**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»**

**Расчетно-аналитическая работа по дисциплине**

**«Математические методы принятия решений»**

Студент группы БИ20-4:

Алифанов Илья

Руководитель:

Аксенов Дмитрий Андреевич

Подпись:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Москва 2022**

Оглавление

[**1.** **Домашнее задание №1 “Транспортная задача”** 2](#_Toc106403105)

[**1.1.** **Условие задачи** 2](#_Toc106403106)

[**1.2.** **Описание выполнения задачи** 3](#_Toc106403107)

[**1.2.1.** **Математическая модель** 3](#_Toc106403108)

[**1.2.2.** **Реализация** 6](#_Toc106403109)

[**1.2.3.** **Тесты** 9](#_Toc106403110)

[**1.3.** **Итоги** 10](#_Toc106403111)

[**2.** **Домашнее задание №2 “Решение задачи коммивояжёра”** 12](#_Toc106403112)

[**2.1.** **Условие задачи** 12](#_Toc106403113)

[**2.2.** **Описание выполнения задачи** 12](#_Toc106403114)

[**2.2.1.** **Математическая модель** 12](#_Toc106403115)

[**2.2.2.** **Реализация** 13](#_Toc106403116)

[**2.2.3.** **Тесты** 14](#_Toc106403117)

[**2.3.** **Итоги** 16](#_Toc106403118)

[**3.** **Домашнее задание №3-4 “Теория игр”** 18](#_Toc106403119)

[**3.1.** **Условие задачи** 18](#_Toc106403120)

[**3.2.** **Описание выполнения задачи** 18](#_Toc106403121)

[**3.2.1.** **Математическая модель** 19](#_Toc106403122)

[**3.2.2.** **Реализация** 19](#_Toc106403123)

[**3.2.3.** **Тесты** 20](#_Toc106403124)

[**4.1.** **Итоги** 24](#_Toc106403125)

1. **Домашнее задание №1 “Транспортная задача”**
   1. **Условие задачи**

Типография думает как максимизировать свою прибыль. Информация о материалах мы видим в Таблица 1 Количество материалов на складе приведено в Таблица 2. Прибыль от реализации единицы продукции приведена в Таблица 3. Требуется выяснить какого жанра книги (учебная, справочная или детская) и в каком количестве выгоднее производить типографии

Таблица 1 – «Материалы и товары»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Бумага | Чернила | Картон | Нитки | Клей |
| Учебная | 20 | 3 | 2 | 8 | 3 |
| Справочная | 10 | 2 | 3 | 4 | 3 |
| Детская | 15 | 5 | 2 | 6 | 3 |

Таблица 2 – «Ограничение по материалам»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Бумага | Чернила | Картон | Нитки | Клей |
| Склад | 200 | 150 | 200 | 100 | 300 |

Таблица 3 – «Цены продукции»

|  |  |  |  |
| --- | --- | --- | --- |
|  | Учебная | Справочная | Детская |
| Цена | 800 | 400 | 550 |

* 1. **Описание выполнения задачи**
     1. **Математическая модель**

Транспортная задача формулируется следующим образом. Имеется m пунктов отправления (или пунктов производства) …, , в которых сосредоточены запасы однородных продуктов в количестве , ..., единиц. Имеется n пунктов назначения (или пунктов потребления) , ..., , потребность которых в указанных продуктах составляет , ..., единиц. Известны также транспортные расходы , связанные с перевозкой единицы продукта из пункта в пункт , i 1, …, m; j 1, ..., n.

Предположим, что общий объем производства равен общему объему потребления. Требуется составить такой план перевозок (откуда, куда и сколько единиц продукта везти), чтобы удовлетворить спрос всех пунктов потребления за счет реализации всего продукта, произведенного всеми пунктами производства, при минимальной общей стоимости всех перевозок. Приведенная формулировка транспортной задачи называется замкнутой транспортной моделью.

