**Лабораторная работа №**

**ЗАДАЧА КОММИВОЯЖЕРА**

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 2](#_Toc3902960)

[**1.1. Общие указания к выполнению лабораторной работы** 4](#_Toc3902961)

[**1.2. Цель работы** 4](#_Toc3902962)

[**1.3. Теоретический материал** 4](#_Toc3902963)

[**1.4. Алгоритм решения задач** 6](#_Toc3902964)

[**1.5. Представление исходных данных в таблице EXCEL (методический пример)** 10](#_Toc3902965)

[**1.6. Алгоритм поиска оптимального плана с помощью EXCEL** 11](#_Toc3902966)

[**1.7. Пример решения задачи с помощью MS Excel (с описанием)** 12](#_Toc3902967)

[**1.8. Задание к лабораторной работе** 14](#_Toc3902968)

[**Задача №1** 14](#_Toc3902969)

[**Задача №2** 16](#_Toc3902970)

[**Задача №3** 16](#_Toc3902971)

[**СПИСОК ЛИТЕРАТУРЫ** 16](#_Toc3902972)

# **ВВЕДЕНИЕ**

Математические проблемы, связанные с задачей коммивояжера, впервые были сформулированы в позапрошлом веке двумя математиками - Уильямом Гамильтоном и Томасом Киркменом. Гамильтон придумал игру, в которой игрокам необходимо было составить тур через 20 точек, используя только нарисованные связи между точками (фигура соответствовала додекаэдру). Задача коммивояжера в общем виде (с минимизацией длины тура) впервые начала изучаться в 20-ых годах прошлого века гарвардским математиком и экономистом Карлом Менгером. В течение следующих 20 лет изучением проблемы занимается все большее и большее количество ученых и, в конечном счете, задача приобретают дурную славу в связи со своей "труднорешаемостью".

Вскоре появилось известное сейчас название задача странствующего продавца (англ. Traveling Salesman Problem), которую предложил Гаслер Уитни (англ. Hassler Whitney) из Принстонского университета.

Вместе с простотой определения и сравнительной простотой нахождения хороших решений задача коммивояжера отличается тем, что нахождение действительно оптимального пути является достаточно сложной задачей. Учитывая эти свойства, начиная со второй половины 20-го века, исследование задачи коммивояжера имеет не столько практический смысл, сколько теоретический в качестве модели для разработки новых алгоритмов оптимизации.

Многие современные распространенные методы дискретной оптимизации, такие как метод деления плоскостью, ветвей и границ и различные варианты эвристических алгоритмов, были разработаны на примере задачи коммивояжера.

В 1950-е и 1960-е годы задача коммивояжера привлекла внимание ученых в США и Европе. Важный вклад в исследование задачи принадлежит Джорджу Данцигу, Делберту Рею Фалкерсону (англ. Delbert Ray Fulkerson) и Селмеру Джонсону (англ. Selmer M. Johnson), которые в 1954 году в институте RAND Corportation сформулировали задачу в виде задачи дискретной оптимизации и разработали метод деления плоскостью для ее решения. Используя новый метод, они вычислили путь для отдельного набора узлов с 49 городами и доказали, что не существует короткого пути.

В 1960-е и 1970-е годы многочисленные группы исследователей изучали задачу с точки зрения математики и ее применения, например, в информатике, экономике, химии и биологии.

Ричард Карп в 1972 году доказал NP-полноту задачи поиска гамильтоновых путей, из чего, благодаря полиномиальной сводимости, вытекала NP-полнота задачи коммивояжера. На основе этих свойств им было приведено теоретическое обоснование сложности поиска решений задачи на практике.

Больших успехов удалось достичь в конце 1970-х и 1980-х годах, когда Мартин Грётчел (нем. Martin Grötschel), Манфред Падберг (нем. Manfred Padberg) и Гиованни Ринальди (нем. Giovanni Rinaldi) и другие, с применением новых методов деления плоскостью, ветвей и границ вычислили решение для отдельного случая задачи с 2393 городами.

В 1990-е годы Дэвид Аплгейт (англ. David Applegate), Роберт Биксби (англ. Robert Bixby), Вашека Шватал (англ. Vašek Chvátal) и Уильям Кук (англ. William Cook) установили рекорды по программе Конкорд. Герхард Райнельт (нем. Gerhard Reinelt) создал TSPLIB — набор стандартизованных экземпляров задачи коммивояжера различной степени сложности для сравнения результатов работы различных групп исследователей. В марте 2005 года задача с 33 810 узлами была решена программой Конкорд: был вычислен путь длиной в 66 048 945 и доказано отсутствие коротких путей. В апреле 2006 было найдено решение для экземпляра с 85 900 узлами. Используя методы декомпозиции, можно вычислить решения для случаев задачи с миллионами узлов, длина которых менее чем на 1 % больше оптимальной.

Прогресс в решении задачи не стоит на месте, размерности решенных графов растут уже по экспоненте, что связано как с растущей мощностью вычислительных систем (производительность которых согласно одной теории тоже растет по экспоненциальному закону), так и с новыми эвристическими методами. В результате, в настоящее время предпринимаются попытки решить метрическую задачу коммивояжера размерности около двух миллионов, отражающей в себе все населенные пункты земного шара.

Задача коммивояжера – одна из наиболее важных задач транспортной логистики. Суть задачи сводится к нахождению наилучшего маршрута, т.е. наиболее короткого, а следовательно и менее затратного во временном (и соответственно экономическом) отношении. По условию задачи «коммивояжер» выезжает из некоторого начального города и посещает другие города в количестве n-1, где n – общее количество пунктов назначения. Задается матрица расстояний (Cij) между пунктами, где i и j изменяются от 1 до n. При этом необходимо учесть следующие ограничения:

* коммивояжер въезжает в любой пункт только 1 раз;
* коммивояжер выезжает из каждого пункта только 1 раз;
* маршрут является замкнутым, без петель.

# **1.1. Общие указания к выполнению лабораторной работы**

Лабораторная работа выполняется на персональном компьютере в операционной системе Windows, с использованием продуктов Microsoft Office и/или с применением языков программирования (макросы).

# **1.2. Цель работы**

Изучить метод коммивояжера (комбинаторная оптимизация). Задача коммивояжера относится к классу транспортных задач с дополнительными условиями на целостность и замкнутость маршрута.

# **1.3. Теоретический материал**

Рассмотрим задачу коммивояжера (англ. TSP – Travelling Salesman Problem) заключается в отыскании самого короткого маршрута, проходящего через заданные города по одному разу с последующим возвратом в исходный город.

Существует несколько частных случаев общей постановки задачи:

* геометрическая задача коммивояжёра (также называемая планарной или евклидовой, когда матрица расстояний отражает расстояния между точками на плоскости);
* треугольная задача коммивояжёра (когда на матрице стоимостей выполняется неравенство треугольника);
* симметричная задача коммивояжера;
* асимметричная задачи коммивояжёра;
* обобщение задачи задача коммивояжёра.

Математическая модель имеет следующий вид:

Задача коммивояжера на максимум в варианте распознавания определяется следующим образом.

Даны целое число n, (n×n) – матрица (cij) с неотрицательными элементами и число L . Требуется определить, существует ли такая циклическая перестановка (обход) длины n , что ∑ j=1 n c jτ( j)≥ L . Оптимизационную задачу коммивояжера на максимум определим несколькими способами.

1. **Классическая постановка.**

Имеется n городов, которые должен обойти коммивояжер с максимальными затратами. При этом на его маршрут накладывается два ограничения:

1) маршрут должен быть замкнутым, то есть коммивояжер должен вернуться в тот город, из которого он начал движение;

2) в каждом из городов коммивояжер должен побывать точно один раз, то есть надо обязательно обойти все города, при этом не побывав ни в одном городе дважды.

