**Лабораторна робота №7**

**Тема: Задача комівояжера.**

[Історія 1](#_Toc475968793)

[Роз'вязання задачі комівояжера за методом редукції рядків і колонок 1](#_Toc475968794)

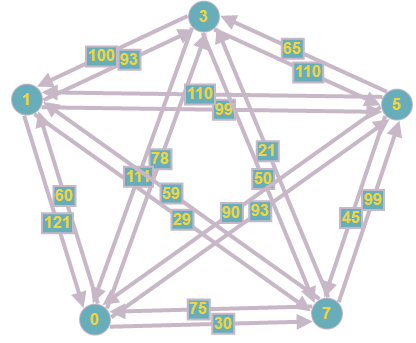
[Варіанти завдань 5](#_Toc475968795)

[Використання Microsoft Excel. 8](#_Toc475968796)

## Історія

Невідомо, коли проблему комівояжера було досліджено вперше. Однак, відома видана в 1832 році книжка з назвою «Комівояжер — як він має поводитись і що має робити для того, аби доставляти товар та мати успіх в своїх справах — поради старого Кур'єра» (нім. Der Handlungsreisende – wie er sein soll und was er zu thun hat, um Aufträge zu erhalten und eines glücklichen Erfolgs in seinen Geschäften gewiß zu sein – von einem alten Commis-Voyageur), в якій описано проблему, але математичний апарат для її розв'язання не застосовується. Натомість, в ній запропоновано приклади маршрутів для деяких регіонів Німеччини та Швейцарії.

***Гамільтон Вільям Роуен***. Раннім варіантом задачі може розглядатись англ. Icosian Game Вільяма Гамільтона 19 століття, яка полягала в тому, щоб знайти маршрути на графі з 20 вузлами. Перші згадки в якості математичної задачі на оптимізацію належать Карлу Менґеру (нім. Karl Menger), який сформулював її в математичному колоквіумі в 1930 році так: «Ми називаємо проблемою посильного (оскільки це питання виникає в кожного листоноші, зокрема, її вирішують багато мандрівників) завдання віднайти найкоротший шлях між скінченною множиною місць, відстань між якими відома.» Невдовзі з'явилась відома зараз назва задача мандруючого продавця (англ. Traveling Salesman Problem), яку запропонував Гаслер Вітні (англ. Hassler Whitney) з Принстонського Університету Разом із простотою визначення та порівняною простотою знаходження гарних розв'язків задача комівояжера відрізняється тим, що визначення насправді оптимального шляху є досить складним завданням. Зважаючи на ці властивості, починаючи з другої половини 20-го століття, дослідження задачі комівояжера, має не так практичний сенс, як теоретичний в якості моделі для розробки нових алгоритмів оптимізації. Багато сучасних поширених методів дискретної оптимізації, таких як метод діленням площиною, гілок та границь та різноманітні варіанти евристичних алгоритмів було розроблено на прикладі задачі комівояжера.

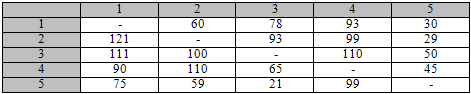


## Роз'вязання задачі комівояжера за методом редукції рядків і колонок

Процес знаходження оптимального маршруту, в задачі комівояжера, методом редукції рядків і колонок розкладається на (n-2) етапа. У межах кожного етапу алгоритм розрахунків однаковий.

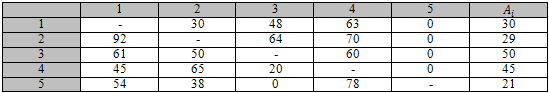
Для наглядності, розглянемо конкретний випадок задачі комівояжера, а саме, комівояжер повинен обїхати 5 міст, побувавши в кожному по одному разу і повернутися у початкове. При цьому витрати повинні бути мінімальними.