Переменными (неизвестными) транспортной задачи являются x\_ij , i=1,2,...,m j=1,2,...,n — объемы перевозок от i-го поставщика каждому j-му потребителю. Эти переменные могут быть записаны в виде матрицы перевозок:

- матрица перевозок (1)

Так как произведение \* определяет затраты на перевозку груза от i-го поставщика j-му потребителю, то суммарные затраты на перевозку всех грузов равны:

- суммарные затраты (2)

По условию задачи требуется обеспечить минимум суммарных затрат. Следовательно, целевая функция задачи имеет вид:

–целевая функция (3)

Система ограничений задачи состоит из двух групп уравнений. Первая группа из m уравнений описывает тот факт, что запасы всех m поставщиков вывозятся полностью и имеет вид:

– группа из m уравнений (4)

Вторая группа из n уравнений выражает требование удовлетворить запросы всех n потребителей полностью и имеет вид:

– группа из n уравнений (5)

Учитывая условие не отрицательности объемов перевозок математическая модель выглядит следующим образом:



В рассмотренной модели транспортной задачи предполагается, что суммарные запасы поставщиков равны суммарным запросам потребителей, т.е.:

- суммарные запасы поставщиков и запросы потребителей

Такая задача называется задачей с правильным балансом, а модель задачи закрытой. Если же это равенство не выполняется, то задача называется задачей с неправильным балансом, а модель задачи — открытой.

Математическая формулировка транспортной задачи такова: найти переменные задачи X=(), i=1,2,...,m; j=1,2,...,n, удовлетворяющие системе ограничений (цифра 2 на математической модели) , (3), условиям не отрицательности (4) и обеспечивающие минимум целевой функции (1)

* + 1. **Реализация**

При запуске кода на входе от пользователя требуется виды товаров и материалов в виде целочисленных неотрицательных показателей. Это станет и размерностью матрицы

Далее составляется пустой список с помощью пустых квадратных скобок. При корректно веденных ранее данных программа запросит значения необходимые для составления матрицы (целое или десятичное неотрицательное число, для этого использовался тип float). Их (значений) количество будет зависеть от введенного ранее размера. По окончании программа составит матрицу и выведет её в соответствии количеством строк и столбцов. Для того, чтобы значения в ней были представлены не в строчку через запятую, а в привычном виде используем функцию «np.array()». Для того, чтобы привести к необходимому виду транспонируем матрицу с помощью функции «.transpose()». Далее выводим результат через print.

Следующим шагом от пользователя на входе требуется внести цену товаров с минусом (целое или десятичное число, для этого использовался тип float). Это делается для корректности ввода и последующей обработки данных, поскольку для вычисления методом линейного программирования (linprog()) используется библиотека scipy. Optimize, при определение максимума (по умолчанию определяется минимум коэффициенты целевой функции). В конце будут выведены значения цены в виде списка. Для этого ранее был создан пустой список price с помощью пустых квадратных скобок, в который уже благодаря функции «.append()» были внесены показатели с клавиатуры пользователя, по количеству не превышающему введенному ранее значению, соответствующему «вид товара»/переменной a (for i in range(a))

Переходим к ограничениям. Ими является количество материалов, хранящихся на складе, поскольку значения, как при составлении матрицы по расходам и списка цены, может являться числом дробным и неотрицательным, используется тип данных float. В начале также создаем пустой список под названием «quantity», где количество значений будет равняться указанному числу видов материалов/переменной b. После получим формат характерный для предыдущих данных (по цене).

Следующим шагом приступаем к формированию функции. Для этого создаем её с помощью def и называем как function. После чего инициируем пустой список под названием «f» и вводим переменную. Далее создаем список func1, где с помощью инициируемой функции, поступающие значения будут вычисляться по модулю, поскольку, как было написано ранее, они записываются со знаком «минус», после к ним добавляться значение переменной х, которая представляет собой искомые объёмы производимых изделий каждого вида, обеспечивающие максимальную прибыль. Чтобы прослеживалось изменение переменных (х1,x2,x3 и т.д.) используем запись x=x+1. Заполняем пустой список «func» с помощью «.append». Так как у нас содержатся ещё и числовые объекты, которые нельзя совмещать со строками при интерации методом «join()», мы преобразовываем все значения у func1 в строку с помощью функций «map» и «str», где разделителем будет пробел При выводе благодаря совместному использованию функции «"+".join» и «map» со «str» также объединяем и преобразовываем значения, приводя их к единому типу, только разделителем между ними будет являться не проблем, а знак сложения – «+».