1. **В терминах графов.**

Пусть дан обыкновенный связный взвешенный граф (веса ребер заданы матрицей (cij) или весовой функцией c (vi, v j ) ). Требуется найти гамильтонов цикл наибольшего веса, где вес цикла определяется как сумма весов входящих в него ребер.

1. **В терминах перестановок.**

Пусть дана произвольная матрица (cij) порядка n с неотрицательными вещественными элементами. Требуется найти такую перестановку (i1 ,i2 ,…,in ) множества {1, 2,…,n } , что сумма ci1 i2 +ci2 i3 +…+cin i1 принимает максимальное значение. В данной работе рассматривается задача коммивояжера на максимум. Для нее будем использовать обозначение MAX TSP (MAX TSP – Maximum Traveling Salesman Problem (задача коммивояжера на максимум)) и в дальнейшем пользоваться определением ее через графы, так как в этом случае легче всего формально описывать задачу. Задача TSP является задачей комбинаторной оптимизации (так как множество ее допустимых решений дискретно и конечно) и является NP -трудной.

**Математическая модель задачи коммивояжера:**

Определим булевы переменные задачи:

xij = 1, если коммивояжер переезжает из города i в город j;

xij = 0, если коммивояжер не переезжает из города i в город j.

Тогда задача заключается в определении минимума целевой функции:

выражающей минимальную длину маршрута, при ограничениях:

j = 1,2,…,n - коммивояжер может въехать в город j только один раз,

i= 1,2,…,n - коммивояжер может выехать из города i только один раз.

В задаче коммивояжера в число ограничений необходимо ввести еще одну систему условий: ui - uj + (n-l) ∙ xij ≤ n-2, i≠j, i,j = 2,..., n, которые обеспечивают устранение распадения единого маршрута на несколько не связанных между собой (частичных) маршрутов, а также предотвращают образование циклов, означающих перемещение коммивояжера по замкнутому частичному маршруту.

Решение задачи коммивояжера возможно разными методами.

1. Решение задачи коммивояжера с помощью метода ветвей и границ;
2. Решение задачи коммивояжера с заданной матрицей расстояний алгоритмом Литтла (или исключения подциклов);
3. Решение задачи коммивояжера с помощью венгерского алгоритма

# **1.4. Алгоритм решения задач**

**Задача коммивояжера**, известная также как **задача о сверлильном станке** или **алгоритм коммивояжера** была поставлена в 1934 году. Эта задача является одной из знаменитых задач теории комбинаторики и широко применяется при разработке программного обеспечения.

Целью данной лабораторной работы является постановка задачи коммивояжера и ее решение методом полного перебора с использованием надстройки MS Excel «Поиск решения».

**Для решения задачи №1 в Excel:**

Разместим исходные данные на рабочем листе (рис 1). По главной диагонали матрица запишем 10000 (на результат решения исключение пути не оказывает влияния).

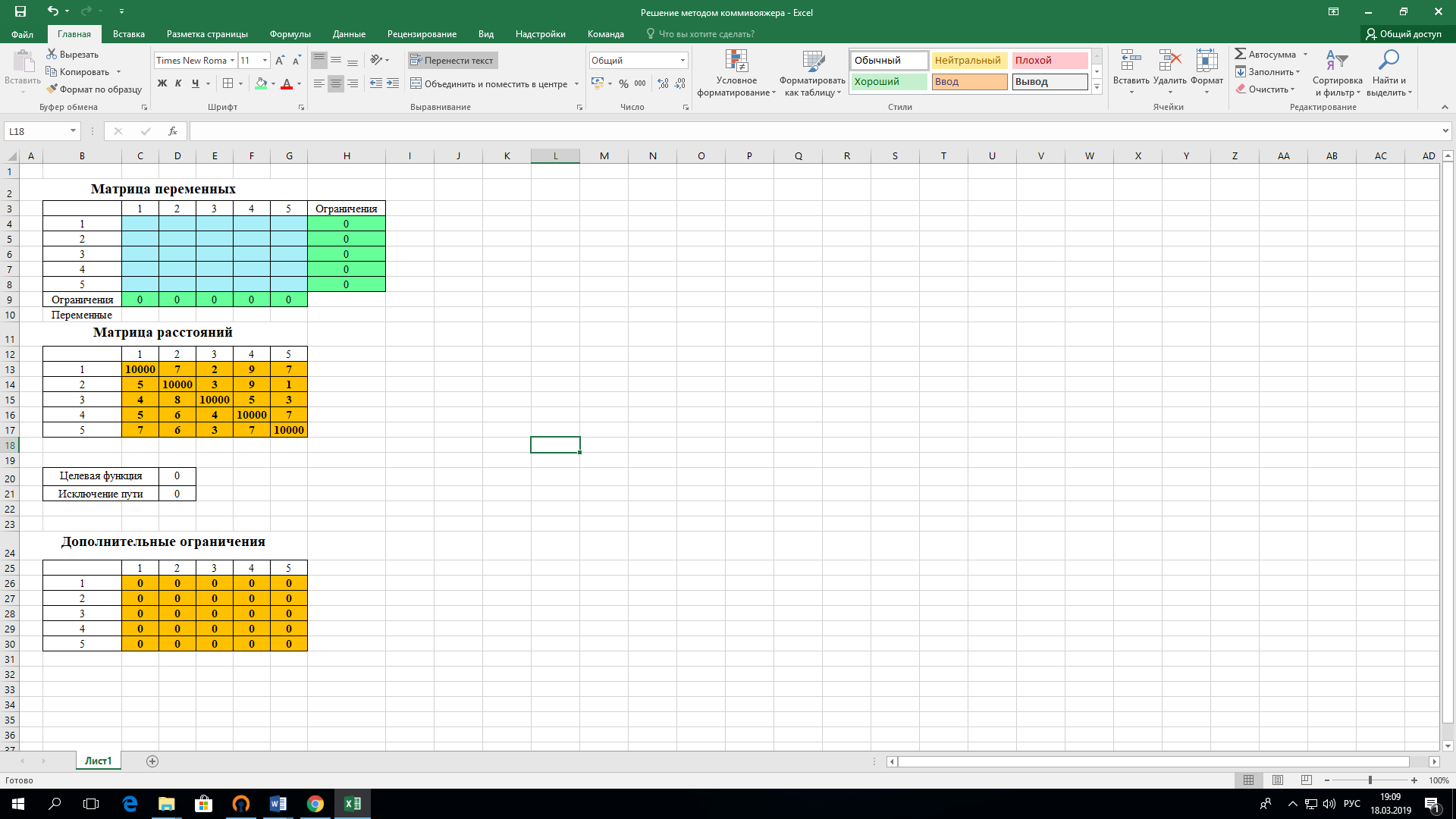


Рисунок 1 Исходные данные задачи

Вводим формулы в указанные ячейки:

|  |  |  |
| --- | --- | --- |
| **Ячейка** | **Формула** | **Примечание** |
| C9 | = СУММ(C4:C8) | Распространяем на диапазон C9:G9 |
| H4 | = СУММ(C4:G4) | Распространяем на диапазон H4:H8 |
| D20 | =СУММПРОИЗВ(C4:G8;C13:G17) | Целевая функция |
| D21 | =C4+D5+E6+F7+G8 | Исключение пути |
| B23 | =$C$10-C10+4\*C5 | Распространяем на диапазон B23:E23 |
| B24 | =$D$10-C10+4\*C6 | Распространяем на диапазон B24:E24 |
| B25 | =$E$10-C10+4\*C7 | Распространяем на диапазон B25:E25 |
| B26 | =$F$10-C10+4\*C8 | Распространяем на диапазон B26:E26 |