Тарифи на переміщення між містами знаходяться в наступній таблиці:



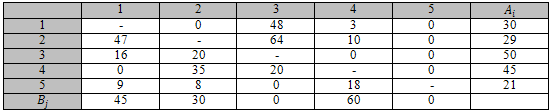
Перш ніж приступити до першого етапу, знайдемо можливу верхню межу функції мети. Для цього, вибираємо довільний маршрут metod_redukcii_radkiv_i_kolonok19, для якого обчислюємо значення функції Метод редукції рядків і колонок. Значення функції мети Метод редукції рядків і колонок повинно бути меншим за Метод редукції рядків і колонок.

Етап 1. Виконуємо редукцію рядків, тобто в кожному рядку знаходимо мінімальний елемент і віднімаємо його від елементів даного рядка. В результаті отримуємо наступну таблицю:



де стовбець Метод редукції рядків і колонокмістить значення мінімальних елементів кожного рядка.

Далі, виконуємо редукцію колонок — в кожній колонці знаходимо мінімальний елемент і віднімаємо йлго від елементів даної колонки (рядок Метод редукції рядків і колонок містить мінімальні елементи кожної колонки).



Після цього, обчислюємо найнижчу можливу межу функції мети.

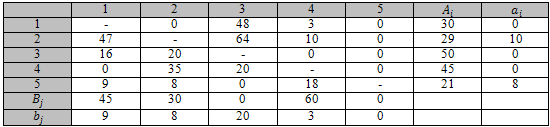
Метод редукції рядків і колонок

Тобто, функція мети Метод редукції рядків і колонок повинна знаходитьсь в межах Метод редукції рядків і колонок.

Далі провіряємо, чи у кожному рядку і колонці міститься по одному нульовому елементі. Якщо дана умова виконується, то розв'язок припиняється і нульові комірки позначають оптимальний шлях комівояжера з оптимальною функцією мети Метод редукції рядків і колонок. Якщо ж дана умова не виконується, то розв'язок продовжується, і переходимо до визначення одного з кроків оптимального шляху комівояжера. Для цього визначаємо величини metod_redukcii_radkiv_i_kolonok12 та metod_redukcii_radkiv_i_kolonok13 , які називають штрафами рядків і колонок і заносимо їх у відповідну колонку і рядок таблиці.

Штрафом i-го  рядка називається найменше значення даного рядка після першого нуля. Якщо i-й рядок містить дві чи більше нульові комірки, то Метод редукції рядків і колонок.

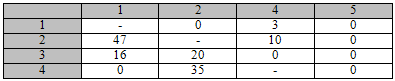
Штрафом j-їколонкиназивається найменше значення даної колонки після першого нуля. Якщо j-та колонка містить дві чи більше нульові комірки, то metod_redukcii_radkiv_i_kolonok15.



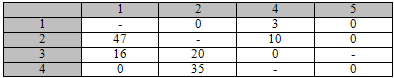
Для кожної нульової комірки визначаємо вторинні штрафи за наступною формулою: Метод редукції рядків і колонок. Після чого заносимо їх в наступну таблицю:

Метод редукції рядків і колонок

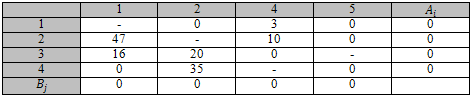
Далі, знаходимо максимальне значення серед знайдених вторинних штрафів. В нашому випадку найбільшим є Метод редукції рядків і колонок. Тобто, в шлях комівояжера потрібно внести комірку (5,3). В результаті, отримуємо нову таблицю, в якій викреслюємо з розгляду п'ятий рядок і третю колонку.



Виходячи з того, що у будь-якому рядку і у будь-якій колонці таблиці комівояжера повинна існувати одна заборонена комірка. В нашому випадку, такої комірки немає у рядку i=3 та колонці j=5. Тому забороняємо дану комірку. В результаті отримуємо нову таблицю, яку використовуємо для розрахунків на другому етапі.



Етап 2. Анілогічно першому етапу виконуємо редукцію рядків і колонок. Отримуємо:

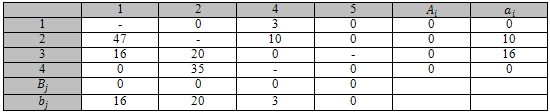


Якщо б дані Метод редукції рядків і колонок і Метод редукції рядків і колонок відрізнялися від нуля, то ми повинні були б визначити нову функцію:

Метод редукції рядків і колонок

і врахувати, що шукане значення функції мети повинно хнаходитись в межах metod_redukcii_radkiv_i_kolonok32. Виходячи з того, що в нашому випадку всі Метод редукції рядків і колонок і Метод редукції рядків і колонок рівні нулю, ми даний крок пропускаємо.