Последним действием является нахождение оптимального значения посредством использования симплекс метода. Для этого выбираем необходимое вводимое (в данном случае 1). Далее с помощью функции linprog() устанавливаем границы с помощью указанных ранее переменных и ищем минимальное значение с помощью метода «simplex». По итогу на выход получаем оптимальное значение целевой функции, где с помощью функции «o.fun» выводим только нужный по условию найденный показатель по модулю с помощью функции «abs», поскольку присутствует отрицательное число. На следующей строке программа выведет значение переменной при которых выводится оптимальный показатель при вычислении функции linprog() с использованием симплекс метода. Стоит отметить, чтобы все вычисления метода были в одной функции, используется def value

В конце программа рассчитает оптимальное значение целевой функции, дополнительно указав значение х. Ответ будет представлен в формате:

оптимальное значение целевой функции (симплекс): /значение/

При х: /значение/

* + 1. **Тесты**

После проверки кода было выполнено тестирование. Также данная задача была решена через MS Excel и онлайн калькулятор (<https://matworld.ru/calculator/simplex-method-online.php>).

Метод линейного программирования

Результат оптимального значения – 7999.999995514576, результат x’ов: x1 = 9.90286644e-02, x2 = 1.98019427e+01, x3 = 4.57054455e-10 (рис. 9).

Симплекс метод

Результатом оптимального значения является 8000, x1 = 0, x2 = 20, x3 = 0 (рис. 11).

По результатам решения данной задачи с помощью Excel: оптимальное значение – 8000, результат x’ов: x1 = 10, x2 = 0, x3 = 0.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1 — Решение ЗЛП через MS Excel

Стороннее решение

В качестве стороннего решения был использован онлайн калькулятор Симплекс метод онлайн. Заполнение количества ограничений и количества переменных:

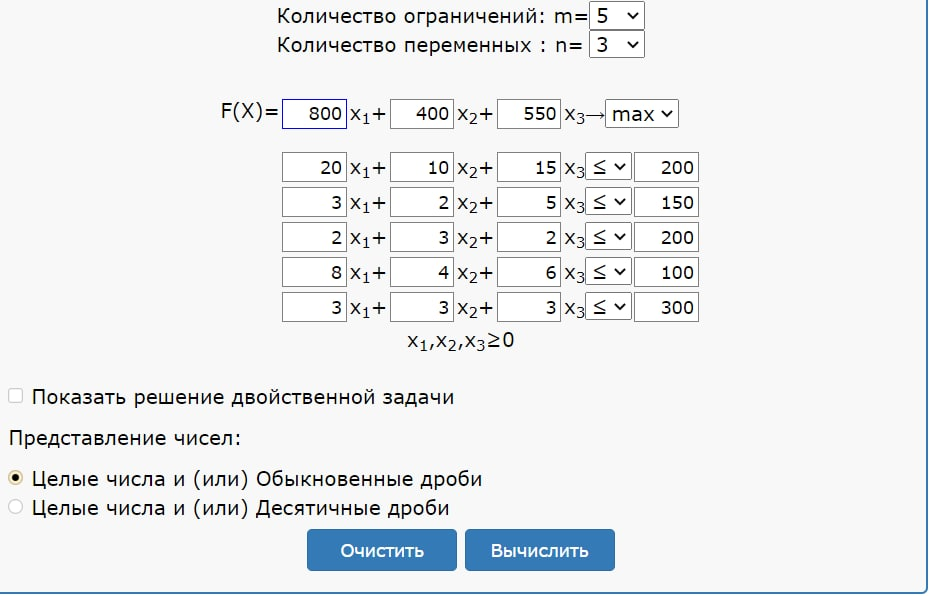


Рисунок 2 — Ввод данных для решения ЗЛП через онлайн калькулятора

Результат оптимального значения – 8000, x’ы: x1 = 10, x2 = 0, x3 = 0.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3 — Результат решения ЗЛП