**Алгоритм решения.**

1. Запускаем надстройку MS Excel «Поиск решения» командой Сервис/ Поиск решения. И заполняем.
2. Для того чтобы выполнялись условия однократного посещения сотрудником организаций и в то же время запланированный Петровым маршрут был пройден полностью, введем ограничения: в строки B9, G4 заводим формулы из таблицы 1 и распространяем их на соответствующие диапазоны B9:F9 и G4:G8. Задаем следующие данные $B$9:$F$9=1 и $G$4:$G$8=1 в Ограничения окна «Поиск решения». Таким образом, мы можем отследить порядок обхода организаций сотрудником, оценить правильность выбора и оптимальность его маршрута.
3. Выбираем ячейку B19 и устанавливаем ее адрес в Целевую ячейку окна «Поиск решения», чтобы определить длину наикратчайшего маршрута. Для этого в ячейку B19 предварительно заносим соответствующую формулу из таблицы 1. Когда программа «Поиск решения» вычислит оптимальный маршрут Петрова и станет известен порядок обхода организаций (из Матрицы переменных) будут известны и расстояния между конкретными парами организаций. Затем при помощи простых математических подсчетов программа рассчитает протяженность оптимального маршрута.
4. Устанавливаем еще одно ограничение в окно «Поиск решения»: $E$19=0. В указанную ячейку вводим формулу из таблицы 1 и исключаем таким образом, заведомо ложный порядок движения Петрова в порядке обхода организаций.
5. В связи с тем, что ячейки диапазона B4:F8 – изменяемые, в Ограничение окна «Поиск решения» необходимо добавить строку B$4$:F$8$=двоичное.
6. Заводим в ячейки B23; B24; B25; B26 соответствующие формулы из таблицы 1 и распространяем их на следующие диапазоны: B23:E23 B24:E24; B25:E25; B26:E26 для учета всех возможных вариантов обхода организаций сотрудником и выбора из них оптимального. Формулы задаем таким образом, чтобы обеспечить исключение ложного пути, соблюдая условие задачи об обходе всех организаций по одному разу.
7. Добавляем в Ограничения окна «Поиск решения» $B$23:$E$26 ≤ 3.

Так как это линейная модель, то необходимо фиксировать в окне Параметры поиска решений на позицию Линейная модель и Неотрицательные значения (рис. 2.3). После того, как все поля и ячейки заполнены нажимаем кнопку «Выполнить» и появляется окно диалога с описанием результатов процесса оптимизации. Чтобы отобразить найденное решение в ячейках листа, устанавливаем переключатель «Сохранить найденное решение» и нажимаем кнопку ОК. Найденная минимальная величина помещается в целевую ячейку, а переменные ячейки заполняются оптимальными значениями переменных, которые удовлетворяют установленным ограничениям.

# **1.5. Представление исходных данных в таблице EXCEL (методический пример)**

**Для решения задачи №2 в Excel:**

1. Даны координаты городов:

**Таблица 1 Координаты городов**

|  |  |  |  |
| --- | --- | --- | --- |
| **Город** | **№ Города** | **Х** | **Y** |
| *Город0* | **0** | 500 | 500 |
| *Город1* | **1** | 1000 | 2500 |
| *Город2* | **2** | 2465 | 1779 |
| *Город3* | **3** | 2000 | 250 |
| *Город4* | **4** | 1200 | 1400 |

1. Рассчитать расстояние между городами. В таблицу 2 получен результат расчета (формула расчета приведена ниже):

**=КОРЕНЬ((ИНДЕКС($C$7:$D$11;$F7+1;1)-ИНДЕКС($C$7:$D$11;G$6+1;1))^2+(ИНДЕКС($C$7:$D$11;$F7+1;2)-ИНДЕКС($C$7:$D$11;G$6+1;2))^2)**

**Таблица 2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Расстояния между городами** | | |  |  |  |
| **Города** | **0** | **1** | **2** | **3** | **4** |
| **0** | 0,0 | 2061,6 | 2344,6 | 1520,7 | 1140,2 |
| **1** | 2061,6 | 0,0 | 1632,8 | 2462,2 | 1118,0 |
| **2** | 2344,6 | 1632,8 | 0,0 | 1598,1 | 1320,6 |
| **3** | 1520,7 | 2462,2 | 1598,1 | 0,0 | 1400,9 |
| **4** | 1140,2 | 1118,0 | 1320,6 | 1400,9 | 0,0 |

1. Подготовить таблицу для «Принятия решения» (таблица 3). Для принятия решения блокам задать имена:

|  |  |
| --- | --- |
| **Цвет блока** | **Имя блока** |
|  | Номера |
|  | Длина\_пути |
|  | Число\_уникальных |

**Таблица 3**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **А** | **В** | **С** |
| **14** | **Город** | **Последовательность** | **Расстояние** |
| **15** | **Город0** | **0** |  |
| **16** | **Город1** | **1** | **2061,6** |
| **17** | **Город2** | **2** | **1632,8** |
| **18** | **Город3** | **3** | **1598,1** |
| **19** | **Город4** | **4** | **1400,9** |
| **20** | **Город0** | **0** | **1140,2** |
| **21** |  | **Длина пути** | **7833,6** |
| **22** |  |  |  |
| **23** |  | **Число уникальных** | **4** |

1. Ввести в ячейки формулы (пример формул приведен ниже и в таблице 4)

формула в ячейке А15: **="Город"&B15**

формула в ячейке С16: **=ИНДЕКС($G$7:$K$11;B15+1;B16+1)**

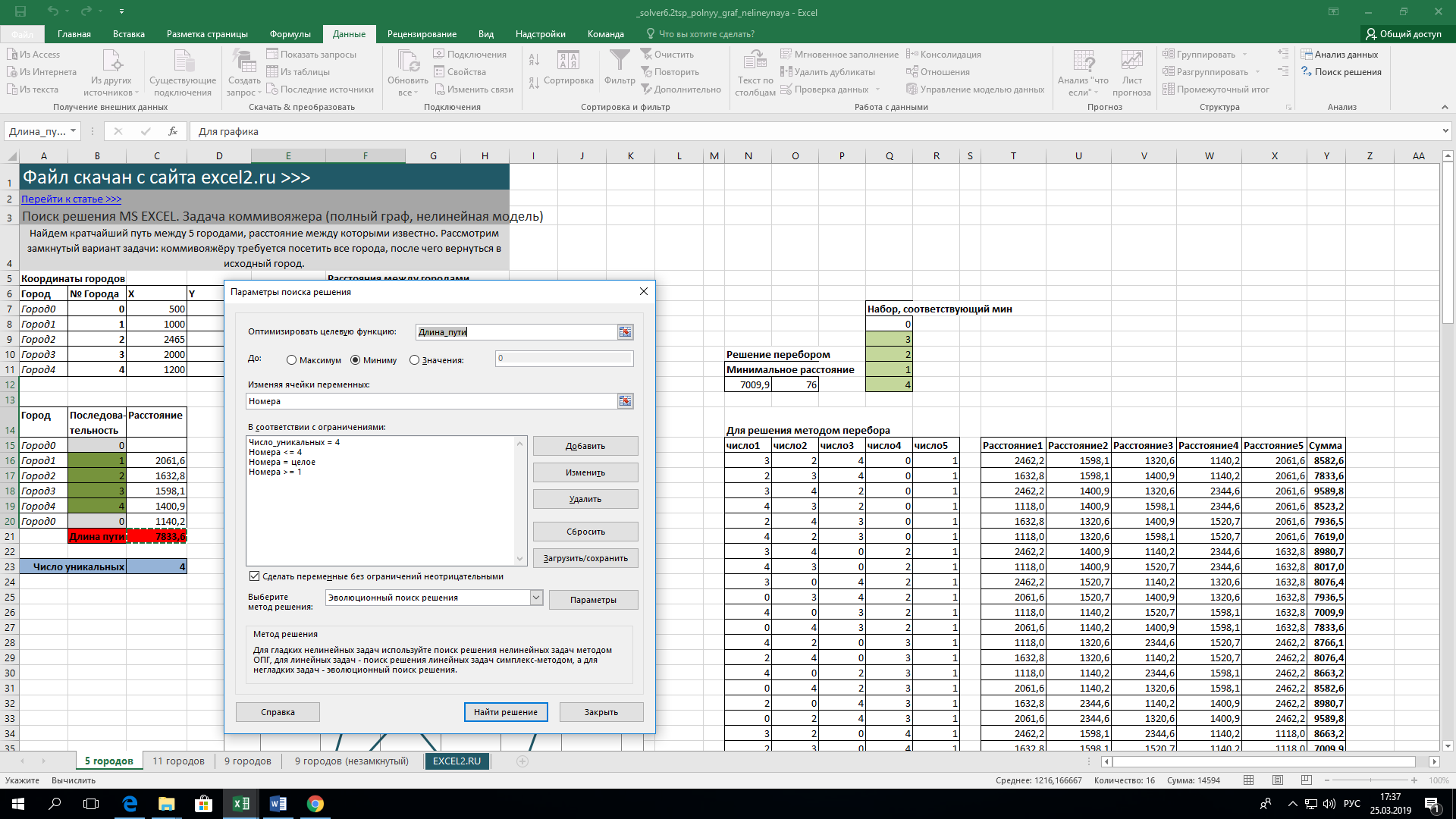
формула в ячейке С23: **=СУММПРОИЗВ(1/СЧЁТЕСЛИ(B16:B19;B16:B19))**