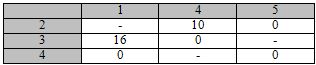
Знаходимо штрафи рядків і колонок.



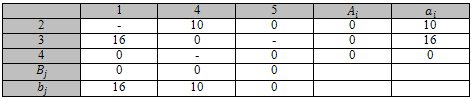
Для кожної нульової комірки визначаємо вторинні штрафи.

Метод редукції рядків і колонок

Серед знайдених значень вторинних штрафів знаходимо максимальне. Це означає, що комівояжер повинен використати на своєму шляху відвідати місто (1,2). В результаті, для третього етапу отримуємо нову таблицю, в якій відсутні рядок i=1 та колонку j=2, та комірка (2, 1) являється забороненою.



Етап 3.Виконуємо редукцію рядків і колонок, а також знаходимо штрафи рядків і колонок.



Для кожної нульової комірки визначаємо вторинні штрафи.

Метод редукції рядків і колонок

Серед знайдених значень вторинних штрафів знаходимо максимальне і заносимо до оптимального шляху комівояжера місто (3,4). В результаті ми отримали ма таблицю, яка містить два рядки і дві колонкт. На цьому алгоритм припиняється, бо дана таблиця вказує завершальний шлях комівояжера (2,5) і (4,1).

Метод редукції рядків і колонок

Таким чином, ми отримали маршрут:

Метод редукції рядків і колонок

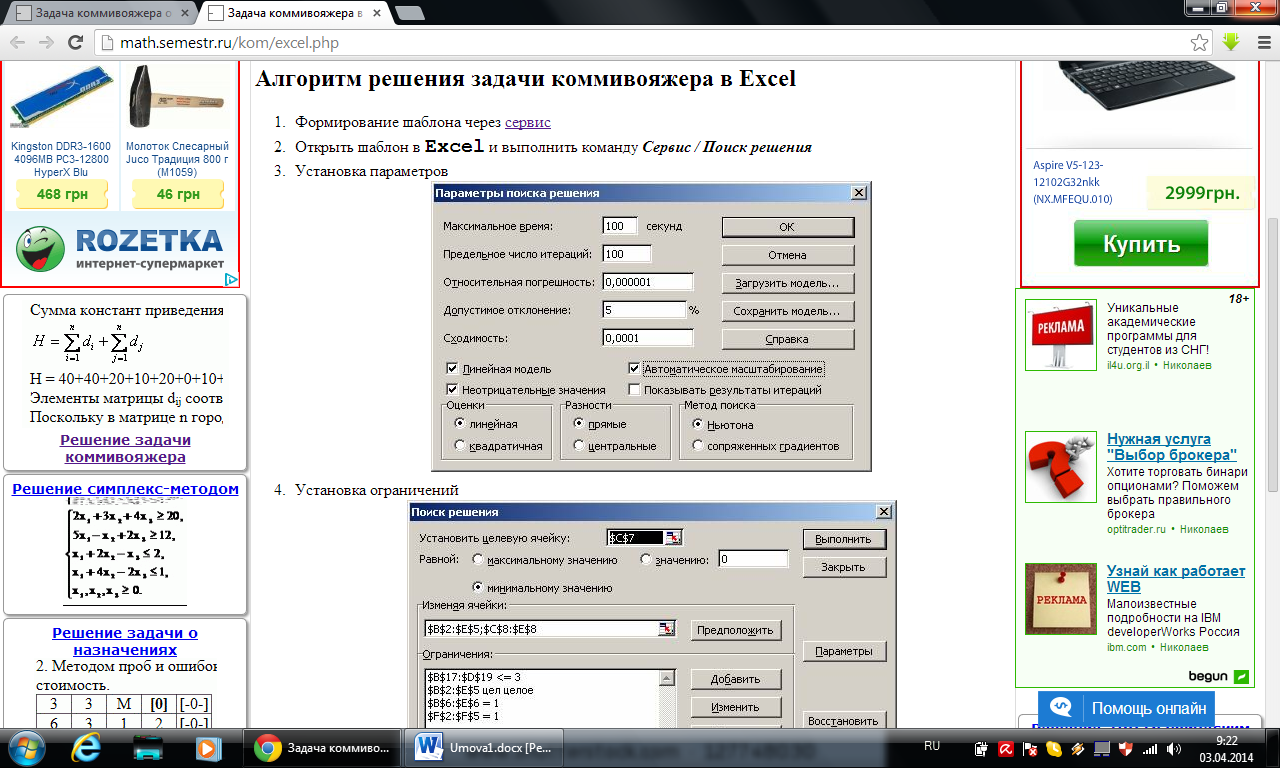
з витратами Метод редукції рядків і колонок. При цьому виконується умова Метод редукції рядків і колонок.

