**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 № 6**

«Решение задач паретооптимальным методом.»

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](#_Toc103633706)

[1. Постановка задачи (физическая модель) 4](#_Toc103633707)

[2. Математическая модель 5](#_Toc103633708)

[2.1. Поиск Пареттооптимального решения 5](#_Toc103633709)

[2.2. Линейная свертка критериев 5](#_Toc103633710)

[2.3. Метод идеальной точки 6](#_Toc103633711)

[3. Алгоритмы решения задачи 7](#_Toc103633712)

[3.1. Описание входных данных. 7](#_Toc103633713)

[3.2. Описание алгоритма решения 8](#_Toc103633714)

[3.3. Описание выходных данных 9](#_Toc103633715)

[4. Вариант использования 11](#_Toc103633716)

[5. Архитектура решения 13](#_Toc103633717)

[5.1 Функции считывания информации 13](#_Toc103633718)

[5.2 Функции обработки информации 20](#_Toc103633719)

[5.3 Функции вывода информации 27](#_Toc103633720)

[6. Тестирование 28](#_Toc103633721)

[6.1. Тестирование Датасета №1: 28](#_Toc103633722)

[6.1.1 Метод Python: 28](#_Toc103633723)

[6.1.2 Метод Excel: 32](#_Toc103633724)

[6.2. Тестирование Датасета №2: 33](#_Toc103633725)

[6.2.1 Метод Python: 33](#_Toc103633726)

[6.1.2 Метод Excel: 37](#_Toc103633727)

[6.3. Тестирование Датасета №3: 38](#_Toc103633728)

[6.3.1 Метод Python: 38](#_Toc103633729)

[6.3.2 Метод Excel: 42](#_Toc103633730)

[6.4. Тестирование Датасета №4: 43](#_Toc103633731)

[6.4.1 Метод Python: 43](#_Toc103633732)

[6.4.2 Метод Excel: 47](#_Toc103633733)

[6.5. Тестирование Датасета №5: 48](#_Toc103633734)

[6.5.1 Метод Python: 48](#_Toc103633735)

[6.5.2 Метод Excel: 52](#_Toc103633736)

[7. Заключение 53](#_Toc103633737)

# Постановка задачи (физическая модель)

Сеть кофеен ООО «ВКУСНОЕ КОФЕ» столкнулось с экономическими трудностями после введения множества санкций на Россию. Одна из главных трудностей – это логистическая проблема.

Дело в том, что зерна поставляли из Испании, а одноразовые стаканчики и крышки из Нидерландов, но из-за закрытия воздушного пространства и наземных границ ЕС, поставлять эти материалы стало крайне затруднительно и не выгодно.

Наша задача найти оптимальный способ доставки этих ресурсов с двумя важными условиями, которые критичны для заказчика: срок доставки не более 7 дней (как это было до введенных санкций) и сохранить закупочную стоимость на уровне 5000 $ за партию. Остальные параметры, такие как страна производства, производитель, качество и т.д. не критичны и могут быть изменены.

Для предоставления заказчику качественного решения мы провели исследование рынка и составили сводную таблицу по поставщикам:

Таблица 1. Результаты исследования

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Качество | Страна | Производительность в неделю | Стоимость в $ | Способ доставки | Срок доставки |
| Поставщик №1 | Низкое | Индия | 8 партий | 4 000 | Корабль | 7 дней |
| Поставщик №2 | Среднее | Китай | 10 партий | 3 500 | Поезд | 4 дня |
| Поставщик №3 | Высокое | Бразилия | 6 партий | 4 900 | Корабль | 6 дней |
| Поставщик №4 | Среднее | Аргентина | 6 партий | 4 600 | Корабль | 5 дней |
| Поставщик №5 | Высокое | Турция | 4 партии | 5 000 | Самолет | 2 дня |

# Математическая модель

## 2.1. Поиск Пареттооптимального решения

Найти 𝑥, доставляющий экстремум критериям:

…

при выполнении ограничений:

Соответственно, если достигается минимум — точка экстремума называется точкой минимума, а если максимум — точкой максимума.

Если 𝑓(𝑥) → max, то критерий называется позитивным, в противном случае – негативным.

Решением задачи является паретооптимальное множество 𝑥: {𝑥i, …, 𝑥n}, доставляющее необходимый экстремум критериям W

## 2.2. Линейная свертка критериев

Суть линейной сверстки заключается в том, чтобы перейти от нескольких критериев к одному путем введения соответствующих коэффициентов к каждому такому критерию, за счет которых возможно будет «свернуть» все в один критерий.

В общем виде линейная сверстка будет выглядеть так:

Где ai – коэффициент значимости конкретного критерия

При условии, что все критерии fi(x) стремятся к максимуму

Если же некоторые критерии стремятся к минимуму, например, если f1(x) → max, а f2(x) → min, то сверстка видоизменяется и приобретает следующий вид:

## 2.3. Метод идеальной точки

Алгоритмом решения задачи методом идеальной точки будет являться следующий ход действий:

Выбирается точка, которая лучше остальных по всем критериям, причем она должна быть относительно приближена к ним. Затем из каждой точки строится расстояние до идеальной, соответственно поиск будет сводится к нахождению гипотенузы через построения проекции для прямоугольного треугольника. Такой алгоритм используется для поиска сбалансированного решения, наиболее близкого к идеальному.

Математически интерпретироваться такая задача будет так:

Пусть имеется n-критериальная задача:

Выберем «идеальную точку» - наилучшие значения каждого критерия:

Далее необходимо будет найти наименьшее расстояние до такой точки

Для этого ведем глобальный критерий 𝑊, который минимизирует нормированное расстояние до идеальной точки:

Решая задачу, находим такое 𝑊, при котором все функции 𝑓𝑖(𝑥) ближе всего к своим идеальным значениям

Преимущество метода идеальной точки заключается в том, что присутствие эксперта, который анализирует какие-либо параметры, не требуется, поэтому данный способ может быть полностью автоматизирован.

# 3. Алгоритмы решения задачи

Алгоритмы решения реализованы с помощью программного кода в Python.

### 3.1. Описание входных данных.

Формат входных данных определяется тем, что программа принимает только CSV файл, который содержит в себе массивы данных.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 1. Пример входных данных в CSV файле



Рисунок 2. Пример заполнения пути к файлу

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 3. Пример импортированных данных в программу

### 3.2. Описание алгоритма решения

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

Шаг 1: создаем список критериев, по которым осуществляется оптимизация.

Шаг 2: заполнение массива значений идеальной точки.

Шаг 3: рассчитываем дисперсию по выборке.

Шаг 4: нахождение набора точек поверхности Парето.

Шаг 5: построение точечной диаграммы для двухкритериальной оптимизации по Парето.

Шаг 6: построение лепестковой диаграммы для многокритериальной задачи.

Шаг 7: линейная свертка критериев, при помощи возращения множителя, формирующий элемент линейной свертки.

Шаг 8: построение диаграммы линейной свертки критериев.

Шаг 9: линейная свертка для более двух критериев.

Шаг 10: расчет оптимального значения методом идеальной точки.

Шаг 11: построение лепестковой диаграммы.

### 3.3. Описание выходных данных

В конце программа выдает оптимальные значения по Паретто, по методу идеальной точки и линейной свертки.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 4. Пример выходных данных

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 5. Поиск Пареттооптимального решения

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 6. Линейная сверстка критериев и метод идеальной точки

# 4. Вариант использования

Существует единственный вариант использования кода – это загрузка файла. Данный вариант использования включает в себя ввод данных с помощью файла csv. Для того, чтобы ввести путь к файлу необходимо запустить программу.

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



Рисунок 7. Выбор местонахождения csv файла

2. Определите количество критериев, по которым необходимо осуществить оптимизацию (более одного).

3. Для каждого оптимизационного критерия последовательно введите его название (им является имя столбца импортируемой таблицы) и направление оптимизации (укажите “Макс”, если следует оптимизировать критерий к максимуму, и “Мин”, если к минимуму.