**Таблица 4**

|  |  |  |
| --- | --- | --- |
| **Город** | **Последовательность** | **Расстояние** |
| *="Город"&B15* | 0 |  |
| *="Город"&B16* | 1 | =ИНДЕКС($G$7:$K$11;B15+1;B16+1) |
| *="Город"&B17* | 2 | =ИНДЕКС($G$7:$K$11;B16+1;B17+1) |
| *="Город"&B18* | 3 | =ИНДЕКС($G$7:$K$11;B17+1;B18+1) |
| *="Город"&B19* | 4 | =ИНДЕКС($G$7:$K$11;B18+1;B19+1) |
| *="Город"&B20* | 0 | =ИНДЕКС($G$7:$K$11;B19+1;B20+1) |
|  | **Длина пути** | **=СУММ(C16:C20)** |
|  |  |  |
|  | **Число уникальных** | **=СУММПРОИЗВ(1/СЧЁТЕСЛИ(B16:B19;B16:B19))** |

1. Запустить надстройку MS Excel «Поиск решения» (Данные - Поиск решения). Заполнить диалоговое окно в соответствии с образцом (таблица 5). В качестве заполненных полей использовать поименованные блоки.

**Таблица 5**



1. Получить результат как в таблице 6.

**Таблица 6**

|  |  |  |
| --- | --- | --- |
| **Город** | **Последовательность** | **Расстояние** |
| *Город0* | 0 |  |
| *Город4* | 4 | 1140,2 |
| *Город1* | 1 | 1118,0 |
| *Город2* | 2 | 1632,8 |
| *Город3* | 3 | 1598,1 |
| *Город0* | 0 | 1520,7 |
|  | **Длина пути** | **7009,9** |
|  |  |  |
|  | **Число уникальных** | **4** |

1. Подготовить данные для построения графика (таблица 7) (в режиме формул - таблица 8):

**Таблица 7 Для построения графика**

|  |  |
| --- | --- |
| **Кратчайший путь** | |
| **X** | **Y** |
| 500 | 500 |
| 1000 | 2500 |
| 2465 | 1779 |
| 2000 | 250 |
| 1200 | 1400 |
| 500 | 500 |

**Таблица 8 Для построения графика (в виде формул)**

|  |  |
| --- | --- |
| **Кратчайший путь** |  |
| **X** | **Y** |
| =ИНДЕКС($B$7:$C$11;B15+1;2) | =ИНДЕКС($B$7:$D$11;B15+1;3) |
| =ИНДЕКС($B$7:$C$11;B16+1;2) | =ИНДЕКС($B$7:$D$11;B16+1;3) |
| =ИНДЕКС($B$7:$C$11;B17+1;2) | =ИНДЕКС($B$7:$D$11;B17+1;3) |
| =ИНДЕКС($B$7:$C$11;B18+1;2) | =ИНДЕКС($B$7:$D$11;B18+1;3) |
| =ИНДЕКС($B$7:$C$11;B19+1;2) | =ИНДЕКС($B$7:$D$11;B19+1;3) |
| =ИНДЕКС($B$7:$C$11;B20+1;2) | =ИНДЕКС($B$7:$D$11;B20+1;3) |

1. Построить график движения коммивояжера.

Решение задачи представлено на рисунке 1

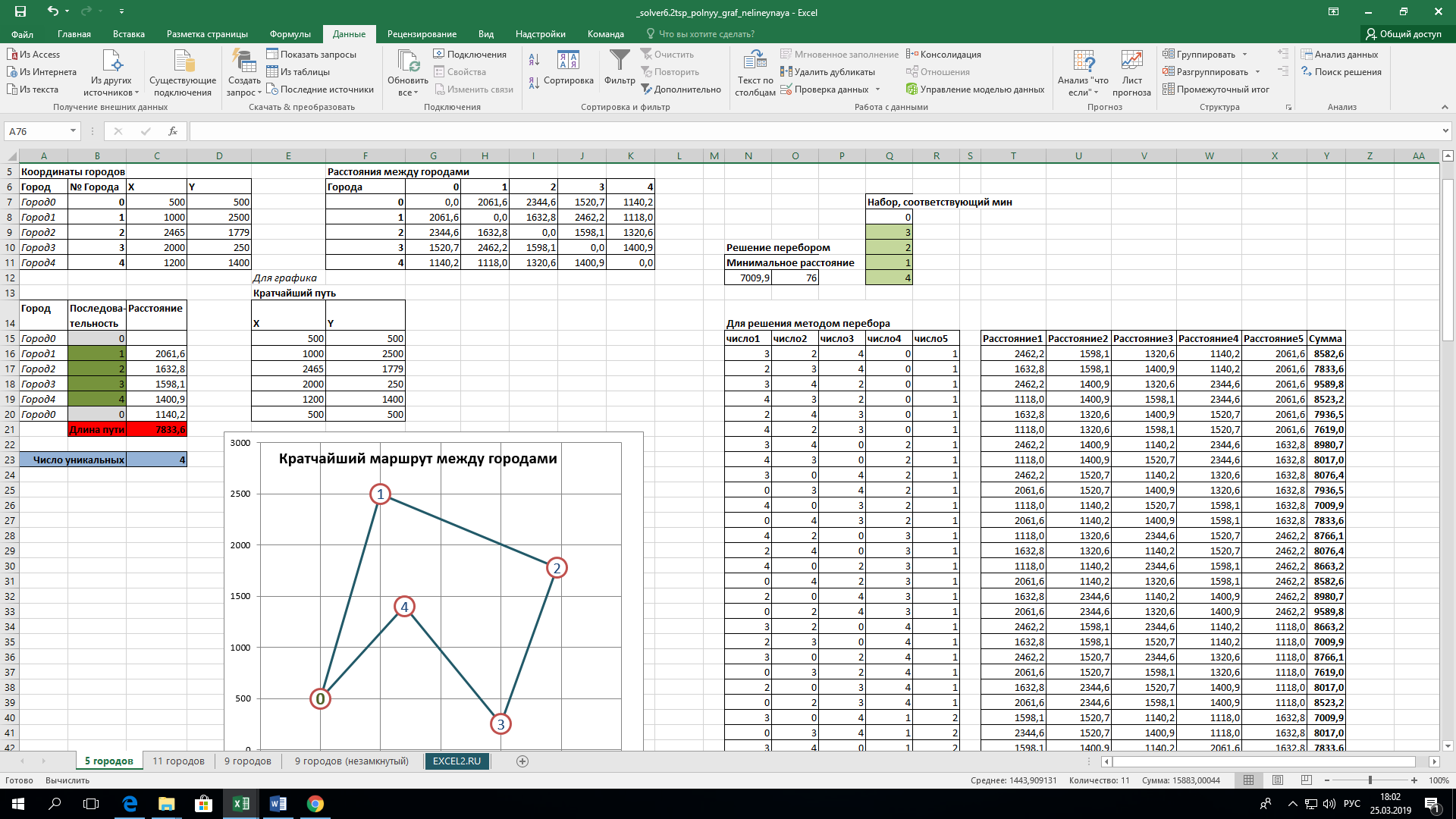


Рисунок 1. Решение задачи №2

**Для решения задачи №3 в Excel:**

Исходные данные: Имеется 9 пунктов, которые должен посетить коммивояжёр. Известна широта и долгота для каждого пункта, расстояния между которыми нужно рассчитать. Найти кратчайший путь между 2-мя заданными городами.