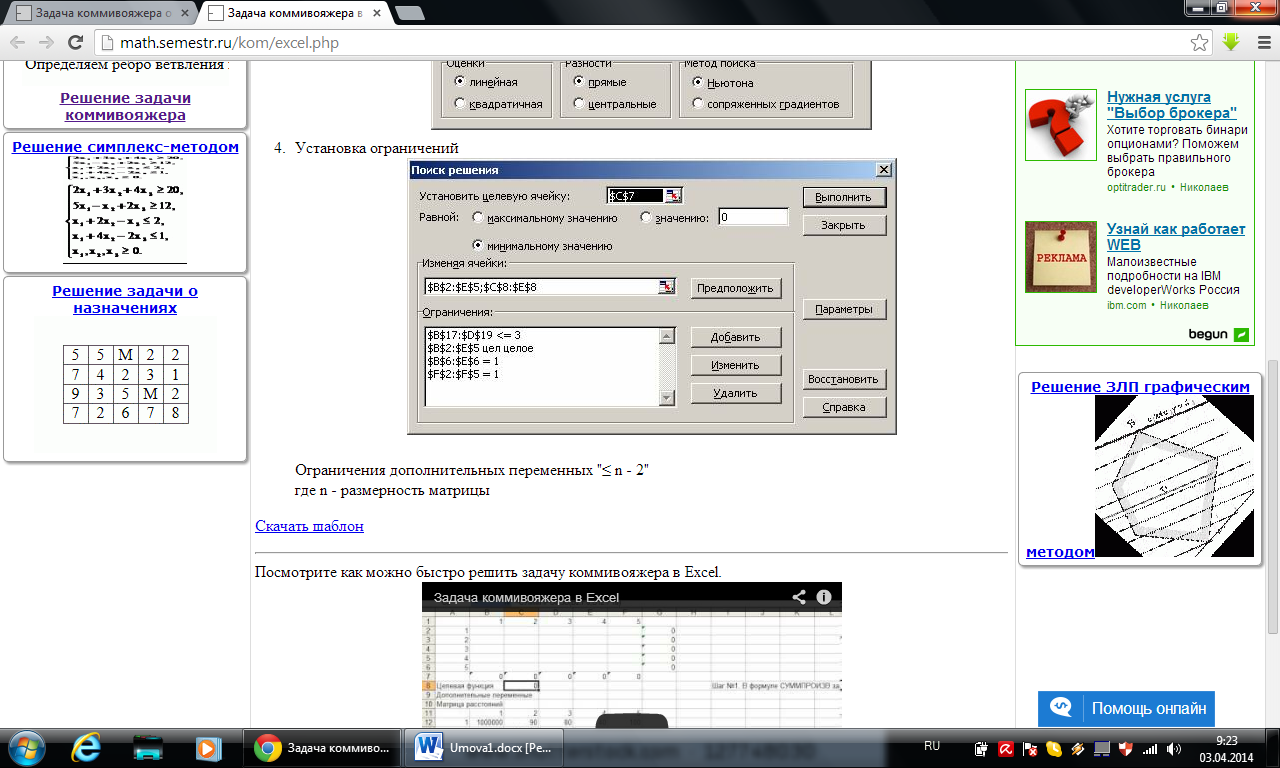
## Варіанти завдань

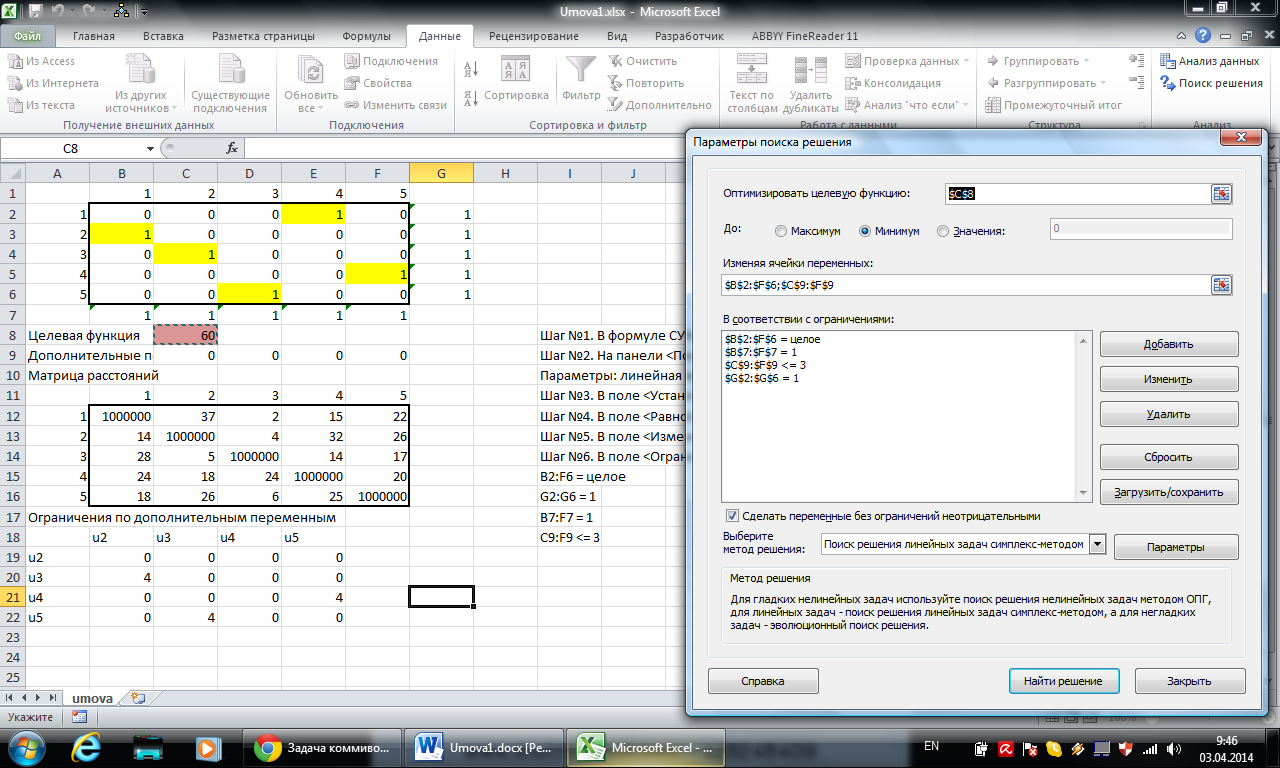
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| V1 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1000000 | 90 | 8 | 40 | 100 |
| 2 | 60 | 1000000 | 40 | 5 | 70 |
| 3 | 5 | 30 | 1000000 | 60 | 20 |
| 4 | 10 | 7 | 20 | 1000000 | 50 |
| 5 | 20 | 14 | 50 | 20 | 1000000 |
|  |  |  |  |  |  |
| V2 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1000000 | 20 | 29 | 26 | 25 |
| 2 | 3 | 1000000 | 5 | 15 | 24 |
| 3 | 19 | 2 | 1000000 | 4 | 13 |
| 4 | 20 | 6 | 1 | 1000000 | 19 |
| 5 | 11 | 11 | 11 | 11 | 1000000 |
|  |  |  |  |  |  |
| V3 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1000000 | 26 | 24 | 26 | 29 |
| 2 | 15 | 1000000 | 29 | 26 | 23 |
| 3 | 4 | 10 | 1000000 | 30 | 7 |
| 4 | 9 | 8 | 29 | 1000000 | 3 |
| 5 | 12 | 12 | 12 | 21 | 1000000 |
|  |  |  |  |  |  |
| V4 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1000000 | 22 | 2 | 13 | 7 |
| 2 | 27 | 1000000 | 4 | 24 | 9 |