4. Далее для каждого критерия будет предложено ввести весовой коэффициент (важность). Он является необязательным, поэтому его допустимо пропустить (просто нажмите enter, не заполняя ячейку).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 8. Пример

5. После программа предложит ввести идеальную точку (для метода оптимизации по идеальной точке). Это также является необязательным, поэтому укажите “нет”, если вводить ее не будете, и “да”, если данные по ней имеются.

6. При вводе “нет” программа выведет результат оптимизации с имеющимися данными без идеальной точки, при “да” – пользователю следует последовательно ввести ее весовые коэффициенты для каждого критерия.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 9. Пример

# 5. Архитектура решения

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

## 5.1 Функции считывания информации

**Функция import\_data:**

**Функция import\_csv:**

**Входные данные:**

* Нет входных данных.

**Выходные данные:**

* A – массив данных в формате dataframe.

**Переменные, затрагиваемые в ходе работы:**

* csv – пустой список.
* A – массив данных в формате dataframe.
* x – переменная, затрагиваемая в ходе работы.
* columns – список значений столбца.
* index – список значений индекса.

**Функция input\_num\_criteria:**

**Входные данные:**

* Нет входных данных.

**Выходные данные:**

* Нет выходных данных.

**Переменные, затрагиваемые в ходе работы:**

* number – количество критериев, которых нужно оптимизировать.
* А - массив данных в формате dataframe.

**Функция input\_name\_criteria:**

**Входные данные:**

* Нет входных данных.

**Выходные данные:**

* Нет выходных данных.

**Переменные, затрагиваемые в ходе работы:**

* name – название критериев.
* A – массив данных в формате dataframe.

**Функция input\_direction\_of\_optimiz:**

**Входные данные:**

* Нет входных данных.

**Выходные данные:**

* Нет выходных данных.

**Переменные, затрагиваемые в ходе работы:**

* A – массив данных в формате dataframe.
* direction – значение направления.

**Функция input\_weight\_coeff:**

**Входные данные:**

* Нет входных данных.

**Выходные данные:**

* Нет выходных данных.

**Переменные, затрагиваемые в ходе работы:**

* weight – значение важности.
* A – массив данных в формате dataframe.

**Функция optimiz\_criteria:**

**Входные данные:**

* Нет входных данных.

**Выходные данные:**

* Нет выходных данных.

**Переменные, затрагиваемые в ходе работы:**

* weight – значение важности.
* list\_optimiz\_criteria – список критериев.
* A – массив данных в формате dataframe.
* number – количество критериев, которых нужно оптимизировать.

**Функция input\_ideal\_point:**

**Функция input\_value:**

**Входные данные:**

* Нет входных данных.

**Выходные данные:**

* Нет выходных данных.

**Переменные, затрагиваемые в ходе работы:**

* value – значения по столбцу массивов данных.
* data – переменная, используемая в функции.
* i - переменная, используемая в функции.
* inpt – присутствие идеальной точки.
* ideal\_point – список значений идеальной точки.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок . Часть программного кода, отвечающего за функцию считывания информации

Изображение выглядит как текст

Автоматически созданное описание

Рисунок . Часть программного кода, отвечающего за функцию считывания информации

Изображение выглядит как текст

Автоматически созданное описание

Рисунок . Часть программного кода, отвечающего за функцию считывания информации

Изображение выглядит как текст

Автоматически созданное описание

Рисунок . Часть программного кода, отвечающего за функцию считывания информации

## 5.2 Функции обработки информации

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

Поскольку 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 массива данных, а также список с названиями строк.

Функция graph\_pareto строит точечную диаграмму для двухкритериальной оптимизации по Парето и лепестковую диаграмму для многокритериальной задачи, более того, она включает функцию opt, pareto\_surface, graph\_for\_two\_criteria. В функции opt происходит проверка точки на вхождение в набор оптимальных значений по Парето. В pareto\_surface находятся наборы точек поверхности Парето, а в функции graph\_for\_two\_criteria строятся точечная диаграмма для двухкритериальной оптимизации по Парето.

В функция data\_normalization происходит нормирование исходных данных с помощью цикла for in i.

Функция linear\_convolution реализует оптимальное значение и набор точек поверхности Парето с помощью линейной свертки, а также строит диаграмму линейной свертки критериев. Также она включает в себя следующие функции: func, которая возвращает множитель, формирующий элемент линейной свертки; convolution\_matrix, которая возвращает таблицу линейной свертки для двух оптимизационных критериев; output, которая выводит оптимальные значения на экран (для оптимизации двух критериев); graph\_linear\_convolution, которая строит диаграмму линейной свертки критериев (для оптимизации двух критериев); convolution\_more\_2\_cr, которая возвращает таблицу линейной свертки и оптимальное значение для более двух оптимизационных критериев.

В функции meth\_ideal\_point рассчитывается оптимальное значение методом идеальной точки и формирует лепестковую диаграмму указанием Паретооптимального множества решений.

**Функция graph\_pareto:**

**Входные данные:**

* Нет входных данных.

**Выходные данные:**

* data – переменная, затрагиваемая в функции.
* criteria – значение критерия.
* Pareto\_surface – точки поверхности Парето.

**Переменные, затрагиваемые в ходе работы:**

* data - переменная, затрагиваемая в функции.
* criteria – значение критерия.
* i – переменная, затрагиваемая в функции.
* j - переменная, затрагиваемая в функции.
* Pareto\_surface – точки поверхности Парето.
* optimal\_point\_x – список набора точек по х.
* optimal\_point\_y – список набора точек по у.
* x – значение критерия по х.
* y - значение критерия по у.

**Функция linear\_convolution:**

**Входные данные:**

* Нет входных данных.

**Выходные данные:**

* Нет выходных данных.

**Переменные, затрагиваемые в ходе работы:**

* data - переменная, принимающая значения в функции.
* criteria – значение критерия.
* i\_data – переменная, принимающая значения в функции.
* number – количество критериев.
* cr – значение критерия.
* name\_columns – название столбцов.
* row – список значений.
* lin\_conv - список значений.

**Функция meth\_ideal\_point:**

**Входные данные:**

* Нет входных данных.

**Выходные данные:**

* Нет выходных данных.

**Переменные, затрагиваемые в ходе работы:**

* data - переменная, принимающая значения в функции.
* criteria – значение критерия.
* i\_data – переменная, принимающая значения в функции.
* ideal\_point – значение идеальной точки.
* row – список значений.
* norm\_meth\_ideal\_point – список значений.
* distance – список значений.

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок . Часть кода, отвечающая за функцию

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок . Часть кода, отвечающая за функцию

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок . Часть кода, отвечающая за функцию

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок . Часть кода, отвечающая за функцию

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок . Часть кода, отвечающая за функцию

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок . Часть кода, отвечающая за функцию