1. Построить Линейную модель.
2. Написать Макросы: «Сброс» - сброс решения, «Случайно» - случайная генерация широты и долготы.

На рисунке 2 пример решенного задания №3.

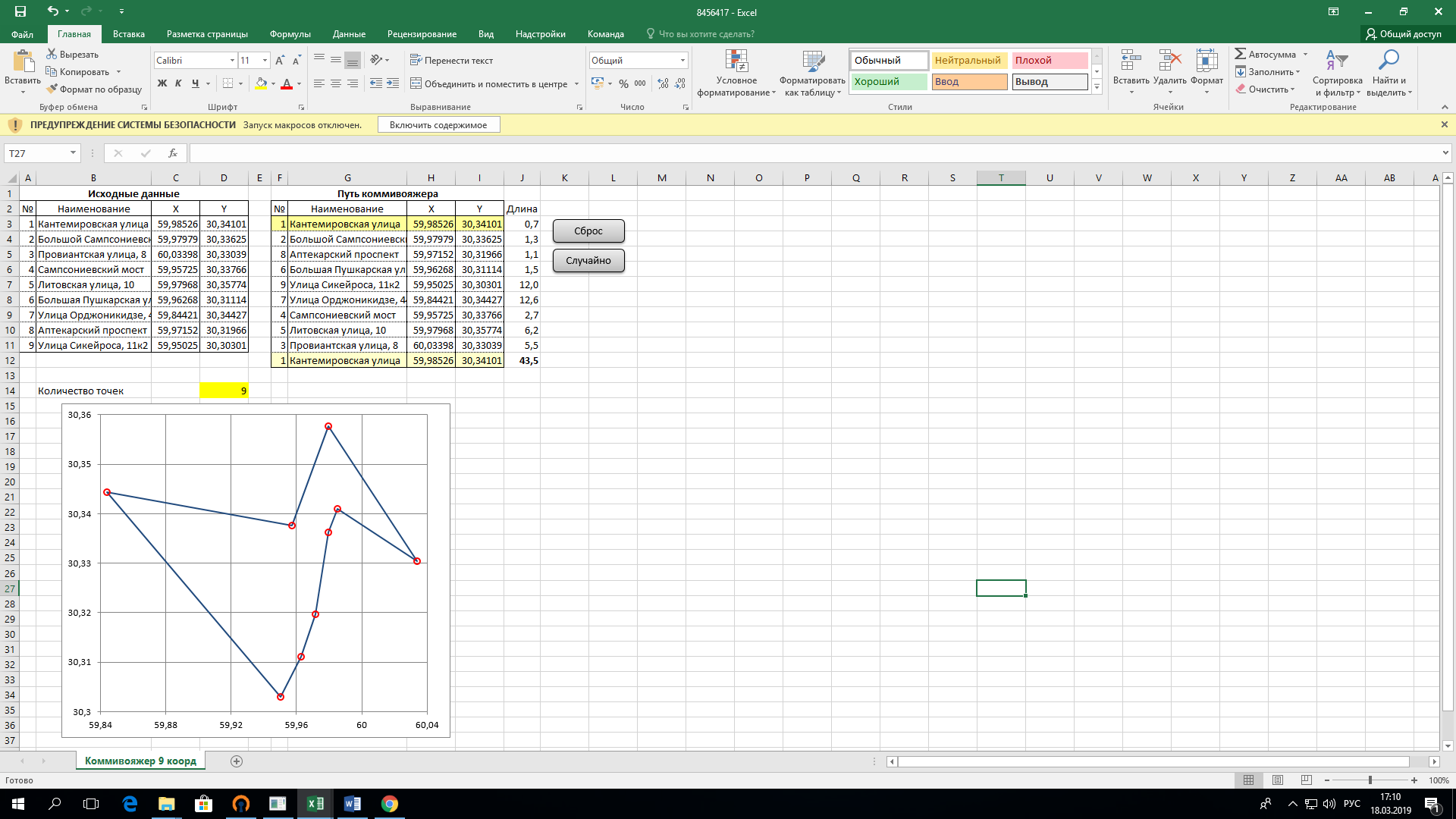
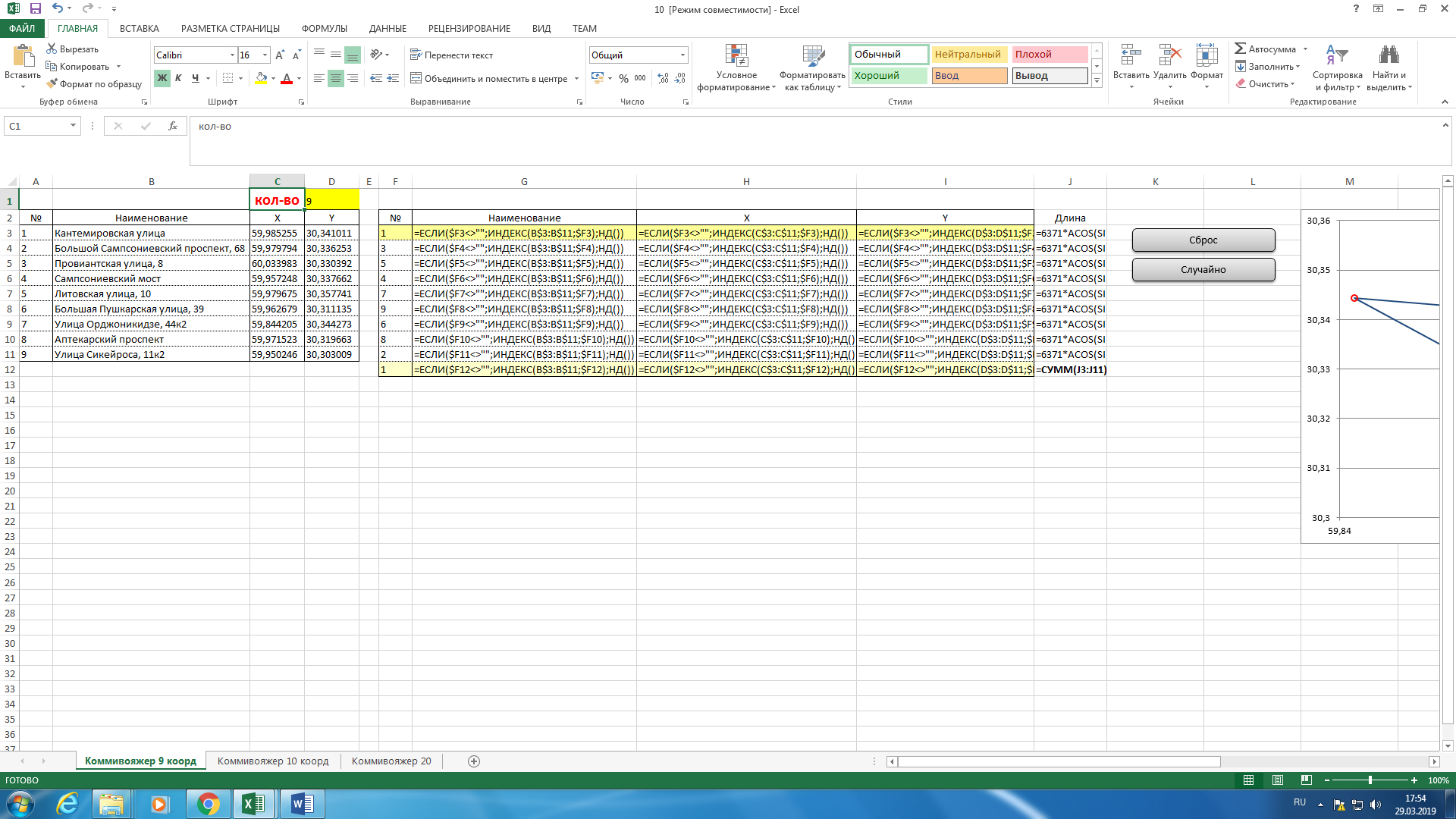