| 3 | 3 | 16 | 1000000 | 5 | 4 |
| 4 | 28 | 10 | 17 | 1000000 | 29 |
| 5 | 8 | 8 | 8 | 8 | 1000000 |
|  |  |  |  |  |  |
| V5 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1000000 | 17 | 9 | 20 | 30 |
| 2 | 13 | 1000000 | 24 | 26 | 26 |
| 3 | 22 | 24 | 1000000 | 27 | 29 |
| 4 | 25 | 5 | 11 | 1000000 | 23 |
| 5 | 9 | 24 | 9 | 9 | 1000000 |
|  |  |  |  |  |  |
| V6 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1000000 | 24 | 11 | 12 | 25 |
| 2 | 26 | 1000000 | 29 | 20 | 24 |
| 3 | 27 | 14 | 1000000 | 10 | 18 |
| 4 | 6 | 15 | 28 | 1000000 | 2 |
| 5 | 15 | 15 | 15 | 15 | 1000000 |
| V7 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1000000 | 15 | 3 | 6 | 10 |
| 2 | 23 | 1000000 | 13 | 27 | 12 |
| 3 | 30 | 1 | 1000000 | 24 | 25 |
| 4 | 8 | 20 | 7 | 1000000 | 9 |
| 5 | 8 | 9 | 13 | 8 | 1000000 |
|  |  |  |  |  |  |
| V8 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1000000 | 27 | 29 | 28 | 8 |
| 2 | 13 | 1000000 | 27 | 16 | 29 |
| 3 | 20 | 30 | 1000000 | 7 | 26 |
| 4 | 11 | 15 | 30 | 1000000 | 2 |