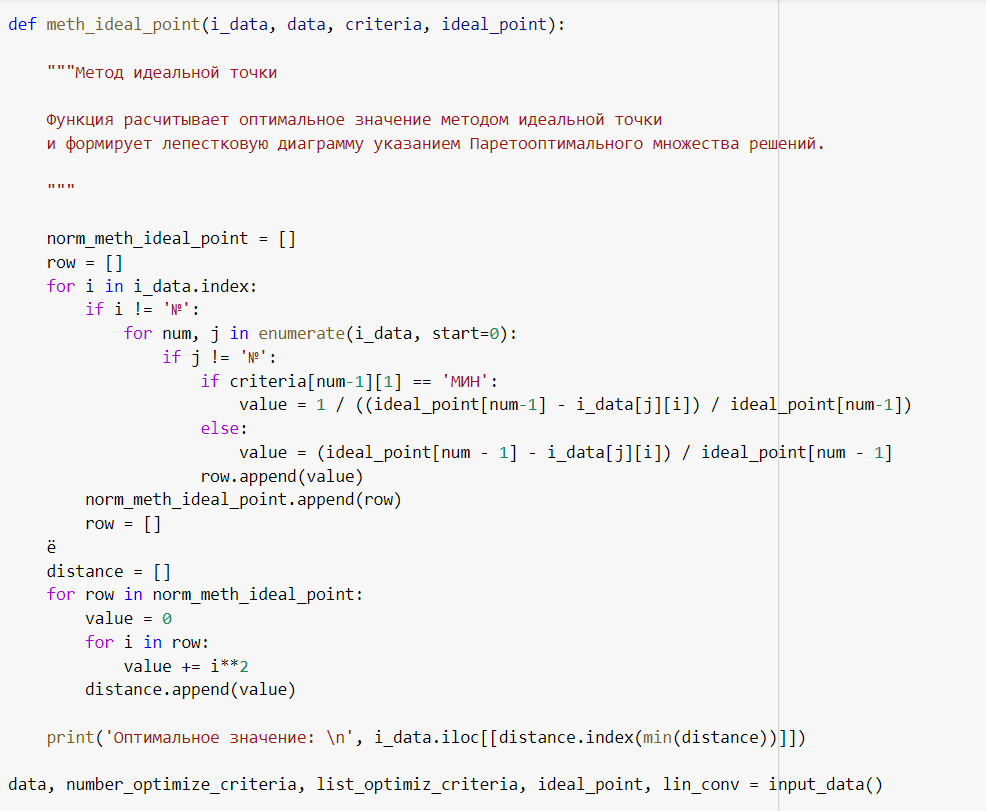
****

Рисунок . Часть кода, отвечающая за функцию

## 5.3 Функции вывода информации

Метод вывода информации

Что делает: осуществляет вывод необходимой информации

В данном методе осуществляется непосредственно вызов функций с помощью метода print ().**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок . Часть кода, отвечающая за функцию

# 6. Тестирование

Проведём тестирование нашей программы и сравним полученные показатели,

чтобы сделать вывод о предпочтительном варианте использования нашей программы

или Excel под условия заказчика.

## 6.1. Тестирование Датасета №1:

### 6.1.1 Метод Python:

Импортируем датасет в Python, выбираем количество оптимизируемых критериев, названия критериев, направление и важность их оптимизации. Уточняем наличие идеальной точки, вводим идеальные параметры.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок . импорт

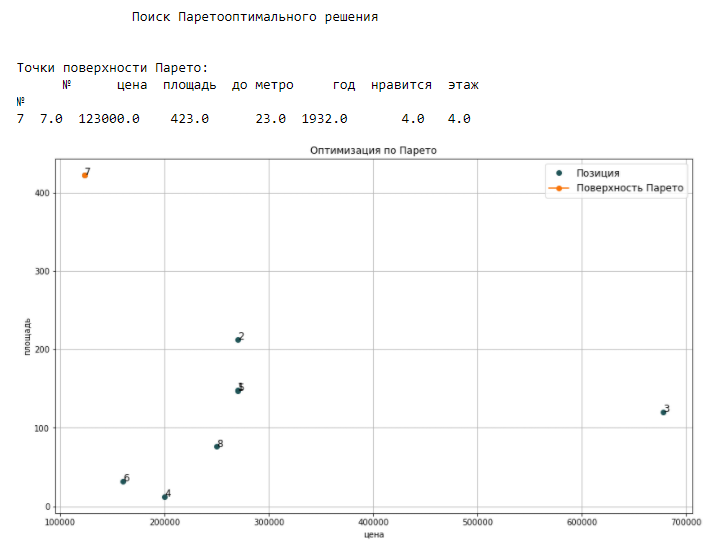


Рисунок . паретооптимальное решение

Изображение выглядит как текст

Автоматически созданное описание

Рисунок . нормированные данные

Изображение выглядит как текст, стол

Автоматически созданное описание

Рисунок . линейная свёрстка

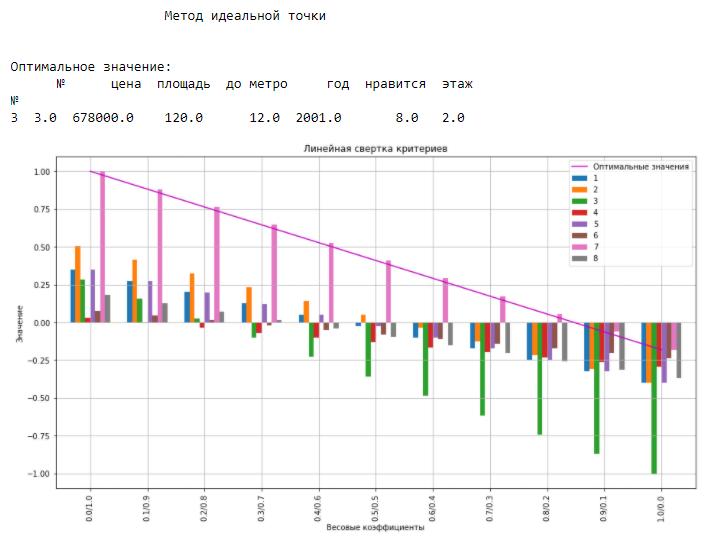


Рисунок . идеальная точка

### 6.1.2 Метод Excel:

Вводим данные датасета в необходимые поля и получаем результат:

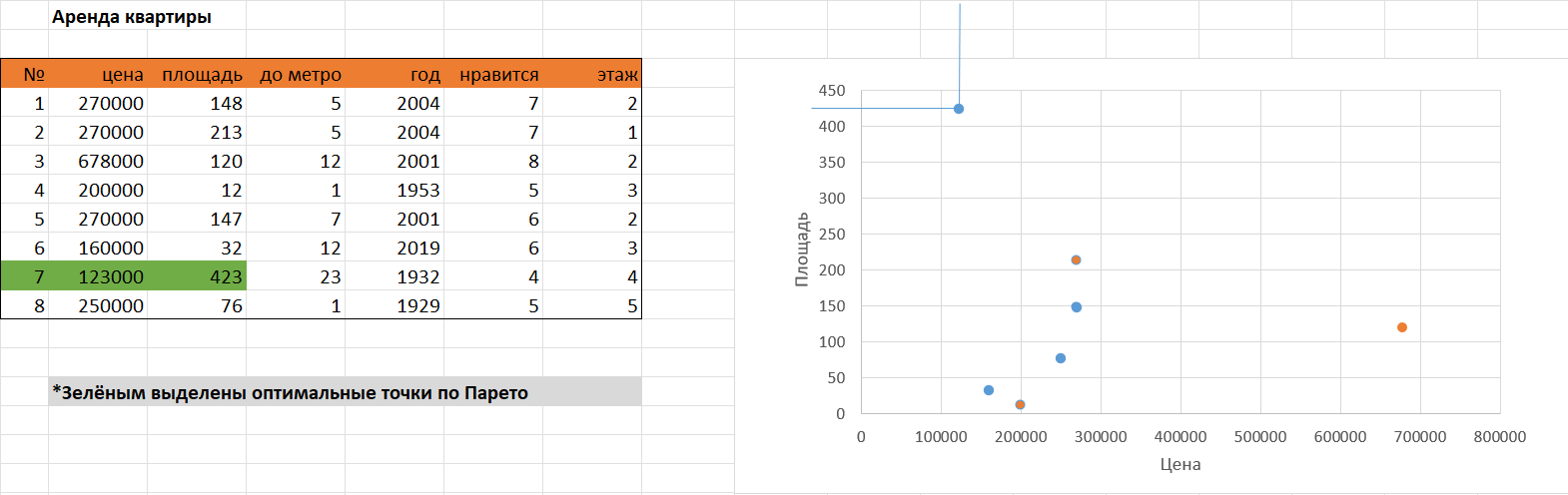


Рисунок . паретто

Изображение выглядит как стол

Автоматически созданное описание

Рисунок . линейная свёрстка

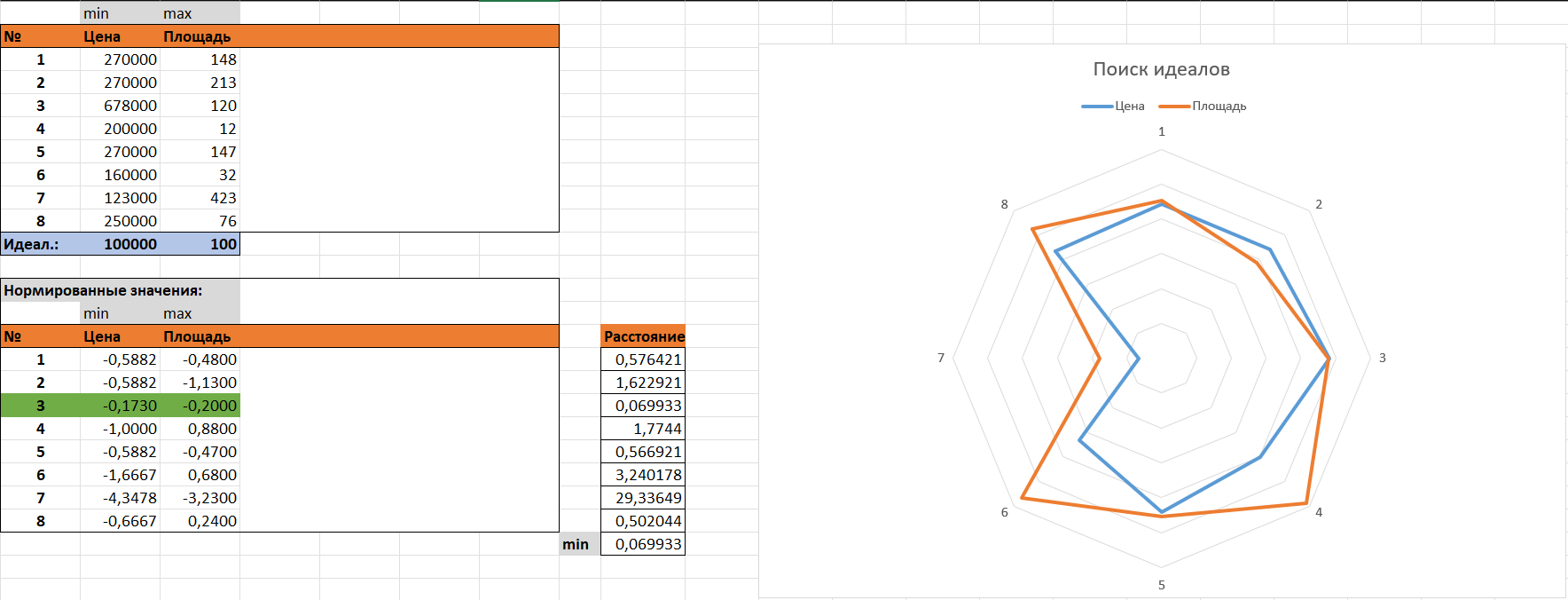


Рисунок . Идеальная точка

## 6.2. Тестирование Датасета №2:

### 6.2.1 Метод Python:

Импортируем датасет в Python, выбираем количество оптимизируемых критериев, названия критериев, направление и важность их оптимизации. Уточняем наличие идеальной точки, вводим идеальные параметры.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок . Импорт

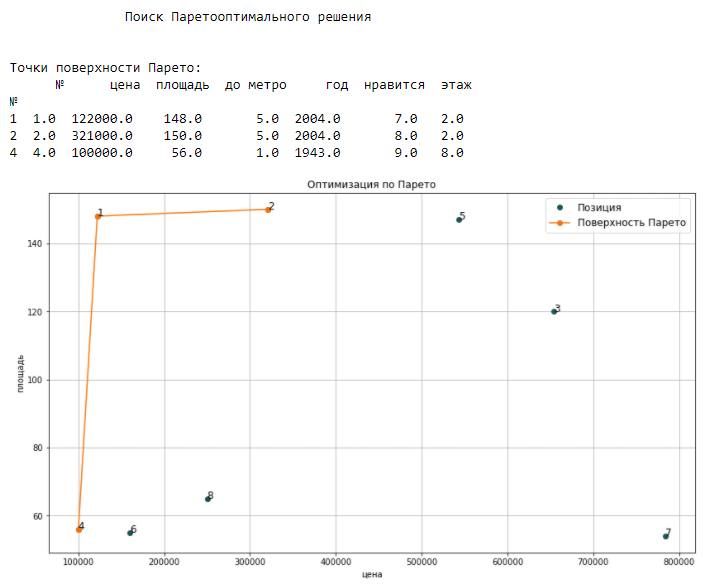


Рисунок . паретто

Изображение выглядит как текст

Автоматически созданное описание

Рисунок . нормированные данные

Изображение выглядит как стол

Автоматически созданное описание

Рисунок . линейная свёрстка

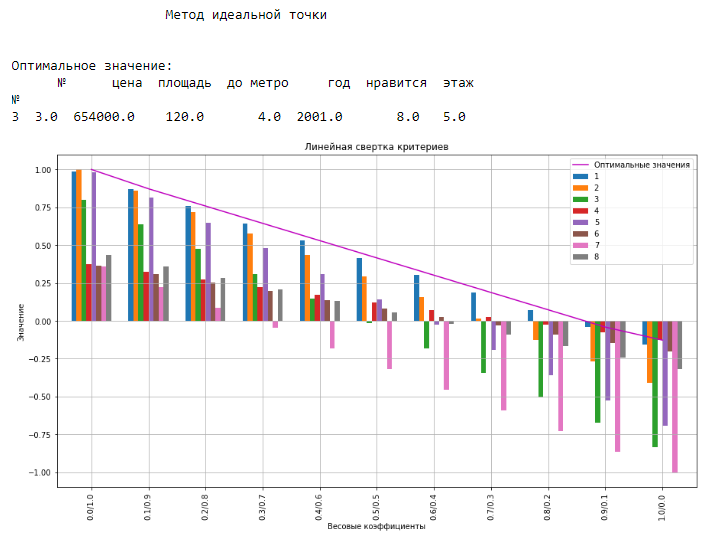


Рисунок . идеальная точка

### 6.1.2 Метод Excel:

Вводим данные датасета в необходимые поля и получаем результат:

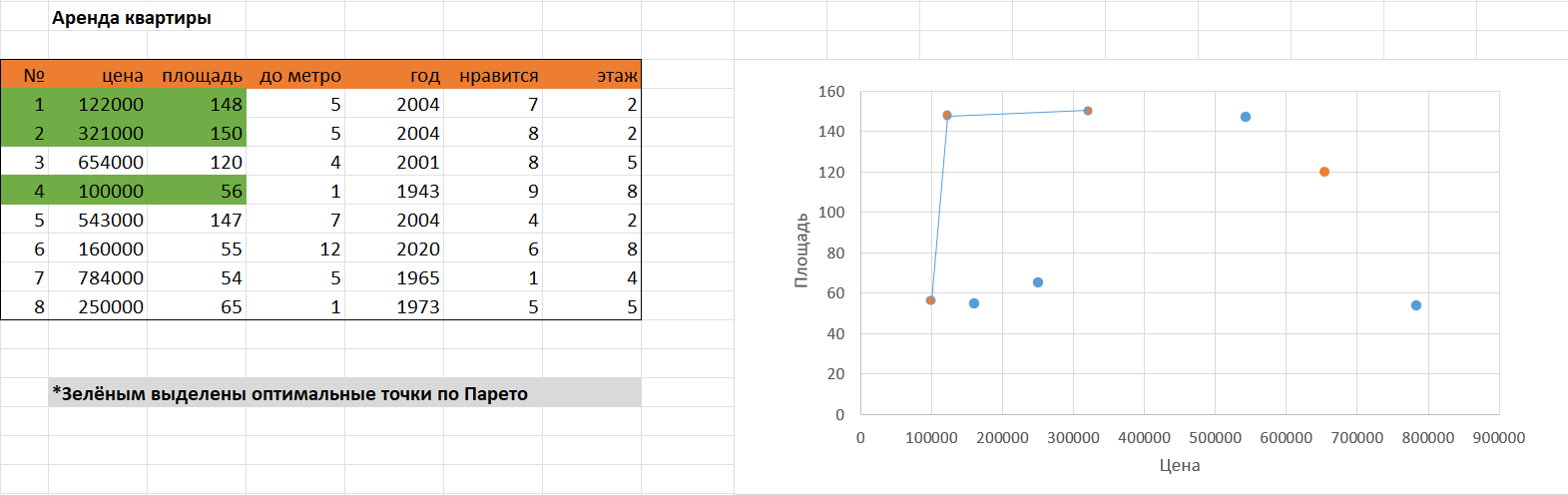


Рисунок . Паретооптимальное

Изображение выглядит как стол

Автоматически созданное описание

Рисунок . линейная свёрстка

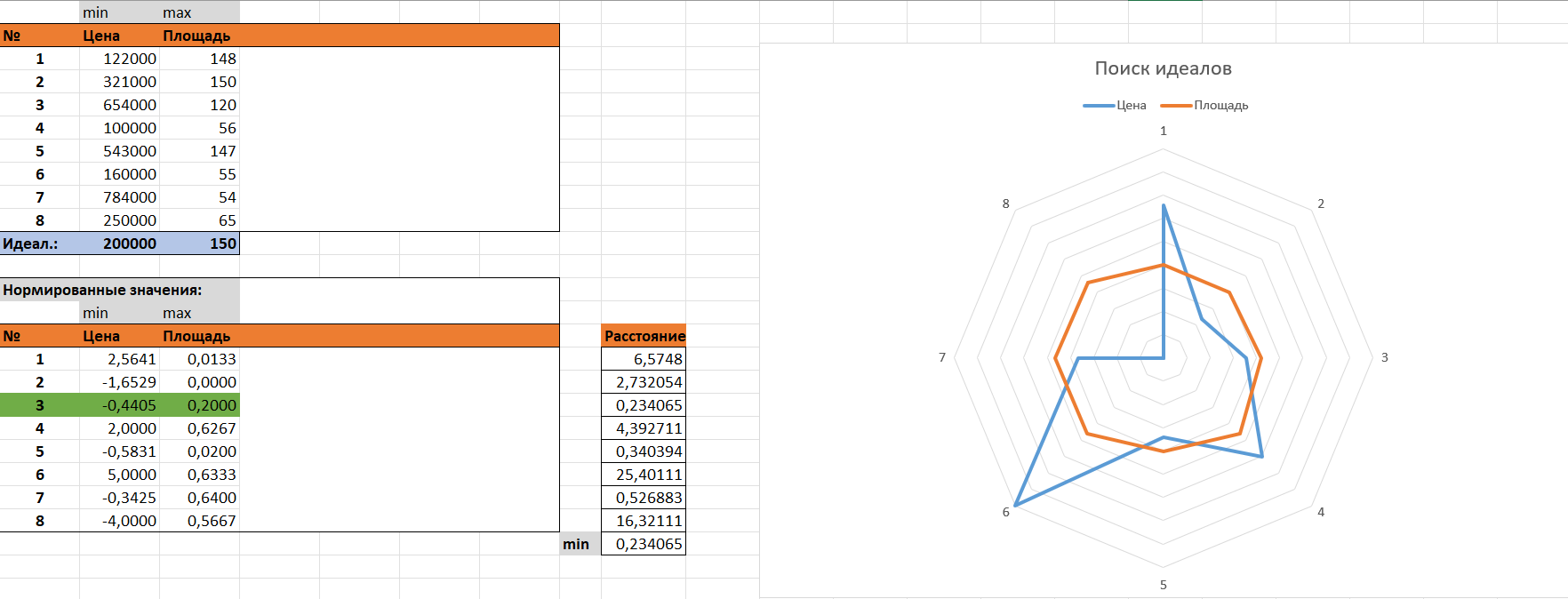


Рисунок . идеальная точка

## 6.3. Тестирование Датасета №3:

### 6.3.1 Метод Python:

Импортируем датасет в Python, выбираем количество оптимизируемых критериев, названия критериев, направление и важность их оптимизации. Уточняем наличие идеальной точки, вводим идеальные параметры.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок . импорт

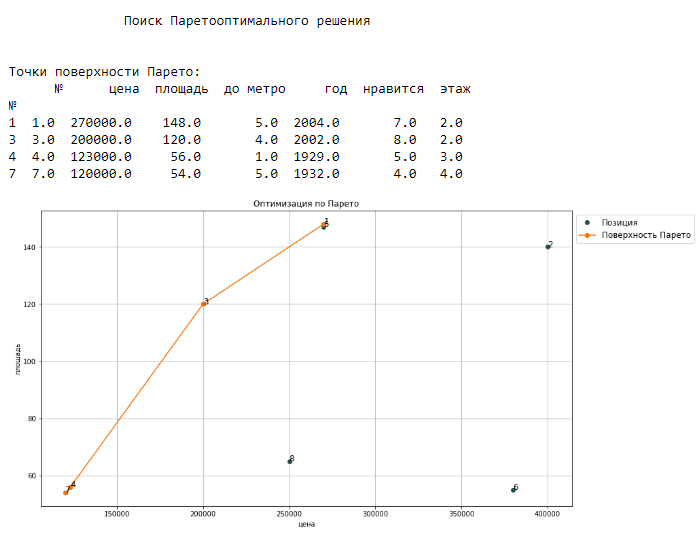


Рисунок . Паретооптимальное

Изображение выглядит как текст

Автоматически созданное описание

Рисунок . нормированные данные

Изображение выглядит как стол

Автоматически созданное описание

Рисунок . линейная свёрстка

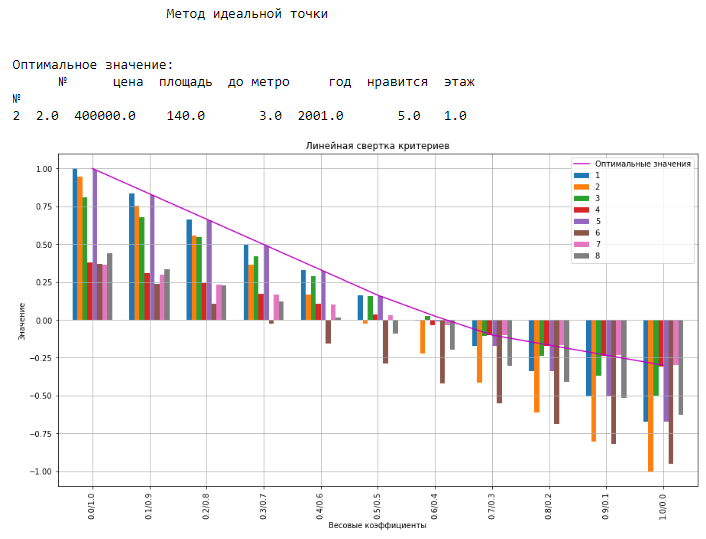


Рисунок . идеальная точка

### 6.3.2 Метод Excel:

Вводим данные датасета в необходимые поля и получаем результат:

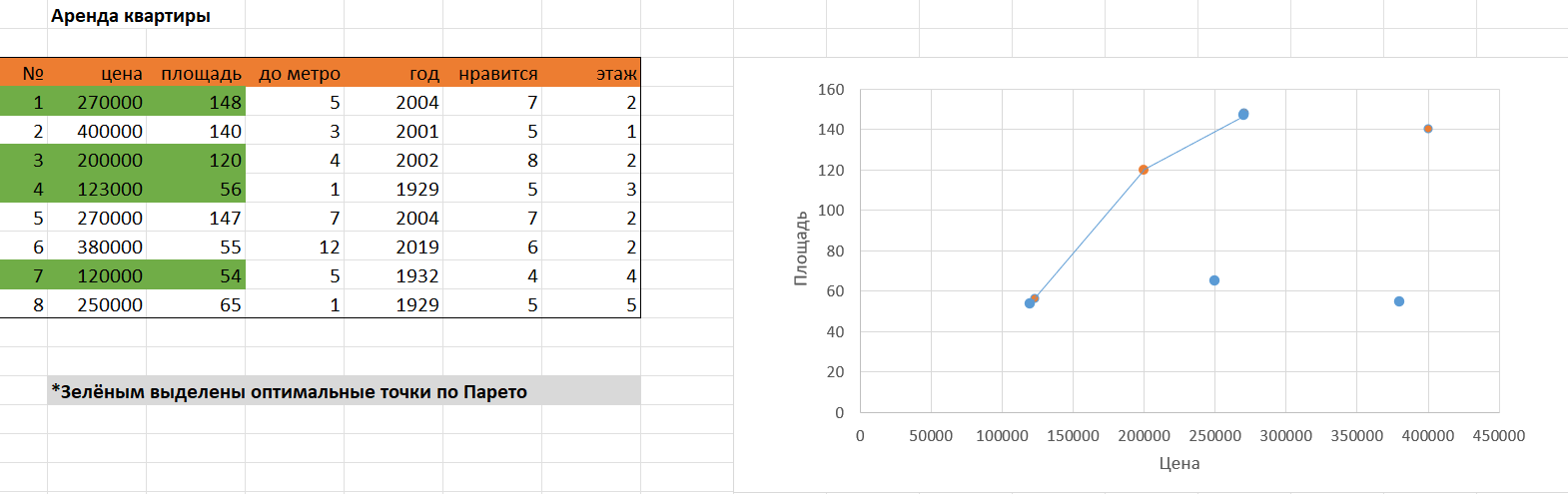


Рисунок . Паретооптимальное

Изображение выглядит как стол

Автоматически созданное описание

Рисунок . линейная свёрстка

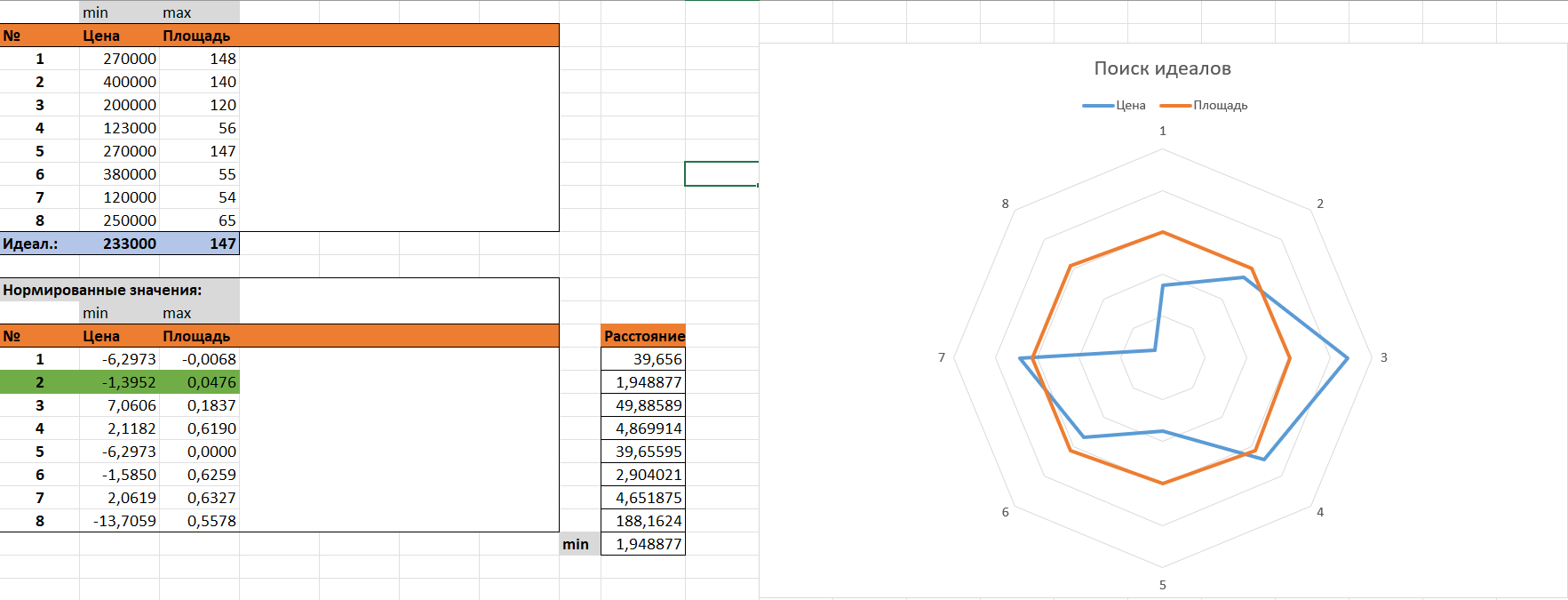


Рисунок . идеальная точка

## 6.4. Тестирование Датасета №4:

### 6.4.1 Метод Python:

Импортируем датасет в Python, выбираем количество оптимизируемых критериев, названия критериев, направление и важность их оптимизации. Уточняем наличие идеальной точки, вводим идеальные параметры.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок . Импорт

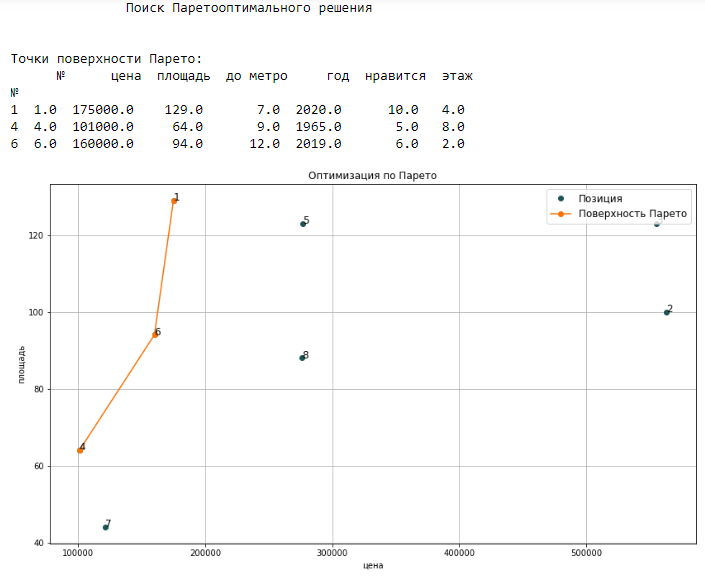


Рисунок . Паретооптимальное

Изображение выглядит как текст

Автоматически созданное описание

Рисунок . нормированные данные

Изображение выглядит как стол

Автоматически созданное описание

Рисунок . линейная свёрстка критериев

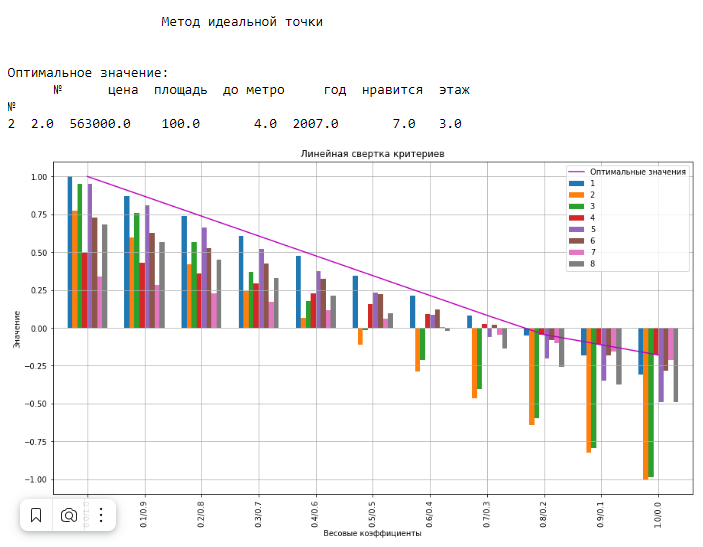


Рисунок . идеальная точка

### 6.4.2 Метод Excel:

Вводим данные датасета в необходимые поля и получаем результат:

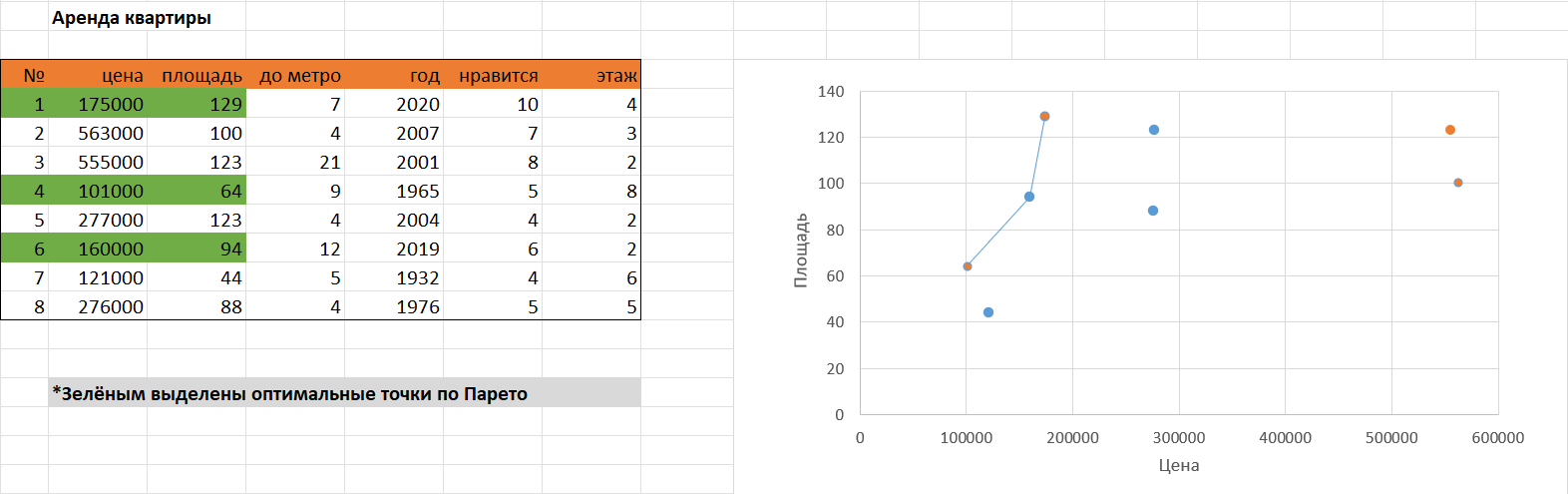


Рисунок . Паретооптимальное

Изображение выглядит как стол

Автоматически созданное описание

Рисунок . линейная свёрстка

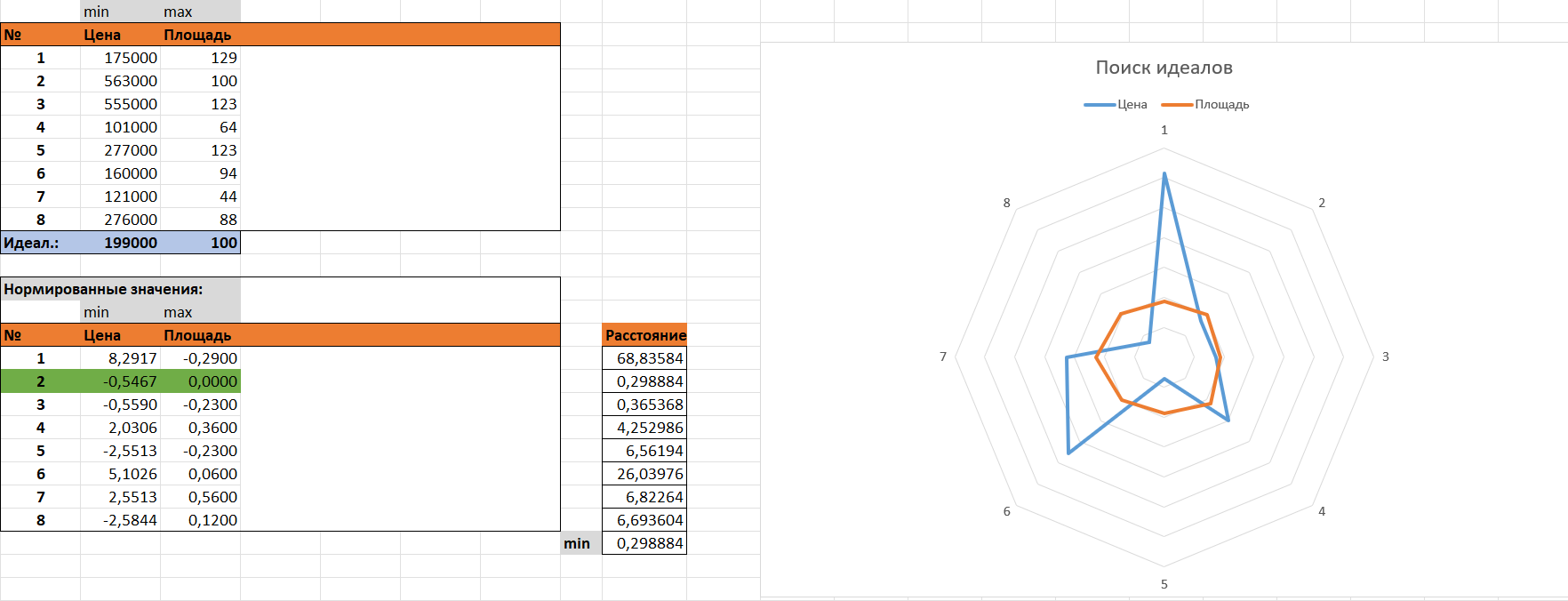


Рисунок . идеальная точка

## 6.5. Тестирование Датасета №5:

### 6.5.1 Метод Python:

Импортируем датасет в Python, выбираем количество оптимизируемых критериев, названия критериев, направление и важность их оптимизации. Уточняем наличие идеальной точки, вводим идеальные параметры.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок . Импорт