Рисунок 2. Решение задачи №3



формула в ячейке G3: **=ЕСЛИ($F3<>"";ИНДЕКС(B$3:B$11;$F3);НД())**

формула в ячейке H3: **=ЕСЛИ($F3<>"";ИНДЕКС(C$3:C$11;$F3);НД())**

формула в ячейке I3: **=ЕСЛИ($F3<>"";ИНДЕКС(D$3:D$11;$F3);НД())**

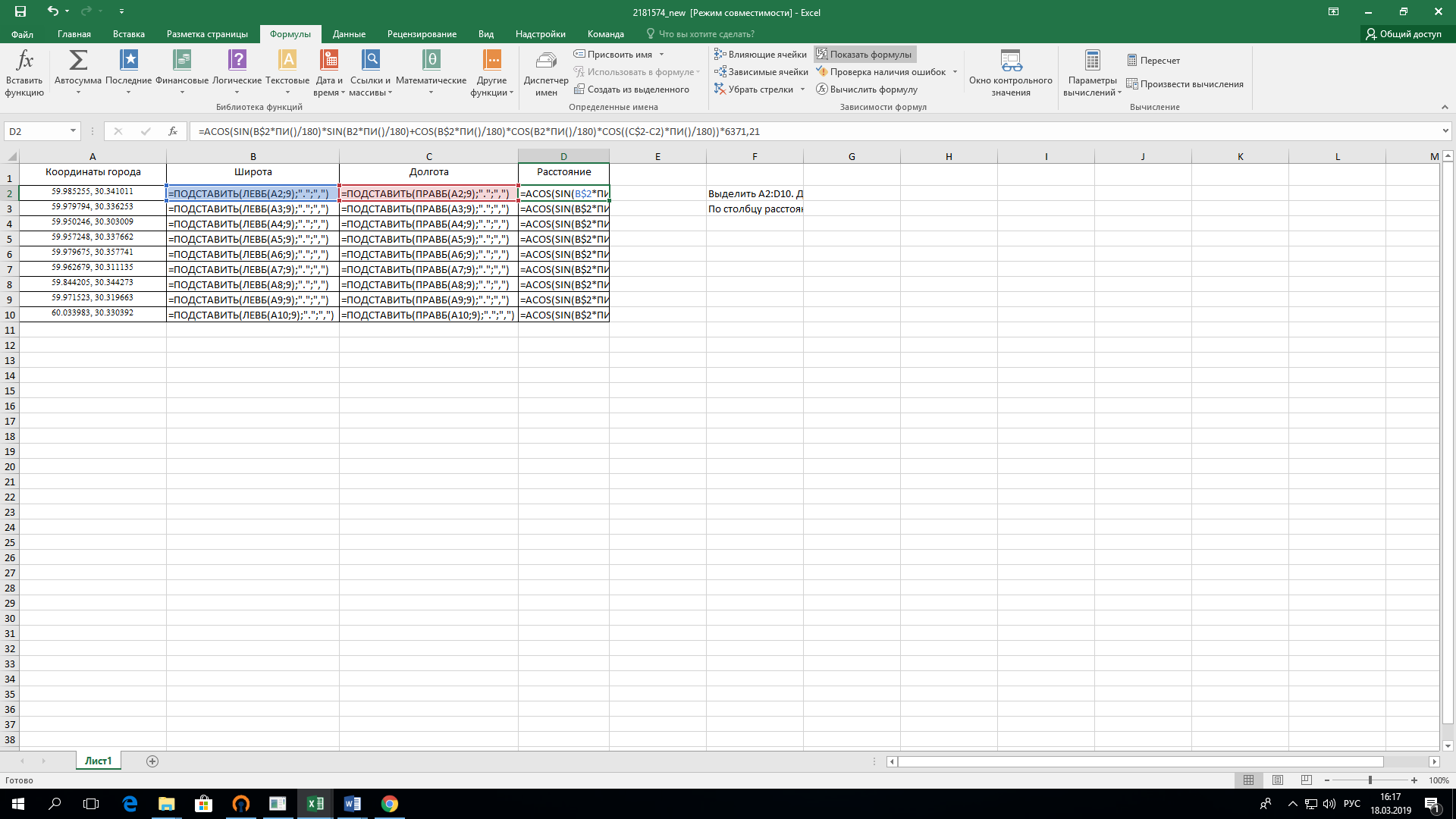
формула в ячейке J3: **=6371\*ACOS(SIN(РАДИАНЫ(H3))\*SIN(РАДИАНЫ(H4))+COS(РАДИАНЫ(H3))\*COS(РАДИАНЫ(H4))\*COS((РАДИАНЫ(I3-I4))))**

Если известны координаты (широта и долгота) города, то можно высчитать расстояние (пример расчета показан ниже таблица 1 и таблица 2).

Таблица 1.

|  |  |  |  |
| --- | --- | --- | --- |
| Координаты города | Широта | Долгота | Расстояние |
| 59.985255, 30.341011 | 59,985255 | 30,341011 | 0 |
| 59.979794, 30.336253 | 59,979794 | 30,336253 | 0,662431419 |
| 59.950246, 30.303009 | 59,950246 | 30,303009 | 4,430356403 |
| 59.957248, 30.337662 | 59,957248 | 30,337662 | 3,119910036 |
| 59.979675, 30.357741 | 59,979675 | 30,357741 | 1,118548511 |
| 59.962679, 30.311135 | 59,962679 | 30,311135 | 3,010938579 |
| 59.844205, 30.344273 | 59,844205 | 30,344273 | 15,68561533 |
| 59.971523, 30.319663 | 59,971523 | 30,319663 | 1,93450715 |
| 60.033983, 30.330392 | 60,033983 | 30,330392 | 5,450537523 |

Таблица 2.



**=ACOS(SIN(B$2\*ПИ()/180)\*SIN(B2\*ПИ()/180)+COS(B$2\*ПИ()/180)\*COS(B2\*ПИ()/180)\*COS((C$2-C2)\*ПИ()/180))\*6371,21**

где 6371,21 - радиус земли. Для сортировки на радиус земли можно не умножать.

Далее сортируем:

Выделить A2:D10. *Данные – Сортировка -* По столбцу расстояние, по возрастанию, Ок

# **1.6. Алгоритм поиска оптимального плана с помощью EXCEL**

1. Откройте меню Сервис и убедитесь, что в Вашей программе установлена надстройка ***Поиск решения***. Программа «Поиск решения» является надстройкой (Надстройка. Вспомогательная программа, служащая для добавления в Microsoft Office специальных команд или возможностей.) Excel, которая доступна после установки Microsoft Office или Microsoft Excel. Если же она отсутствует в Вашей установке, то для того чтобы использовать эту надстройку в Excel, необходимо прежде загрузить ее:

- в меню ***Сервис*** выберите команду ***Надстройки***.

- в поле ***Список надстроек*** установите флажок рядом с элементом ***Поиск решения***, а затем нажмите кнопку ***ОК***.

Совет. Если в списке отсутствует элемент Поиск решения, нажмите кнопку Обзор, чтобы найти надстройку самостоятельно.