| 5 | 7 | 14 | 7 | 14 | 1000000 |
|  |  |  |  |  |  |
| V9 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1000000 | 2 | 5 | 6 | 15 |
| 2 | 5 | 1000000 | 9 | 5 | 7 |
| 3 | 16 | 24 | 1000000 | 6 | 26 |
| 4 | 13 | 17 | 4 | 1000000 | 8 |
| 5 | 6 | 6 | 13 | 20 | 1000000 |
|  |  |  |  |  |  |
| V10 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1000000 | 11 | 25 | 17 | 21 |
| 2 | 22 | 1000000 | 14 | 8 | 1 |
| 3 | 9 | 13 | 1000000 | 28 | 15 |
| 4 | 26 | 22 | 3 | 1000000 | 27 |
| 5 | 19 | 22 | 23 | 17 | 1000000 |
|  |  |  |  |  |  |
| V11 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1000000 | 24 | 4 | 2 | 3 |
| 2 | 20 | 1000000 | 15 | 27 | 7 |
| 3 | 15 | 15 | 1000000 | 25 | 19 |
| 4 | 2 | 14 | 3 | 1000000 | 5 |
| 5 | 27 | 16 | 25 | 11 | 1000000 |
|  |  |  |  |  |  |
| V12 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1000000 | 6 | 25 | 11 | 12 |
| 2 | 13 | 1000000 | 20 | 27 | 30 |
| 3 | 16 | 7 | 1000000 | 10 | 21 |
| 4 | 1 | 16 | 23 | 1000000 | 18 |
| 5 | 11 | 22 | 31 | 6 | 1000000 |