* 1. **Итоги**

Мы смогли решить задачу при помощи методов «interior-point» и «simplex». Их сравнение можно наблюдать в Таблице – 4

Таблица 4 — “Сравнительная таблица”

|  |  |  |  |
| --- | --- | --- | --- |
| Критерий | interior-point | simplex | граф метод |
| Пример:  Количество возможных параметров | Не ограничено | Не ограничено | Не более 2 |
| Простота восприятия | не всем  понятно | легки в понимании | сложно определить точное значение,но позволяет увидеть как оно получилось |
| Время | быстрее достичь оптимальности | быстрее для небольших задач | затрачивает достаточно много времени |
| Тестирование | Число приближенное к тестированию | Число такое же как при тестировании в Excel и калькуляторе | Сложно сравнить |
| Точность | До 1000 | До 1000 | - |

Наиболее оптимальным алгоритмом мы считаем “simplex” метод, т.к. он предоставляет выходные данные, которые совпадают с результатами сторонних методов проверки

Предоставленный нами алгоритм позволил нам решить задачу в кратчайшие сроки, и предоставить заказчику наиболее точный ответ на поставленную задачу.

Варианты улучшения:

1. Предоставить пользователю возможность увидеть по какому материалу и продукции он набирает число в весовой матрице;

2. Возможность редакции данных;

3. Возможность выгрузки данных из excel формата;

4. Ввод цены без минуса, так как пользователь может не обратить на это внимания.

1. **Домашнее задание №2 “Решение задачи коммивояжёра”**
   1. **Условие задачи**

Наш заказчик коллекторское агентство, поставила перед нами задачу разработать алгоритм, который находил бы самый короткий маршрут между всеми адресами, в которых нужно навестить должников и вернутся обратно. Ниже представлена таблица с расстояниями между адресами должников.

Таблица 1 «Расстояния между адресами должников»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Пункт | “1” | “2” | “3” | “4” | “5” |
| “1” | 0 | 49 | 38 | 28 | 20 |
| “2” | 42 | 0 | 41 | 6 | 17 |
| “3” | 32 | 38 | 0 | 24 | 37 |
| “4” | 47 | 25 | 53 | 0 | 8 |
| “5” | 1 | 10 | 6 | 26 | 0 |

* 1. **Описание выполнения задачи**
     1. **Математическая модель**

Исходные данные:

* Весовая матрица размера n\*n, каждый элемент которой (aij) задает расстояние между городами. (6)

Целевая функция:

* (7)

Искомые переменные:

* Массив x = (x1;…; xj), каждый элемент которого (xj) показывает какие города необходимо посетить для того, чтобы пройти минимальное расстояние . (8)
* Оптимальное значение целевой функции.

Введение ограничений:

* (9)
  + 1. **Реализация**

В ходе решения нами были использованы два алгоритма: решение в Excel и метод ветвей и границ. Рассмотрим каждый из них.

Алгоритм задачи коммивояжёра в Excel пользователь последовательно вводит элементы весовой матрицы, которые обозначают расстояние между адресами, которые нужно посетить. (1) Далее алгоритм с помощью поиска решения перебирает все данные введенные пользователем, учитывая ограничения, и находит минимальное значение пути для каждого условного адреса. После алгоритм на основании вычислений записывает минимальный маршрут. Алгоритм выводит оптимальное значение целевой функции(2) и массив каждый элемент которого (*xj*) показывает какие города необходимо посетить для того, чтобы пройти минимальное расстояние . (3)

Алгоритм решения методом ветвей и границ: на вход подаются размеры весовой матрицы (1). Затем последовательно вводятся её элементы, обозначающие расстояния между городами. Алгоритм начинается с запоминания изначальной матрицы (1). После этого присваиваем главной диагонали матрицы бесконечность, тогда при решении задачи не будет математического смысла выбирать движение от n-ого города к n-му в качестве отрезка итогового маршрута. Упорядоченный перебор вариантов (рассмотрение лишь перспективных и отбрасывание бесперспективных), реализованный в алгоритме позволяет найти наиболее оптимальный маршрут и его затратность (длину, время, стоимость).