- в случае появления сообщения о том, что надстройка «Поиск решения» не установлена на компьютере, нажмите кнопку ***Да***, чтобы установить ее.

- нажмите кнопку ***Сервис*** в строке меню. После загрузки надстройки «Поиск решения» в меню ***Сервис*** добавляется команда ***Поиск решения***.

2. Вызовите команду меню ***Сервис****→****Поиск решения***. Появится диалоговое окно оптимизатора.

3. В диалоговом окне ***Поиск решения*** в поле ***Установить целевую*** введите адрес $F$2. Установите флажок ***Минимальному значению***.

4. В поле ***Изменяя ячейки*** введите адреса диапазона (матрицы) искомых переменных целевой функции $B$1:$E$1.

5. В поле ***Ограничения*** введите пять строк неравенств значений диапазонов:

$H$4>=$G$4

$H$5>=$G$5

$H$6>=$G$6

$H$7>=$G$7

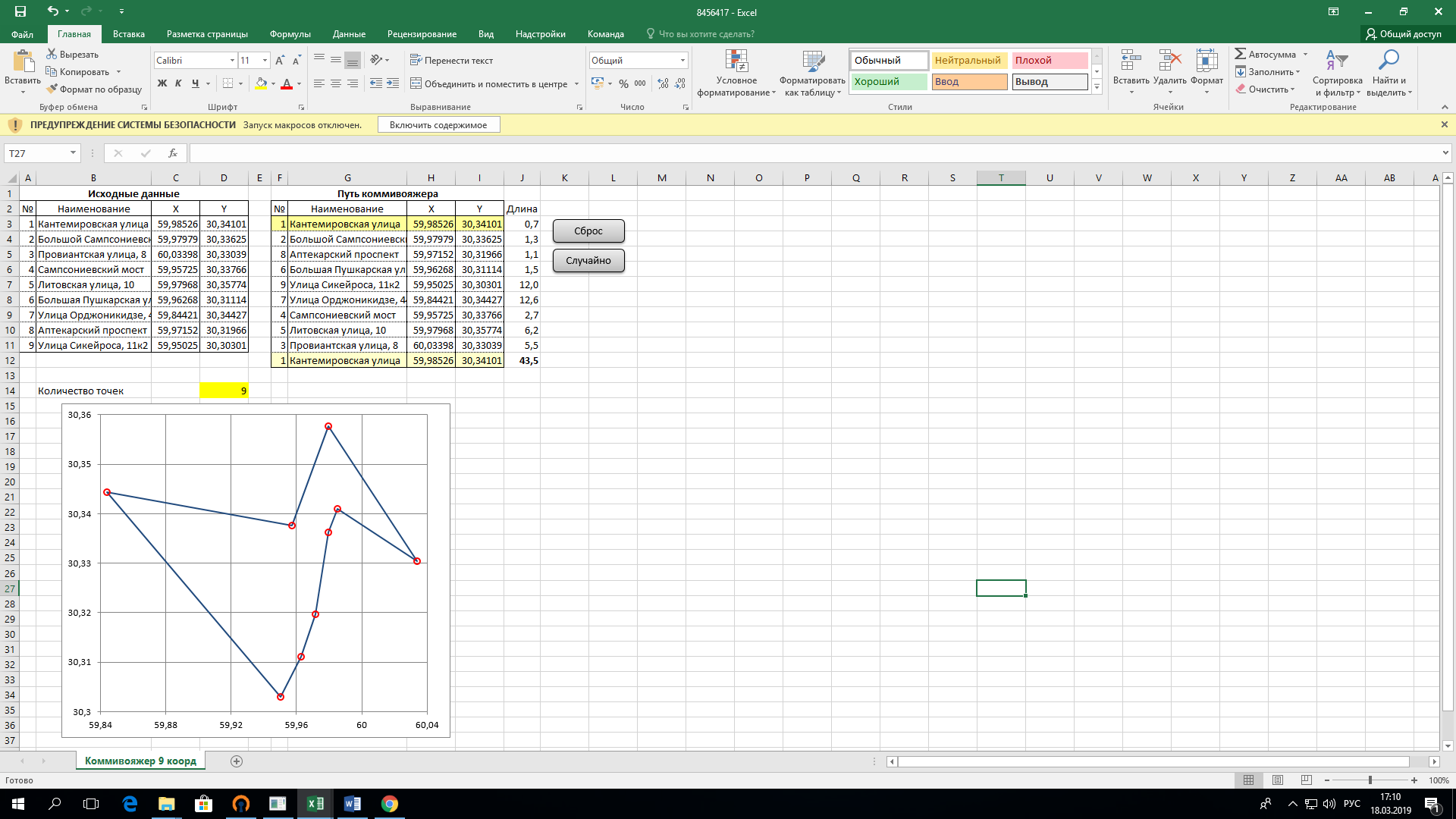
$H$8>=$G$8

6. Осуществить решение, нажав кнопку ***Выполнить*** окна ***Поиск решения***.

7. Анализ результатов и решения менеджера

Получено оптимальное решение, найденное программой EXCEL «Поиск решения»? Удовлетворены все ограничения? Если нет, то внести новые ограничения и повторить ***Поиск решения***.

# **1.7. Пример решения задачи с помощью MS Excel (с описанием)**



# **1.8. Задание к лабораторной работе**

1. Решить задачу №1 коммивояжера в **тетради**
2. Решить задачу №2 коммивояжера с вычислением координат точки и графиком.
3. Решить задачу №3 коммивояжера с привязкой к место положению городов (широта и долгота) + **макросы**.
4. **Отчет** в Word + **Решение** в Excel+**Тетрадь**