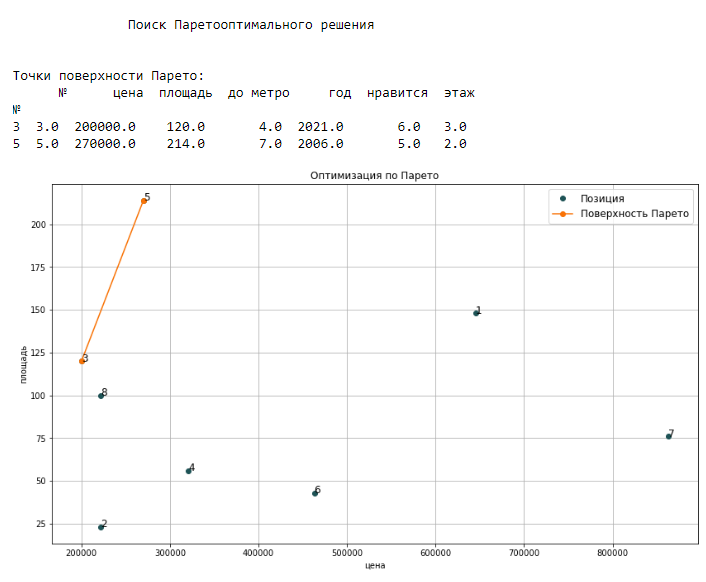


Рисунок . Паретооптимальное

Изображение выглядит как текст, стол

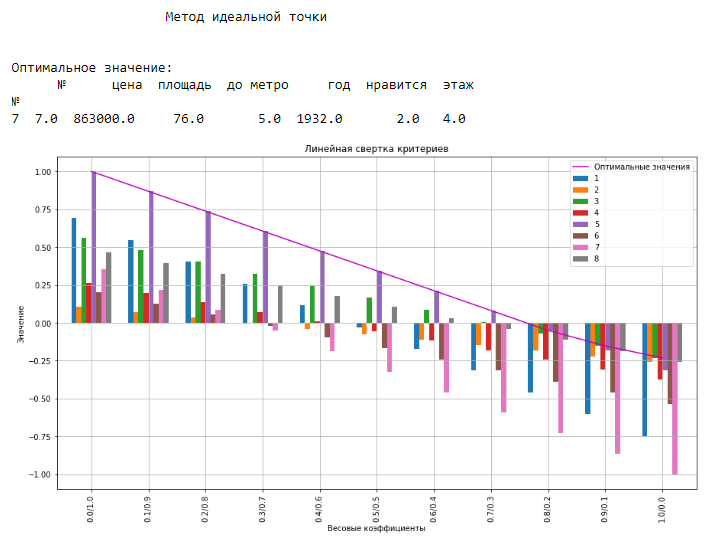
Автоматически созданное описание

Рисунок , нормированные данные

Изображение выглядит как стол

Автоматически созданное описание

Рисунок . линейная свёрстка

идеальная точка

### 6.5.2 Метод Excel:

Вводим данные датасета в необходимые поля и получаем результат:

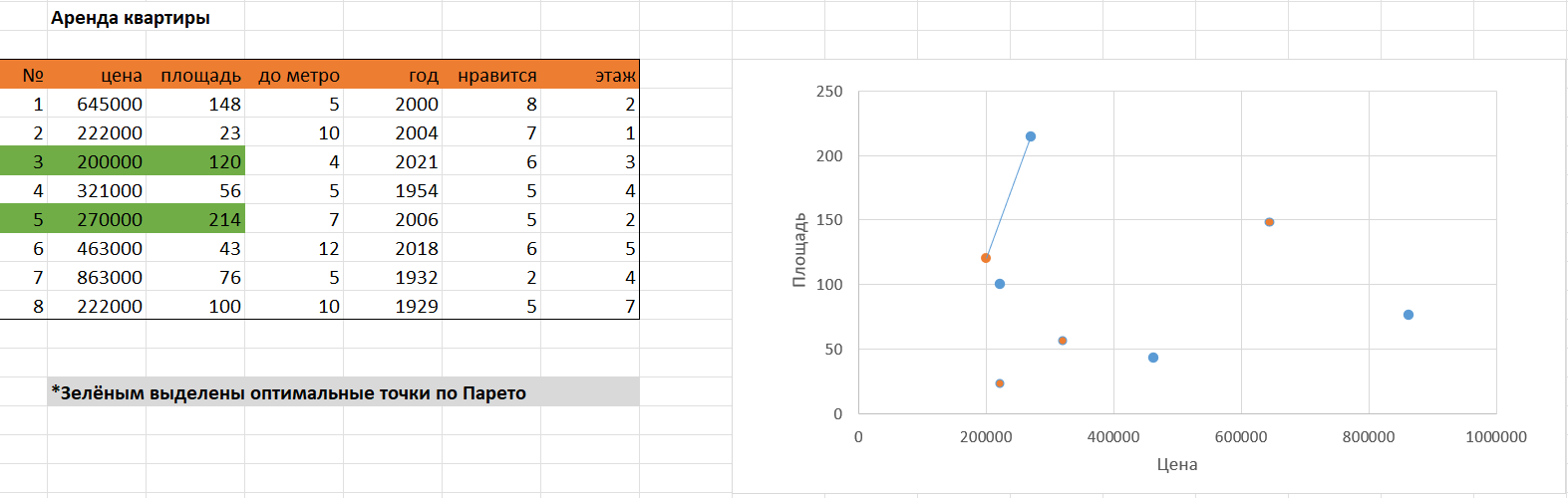


Рисунок . Паретооптимально

Изображение выглядит как стол

Автоматически созданное описание

Рисунок . линейная свёрстка

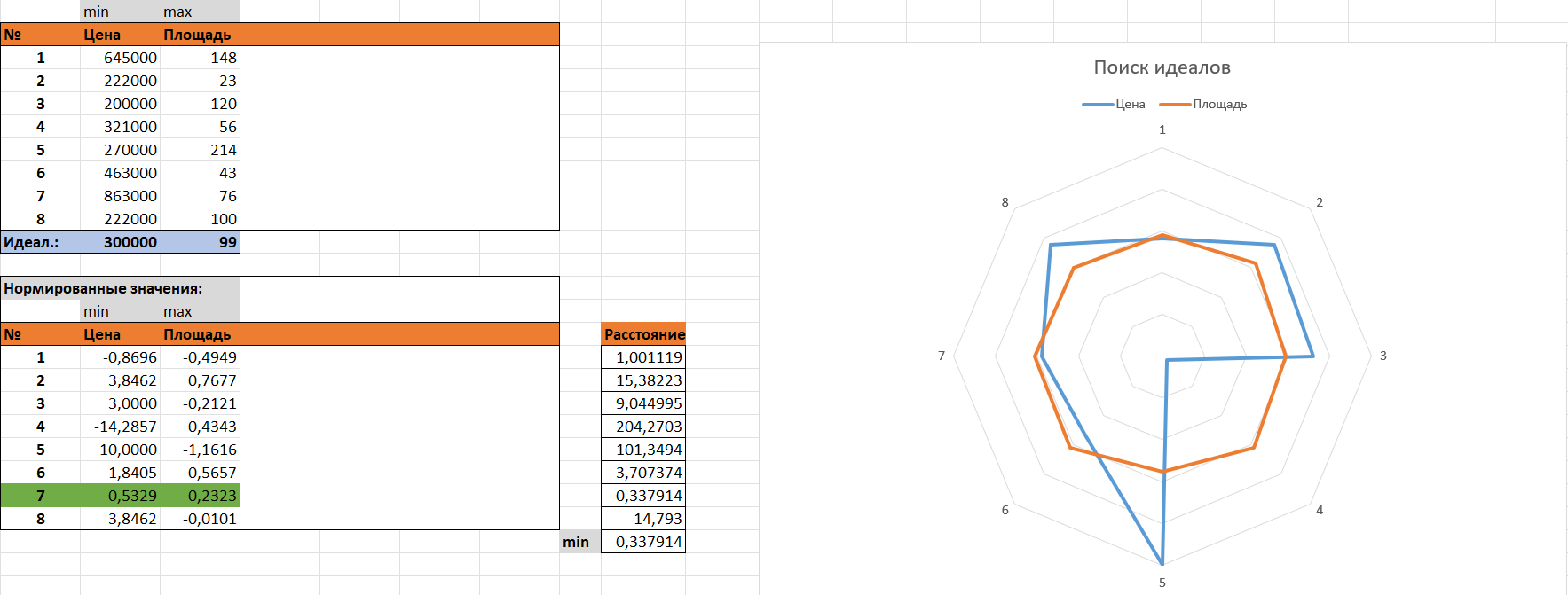


Рисунок .Идеальная точка

# 7. Заключение

Представленный нами код решает поставленную задачу. На основании тестирования данного алгоритма можно сделать вывод о том, что Python выводит самое оптимальное решение быстро и наглядно. Теперь сравним два алгоритма по критериям: эффективности, скорости использования алгоритма, простоты использования, надёжности в разрезе человеческого фактора и точности предоставляемого решения.

Таблица 2. Сравнение Питона и Excel

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

Мы считаем, что представленный рукописный код на языке Python лучше, потому что удобнее, быстрее и проще, имеет функцию импорта исходных данных, а также более наглядную визуализацию относительно Excel. Улучшением кода может послужить добавление времени выполнения запроса, более детальной выводимой информации, ручного ввода данных, в том числе генерация случайных данных.