Алгоритм выводит оптимальное значение целевой функции(2) и массив каждый элемент которого (*xj*) показывает какие города необходимо посетить для того, чтобы пройти минимальное расстояние . (3)

* + 1. **Тесты**

Для тестирования будет использоваться 3 датасета с различными характеристиками:

Таблица 2 «Датасет 1»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Пункт | “1” | “2” | “3” | “4” | “5” |
| “1” | 0 | 49 | 38 | 28 | 20 |
| “2” | 42 | 0 | 41 | 6 | 17 |
| “3” | 32 | 38 | 0 | 24 | 37 |
| “4” | 47 | 25 | 53 | 0 | 8 |
| “5” | 1 | 10 | 6 | 26 | 0 |

Таблица 3 «Датасет 2»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Пункт | “1” | “2” | “3” | “4” | “5” |
| “1” | 0 | 6 | 36 | 7 | 35 |
| “2” | 54 | 0 | 41 | 10 | 25 |
| “3” | 27 | 49 | 0 | 35 | 14 |
| “4” | 4 | 22 | 39 | 0 | 53 |
| “5” | 10 | 18 | 53 | 14 | 0 |

Таблица 4 «Датасет 3»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Пункт | “1” | “2” | “3” | “4” | “5” |
| “1” | 0 | 10 | 54 | 35 | 52 |
| “2” | 45 | 0 | 15 | 43 | 31 |
| “3” | 31 | 14 | 0 | 22 | 24 |
| “4” | 42 | 37 | 44 | 0 | 50 |
| “5” | 11 | 12 | 32 | 26 | 0 |

Таблица 5 «Тест»

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Метод ветвей и границ** | **Задача коммивояжёра в Excel** | **Калькулятор**  **Задачи коммивояжёра** |
| Датасет1 | 101  Маршрут:2->4->5->3->1->2 | 85  Маршрут: 3->2->4->5->1->3 | 91  Маршрут: 1->3->2->4->5->1 |
| Датасет2 | 79  Маршрут: 4 -> 3-> 5-> 1 -> 2-> 4 | 44  Маршрут: 3->5->1->2->4->1 | 79  Маршрут: 3->5->4->1->2->3 |
| Датасет3 | 108  Маршрут:4-> 5-> 1 -> 2-> 3 ->4 | 100  Маршрут: 5->1->2->3->4->1 | 108  Маршрут: 1->2->3->4->5->1 |

Ссылка на калькулятор: https://math.semestr.ru/kom/index.php

* 1. **Итоги**

С помощью нашего алгоритма поставленная нам задача решена. Ответы получены: 85 решением в Excel с маршрутом 3->2->4->5->1->3 и 101 методом ветвей и границ с маршрутом 2->4->5->3->1->2. Нам необходим самый короткий маршрут поэтому мы выбираем вариант полученный решением в Excel.

Таблица 6 «Сравнительная таблица»

|  |  |  |
| --- | --- | --- |
|  | Решение в Excel | Метод ветвей и границ |
| Размерность матрицы | Неограниченно | От 2 до 1000 |
| Формат ответа | Целое число | Целое число |
| Время работы | Около минуты | ±0,000991 сек |
| Формат ввода данных | Ввод пользователем | Ввод пользователем, случайное заполнение, импорт из csv |
| Визуальное представление данных | - | - |

Подводя итог можно сказать, что метод решением в Excel подойдет для нас больше. Он работает немного дольше, но при этом позволяет работать с матрицами большой размерности, а так же выдает кратчайший маршрут.