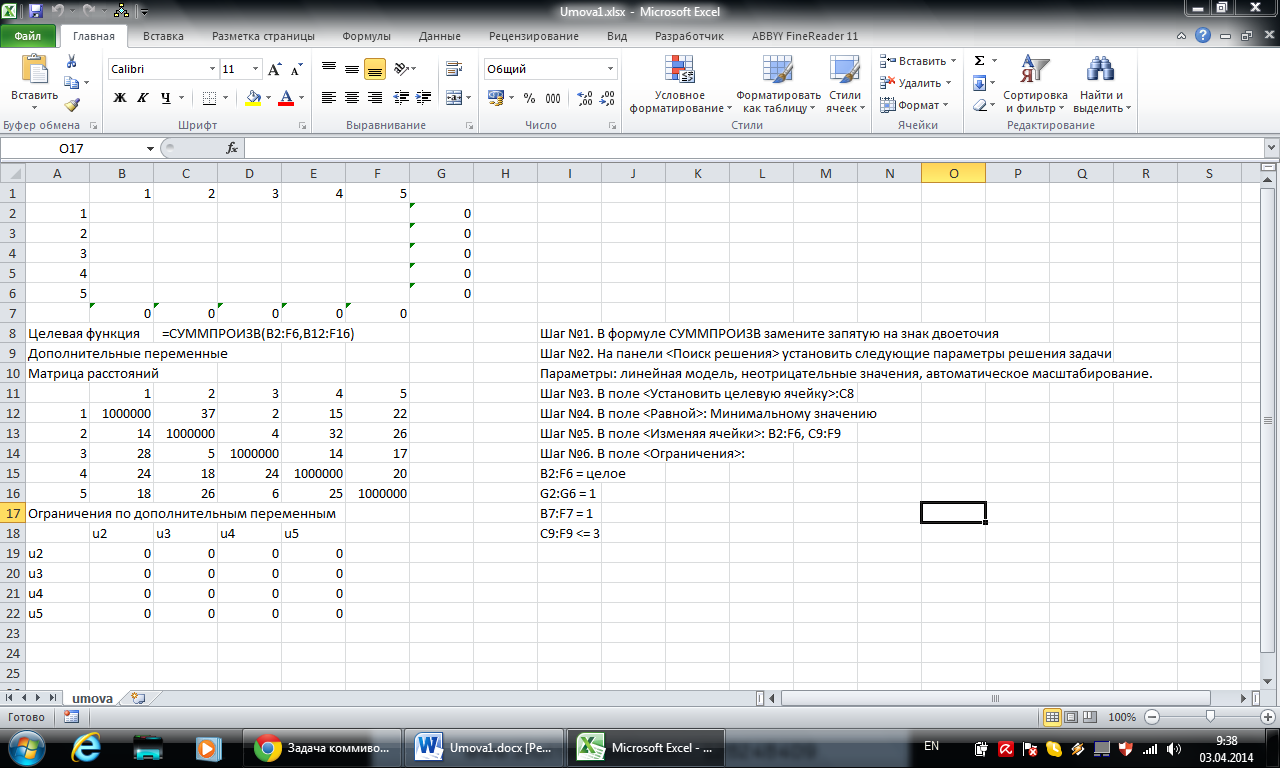
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| V13 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1000000 | 24 | 25 | 23 | 29 |
| 2 | 1 | 1000000 | 10 | 7 | 19 |
| 3 | 22 | 26 | 1000000 | 30 | 27 |
| 4 | 8 | 18 | 29 | 1000000 | 23 |
| 5 | 22 | 9 | 12 | 13 | 1000000 |
|  |  |  |  |  |  |
| V14 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1000000 | 11 | 20 | 17 | 8 |
| 2 | 1 | 1000000 | 3 | 18 | 17 |
| 3 | 9 | 39 | 1000000 | 30 | 31 |
| 4 | 23 | 17 | 4 | 1000000 | 28 |
| 5 | 10 | 8 | 12 | 14 | 1000000 |
|  |  |  |  |  |  |
| V15 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1000000 | 10 | 16 | 27 | 19 |
| 2 | 30 | 1000000 | 8 | 29 | 15 |
| 3 | 3 | 18 | 1000000 | 19 | 18 |
| 4 | 9 | 12 | 2 | 1000000 | 21 |
| 5 | 5 | 15 | 11 | 9 | 1000000 |

## Використання Microsoft Excel.









C={(1,4),(4,5),(5,3),(3,2),(2,1)}

D=15+20+6+5+14=60