## **Задача №1**

Решить задачу коммивояжера в тетради и средствами Excel согласно своему варианту.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Исходная матрица** | **1** | **2** | **3** | **4** | **5** |
| 1 | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | 1 | 2 | 3 | 4 | 5 | | 1 |  | 1 | 4 | 3 | 2 | | 2 | 1 |  | 3 | 2 | 1 | | 3 | 4 | 3 |  | 1 | 2 | | 4 | 3 | 2 | 1 |  | 1 | | 5 | 2 | 1 | 2 | 1 |  | | из п.1 в п.1 | из п.2 в п.2 | из п.3 в п.3 | из п. 4 в п.4 | из п. 5 в п.5 |
| 2 | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | 1 | 2 | 3 | 4 | 5 | | 1 |  | 1 | 1 | 2 | 2 | | 2 | 1 |  | 2 | 3 | 1 | | 3 | 1 | 2 |  | 1 | 3 | | 4 | 2 | 3 | 1 |  | 4 | | 5 | 2 | 1 | 3 | 4 |  | | из п.1 в п.1 | из п.2 в п.2 | из п.3 в п.3 | из п. 4 в п.4 | из п. 5 в п.5 |
| 3 | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | 1 | 2 | 3 | 4 | 5 | | 1 |  | 2 | 1 | 3 | 1 | | 2 | 2 |  | 3 | 1 | 1 | | 3 | 1 | 3 |  | 4 | 2 | | 4 | 3 | 1 | 4 |  | 2 | | 5 | 1 | 1 | 2 | 2 |  | | из п.1 в п.1 | из п.2 в п.2 | из п.3 в п.3 | из п. 4 в п.4 | из п. 5 в п.5 |
| 4 | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | 1 | 2 | 3 | 4 | 5 | | 1 |  | 3 | 4 | 2 | 1 | | 2 | 3 |  | 1 | 1 | 2 | | 3 | 4 | 1 |  | 2 | 3 | | 4 | 2 | 1 | 2 |  | 1 | | 5 | 1 | 2 | 3 | 1 |  | | из п.1 в п.1 | из п.2 в п.2 | из п.3 в п.3 | из п. 4 в п.4 | из п. 5 в п.5 |
| 5 | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | 1 | 2 | 3 | 4 | 5 | | 1 |  | 2 | 1 | 4 | 3 | | 2 | 2 |  | 1 | 2 | 1 | | 3 | 1 | 1 |  | 3 | 2 | | 4 | 4 | 2 | 3 |  | 1 | | 5 | 3 | 1 | 2 | 1 |  | | из п.1 в п.1 | из п.2 в п.2 | из п.3 в п.3 | из п. 4 в п.4 | из п. 5 в п.5 |
| 6 | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | 1 | 2 | 3 | 4 | 5 | | 1 |  | 2 | 4 | 3 | 1 | | 2 | 2 |  | 2 | 1 | 1 | | 3 | 4 | 2 |  | 1 | 3 | | 4 | 3 | 1 | 1 |  | 2 | | 5 | 1 | 1 | 3 | 2 |  | | из п.1 в п.1 | из п.2 в п.2 | из п.3 в п.3 | из п. 4 в п.4 | из п. 5 в п.5 |
| 7 | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | 1 | 2 | 3 | 4 | 5 | | 1 |  | 1 | 2 | 3 | 1 | | 2 | 1 |  | 3 | 4 | 2 | | 3 | 2 | 3 |  | 1 | 1 | | 4 | 3 | 4 | 1 |  | 2 | | 5 | 1 | 2 | 1 | 2 |  | | из п.1 в п.1 | из п.2 в п.2 | из п.3 в п.3 | из п. 4 в п.4 | из п. 5 в п.5 |
| 8 | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | 1 | 2 | 3 | 4 | 5 | | 1 |  | 1 | 1 | 3 | 2 | | 2 | 1 |  | 2 | 2 | 1 | | 3 | 1 | 2 |  | 4 | 3 | | 4 | 3 | 2 | 4 |  | 1 | | 5 | 2 | 1 | 3 | 1 |  | | из п.1 в п.1 | из п.2 в п.2 | из п.3 в п.3 | из п. 4 в п.4 | из п. 5 в п.5 |
| 9 | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | 1 | 2 | 3 | 4 | 5 | | 1 |  | 3 | 1 | 2 | 1 | | 2 | 3 |  | 4 | 1 | 2 | | 3 | 1 | 4 |  | 3 | 2 | | 4 | 2 | 1 | 3 |  | 1 | | 5 | 1 | 2 | 2 | 1 |  | | из п.1 в п.1 | из п.2 в п.2 | из п.3 в п.3 | из п. 4 в п.4 | из п. 5 в п.5 |
| 10 | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | 1 | 2 | 3 | 4 | 5 | | 1 |  | 2 | 1 | 2 | 1 | | 2 | 2 |  | 1 | 4 | 3 | | 3 | 1 | 1 |  | 3 | 2 | | 4 | 2 | 4 | 3 |  | 1 | | 5 | 1 | 3 | 2 | 1 |  | | из п.1 в п.1 | из п.2 в п.2 | из п.3 в п.3 | из п. 4 в п.4 | из п. 5 в п.5 |
| 11 | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | 1 | 2 | 3 | 4 | 5 | | 1 |  | 1 | 2 | 2 | 1 | | 2 | 1 |  | 1 | 3 | 2 | | 3 | 2 | 1 |  | 4 | 3 | | 4 | 2 | 3 | 4 |  | 1 | | 5 | 1 | 2 | 3 | 1 |  | | из п.1 в п.1 | из п.2 в п.2 | из п.3 в п.3 | из п. 4 в п.4 | из п. 5 в п.5 |
| 12 | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | 1 | 2 | 3 | 4 | 5 | | 1 |  | 2 | 1 | 2 | 1 | | 2 | 2 |  | 3 | 4 | 1 | | 3 | 1 | 3 |  | 1 | 2 | | 4 | 2 | 4 | 1 |  | 3 | | 5 | 1 | 1 | 2 | 3 |  | | из п.1 в п.1 | из п.2 в п.2 | из п.3 в п.3 | из п. 4 в п.4 | из п. 5 в п.5 |
| 13 | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | 1 | 2 | 3 | 4 | 5 | | 1 |  | 3 | 4 | 2 | 1 | | 2 | 3 |  | 1 | 1 | 2 | | 3 | 4 | 1 |  | 2 | 3 | | 4 | 2 | 1 | 2 |  | 1 | | 5 | 1 | 2 | 3 | 1 |  | | из п.1 в п.1 | из п.2 в п.2 | из п.3 в п.3 | из п. 4 в п.4 | из п. 5 в п.5 |
| 14 | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | 1 | 2 | 3 | 4 | 5 | | 1 |  | 3 | 1 | 1 | 2 | | 2 | 3 |  | 4 | 2 | 1 | | 3 | 1 | 4 |  | 2 | 3 | | 4 | 1 | 2 | 2 |  | 1 | | 5 | 2 | 1 | 3 | 1 |  | | из п.1 в п.1 | из п.2 в п.2 | из п.3 в п.3 | из п. 4 в п.4 | из п. 5 в п.5 |
| 15 | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | 1 | 2 | 3 | 4 | 5 | | 1 |  | 1 | 2 | 3 | 1 | | 2 | 1 |  | 3 | 4 | 2 | | 3 | 2 | 3 |  | 1 | 1 | | 4 | 3 | 4 | 1 |  | 2 | | 5 | 4 | 2 | 1 | 2 |  | | из п.1 в п.1 | из п.2 в п.2 | из п.3 в п.3 | из п. 4 в п.4 | из п. 5 в п.5 |
| 16 | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | 1 | 2 | 3 | 4 | 5 | | 1 |  | 4 | 3 | 2 | 1 | | 2 | 4 |  | 4 | 3 | 2 | | 3 | 3 | 4 |  | 4 | 3 | | 4 | 2 | 3 | 4 |  | 4 | | 5 | 1 | 2 | 3 | 4 |  | | из п.1 в п.1 | из п.2 в п.2 | из п.3 в п.3 | из п. 4 в п.4 | из п. 5 в п.5 |

## **Задача №2**

Найти кратчайший путь между 5 городами (9 городами, 11 городами), расстояние между которыми известно. Рассмотреть замкнутый вариант задачи: коммивояжёру требуется посетить все города, после чего вернуться в исходный город.

Пример разобран выше для 5 городов – продублировать решение и по аналогии решить задачу для 9 и 11 городов. Расстояние подобрать самостоятельно.

## **Задача №3**

Имеется 11 городов, расстояния между которыми известны. Маршруты проложены только между некоторыми городами (неполный граф). Найти кратчайший путь между 2-мя заданными городами. Построить Линейную модель.

Реальный город с широтой и долготой

# **СПИСОК ЛИТЕРАТУРЫ**

1. Метод ветвей и границ целочисленного программирования. Основные понятия [Электронный ресурс]. URL: <https://pandia.ru/text/79/453/5828.php>
2. Е. Н. Гончаров, А. И. Ерзин, В. В. Залюбовский. Исследование операций Примеры и задачи / Учебное пособие. Новосибирск: Новосибирский государственный университет, 2005. - 78 с.
3. Математическое бюро [Электронный ресурс]. URL: <https://www.matburo.ru/>
4. StudFiles [Электронный ресурс]. URL: <https://studfiles.net/preview/2098257/>
5. Метод ветвей и границ [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/Метод\_ветвей\_и\_границ>
6. Береснев В. Л., Дементьев В. Т. Исследование операций. Введение: Учеб. пособие. Новосибирск: Изд-во НГУ, 1979.
7. Гимади Э. Х., Глебов Н. И. Экстремальные задачи принятия решений: Учеб. пособие. Новосибирск: Изд-во НГУ, 1982.
8. Гимади Э. Х., Глебов Н. И. Дискретные экстремальные задачи принятия решений: Учеб. пособие. Новосибирск: Изд-во НГУ, 1991.
9. Вентцель Е. С. Исследование операций. М.: Сов. радио, 1972.
10. Гермейер Ю. Б. Введение в теорию исследования операций. М.: Наука, 1971.