Варианты улучшения:

1. Визуализация для метода ветвей и границ.
2. Перенос метода ветвей и границ на сайт либо создание приложения под него.
3. **Домашнее задание №3-4 “Теория игр”**
   1. **Условие задачи**

Грабитель намерен отправиться из города A в B в свое убежище, чтобы спрятаться от Полиции преследующей его и спрятать награбленное. Сев в машину, грабитель увидел недалеко от него преследователей. Грабитель допускает, что полиция могла раскрыть местоположение убежища и уже ждать его там. У грабителя есть две альтернативы: продолжать поездку до города B или свернуть в город C, являющейся другим местом где он может спрятаться, но при этом не сможет спрятать награбленное. Мы принимаем, что полиция достаточно разумна, чтобы определить возможности грабителя, поэтому перед ними те же две альтернативы. Оба противника должны выбрать город, чтобы поехать в него, не зная, какое решение примет каждый из них. Если в результате принятия решения оба окажутся в одном и том же городе, то можно однозначно считать, что Грабителя поймает Полиция. Если же Грабитель благополучно доберётся до города B, то он сможет укрыться.

Таблица 1 «Стратегии игроков»

|  |  |  |
| --- | --- | --- |
|  | Грабитель поедет в город B | Грабитель поедет город C |
| Полиция поедет в город B | Грабителя поймают (0) | Грабитель спрячется, но не спрячет награбленное (50) |
| Полиция поедет в город C | Грабитель спрячется сам и спрячет награбленное (100) | Грабителя поймают (0) |

* 1. **Описание выполнения задачи**
     1. **Математическая модель**

Исходные данные:

* – множество стратегий i-го игрока. (10)
* – функция выигрыша i-го игрока (11)
* – количество игроков (12)
* Матрица весовых коэффициентов(13)

Целевая функция:

* (14)

Искомые переменные:

* – функция выигрыша i-го игрока (15)
* – функция выигрыша i - го игрока зависит не только от стратегии U\_i этого игрока, но от стратегий всех остальных игроков. (16)

Введение ограничений:

* все, что выигрывает один игрок, проигрывает другой, и наоборот (16)
  + 1. **Реализация**

В нашем алгоритме для антагонистической игры на вход подаются размеры весовой матрицы (1). Затем последовательно вводятся её элементы, обозначающие выигрыши и проигрыши при каждой стратегии.

Алгоритм получает весовую матрицу (1) и находит гарантированный выигрыш, определяемый нижней ценой игры.

Алгоритм выводит оптимальную чистую стратегию для игрока A и B, цену оптимальной стратегии для игрока A, так же оптимальную чистую стратегию игрока B , цену игры игрока А и оптимальную смешанную стратегию игрока А.

* + 1. **Тесты**

Для тестирования будет использоваться 3 датасета с различными характеристиками:

Таблица 4 «Датасет 1»

|  |  |  |
| --- | --- | --- |
|  | Грабитель поедет в город B | Грабитель поедет город C |
| Полиция поедет в город B | Грабителя поймают (0) | Грабитель спрячется, но не спрячет награбленное (50) |
| Полиция поедет в город C | Грабитель спрячется сам и спрячет награбленное (100) | Грабителя поймают (0) |

Таблица 5 «Датасет 2»

|  |  |  |  |
| --- | --- | --- | --- |
|  | Грабитель поедет в город B | Грабитель поедет в город C | Грабитель решит остаться в городе A |
| Полиция поедет в город B | Грабителя поймают (0) | Грабитель спрячется, но не спрячет награбленное (50) | Грабитель спрячется, но не спрячет награбленное (50) |
| Полиция поедет в город C | Грабитель спрячется сам и спрячет награбленное (100) | Грабителя поймают (0) | Грабитель спрячется, но не спрячет награбленное (50) |
| Полиция останется в городе A | Грабитель спрячется сам и спрячет награбленное (100) | Грабитель спрячется, но не спрячет награбленное (50) | Грабителя поймают (0) |

