel** |
| --- | --- | --- |
| Эффективность | Высокая | Высокая |
| Скорость использования алгоритма | Высокая | Низкая |
| Простота использования | Высокая | Низкая |
| Надёжность (человеческий фактор) | Высокая | Средняя |
| Точность | Высокая | Высокая |

Мы считаем, что представленный рукописный код на языке Python лучше, потому что он удобнее, быстрее и проще, а ещё имеет импорт .csv файлов и случайный ввод данных. Улучшением кода, к примеру, может послужить время выполнения запроса.

## 7.2. Заключение по задачи на нахождение выигрышной стратегии в условиях неопределенности

Наш представленный код решает поставленную задачу. На основании тестирования данного алгоритма можно сделать вывод о том, что Python выводит самое оптимальное решение достаточно быстро. Ниже представлено решение поставленной задачи через Python и проверка в онлайн-калькуляторе:

Изображение выглядит как текст, канцелярские товары, письменная принадлежность, карандаш

Автоматически созданное описание

*Рисунок 93. Решение поставленной задачи через питон*

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 94. Метод Вальда*

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 95. Критерий Сэвиджа*

Изображение выглядит как текст

Автоматически созданное описание

*Рисунок 96. Критерий Гурвица*

Теперь сравним два алгоритма по критериям: эффективности, скорости использования алгоритма, простоты использования, надёжности в разрезе человеческого фактора и точности предоставляемого решения.

*Таблица 13. Сравнение питона и онлайн-калькулятора*

| **Критерий** | **Python** | **Онлайн-калькулятор** |
| --- | --- | --- |
| Эффективность | Высокая | Высокая |
| Скорость использования алгоритма | Высокая | Средняя |
| Простота использования | Высокая | Высокая |
| Надёжность (человеческий фактор) | Высокая | Средняя |
| Точность | Высокая | Высокая |

Мы считаем, что представленный рукописный код на языке Python лучше, потому что он удобнее, быстрее и проще, а ещё имеет импорт .csv файлов и случайный ввод данных. Улучшением кода, к примеру, может послужить добавление времени выполнения запроса.

1. <https://math.semestr.ru/games/stat.php> [↑](#footnote-ref-0)
2. <https://math.semestr.ru/games/stat.php> [↑](#footnote-ref-1)
3. <https://math.semestr.ru/games/stat.php> [↑](#footnote-ref-2)
4. <https://math.semestr.ru/games/stat.php> [↑](#footnote-ref-3)
5. <https://math.semestr.ru/games/stat.php> [↑](#footnote-ref-4)