Таблица 6 «Датасет 3»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Грабитель поедет в город B | Грабитель поедет в город B | Грабитель решит остаться в городе A | Грабитель решит спрятаться с лесах |
| Полиция поедет в город B | Грабителя поймают (0) | Грабитель спрячется, но не спрячет награбленное (50) | Грабитель спрячется, но не спрячет награбленное (50) | Грабитель спрячется, но не спрячет награбленное (50) |
| Полиция поедет в город C | Грабитель спрячется сам и спрячет награбленное (100) | Грабителя поймают (0) | Грабитель спрячется, но не спрячет награбленное (50) | Грабитель спрячется, но не спрячет награбленное (50) |
| Полиция останется в городе A | Грабитель спрячется сам и спрячет награбленное (100) | Грабитель спрячется, но не спрячет награбленное (50) | Грабителя поймают (0) | Грабитель спрячется, но не спрячет награбленное (50) |
| Полиция начнет искать грабителя в лесах | Грабитель спрячется сам и спрячет награбленное (100) | Грабитель спрячется, но не спрячет награбленное (50) | Грабитель спрячется, но не спрячет награбленное (50) | Грабителя поймают (0) |

Таблица 7 «Тест»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Решение в Excel антагонистической игры** | **Решение в Excel биматричной игры** | **Решение в Excel игры в условиях риска** | **Решение в Excel игры в условиях неопределенности** |
| Датасет1 | Оптимальная чистая стратегия грабителя = 100  оптимальная чистая стратегия полиции= 50  X2=0.01  X1=0.02  min F =0.03  цена игры для грабителя 33.33  Вероятность сбежать грабителю при стратегии:сойти в городе А=33%  Вероятность грабителю сбежать при стратегии: сойти в городе В=67 | Оптимальная чистая стратегия грабителя = 0  оптимальная чистая стратегия полиции = 50  цена игры для грабителя равна 33.33  Вероятность грабителя сбежать при стратегии: сойти в городе А=33%  Вероятность грабителя скрыться при стратегии: сойти в городе В=67% | Вывод: в город С выгоднее всего ехать грабителю  Цена игры: 13,78 |  |
| Датасет2 | Оптимальная чистая стратегия грабителя = 100  оптимальная чистая стратегия полиции= 50  X2=0.01  X1=0.02  min F =0.03  цена игры для грабителя 33.33  Вероятность сбежать грабителю при стратегии:сойти в городе А=33%  Вероятность грабителю сбежать при стратегии: сойти в городе В=67 | Оптимальная чистая стратегия грабителя = 0  оптимальная чистая стратегия полиции = 50  цена игры для грабителя равна 33.33  Вероятность грабителя сбежать при стратегии: сойти в городе А=33%  Вероятность грабителя скрыться при стратегии: сойти в городе В=67% | Вывод: в город С выгоднее всего ехать грабителю  Цена игры: 5,42 |  |
| Датасет3 |  |  |  |  |

Калькулятор для антагонистической, который использовался для теста: https://math.semestr.ru/games/antogonist.php

* 1. **Итоги**

С помощью нашего алгоритма поставленная нам задача решена. Ответ полученный нами Оптимальная чистая стратегия игрока Грабитель = 0 , оптимальная чистая стратегия игрока Полиция = 50, X2=0.01 , X1=0.02 , min F =0.03 , Цена игры для грабителя равна 33.33 , вероятность спастись грабителю при стратегии: поехать в город А=33%, вероятность спастись грабителю при стратегии: грабителю поехать в город В=67% .

|  |  |
| --- | --- |
|  | Решение в Excel |
| Кол-во обрабатываемых строк в матрице | Не ограничено |
| Точность ответа | 5 знаков после запятой |
| Удобность работы | - |
| Визуальное представление данных | - |

Таблица 8 «Сравнительная таблица»

Подводя итог можно сказать, что все варианты выдали одинаковый ответ, что доказывает верность ответа.

Варианты улучшения:

1. Визуализация.
2. Заполнение данных не вручную, а через импортирование csv файла.
3. Сделать приложение основанное на нашем алгоритме.