

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЇВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра математики факультету інформатики

**СУЧАСНІ МЕТОДИ РОЗВ'ЯЗАННЯ
ЗАДАЧІ КОМІВОЯЖЕРА**
Курсова робота за спеціальністю “Прикладна математика”

Керівник курсової роботи

канд. фіз.-мат. наук,

доцент

Щестюк Н. Ю.

(підпис)

«____» _____ 2021р.

Виконав студент

3-го року навчання

Чирков М. В.

«____» _____ 2021р.

Київ - 2021

Тема: Сучасні методи розв'язання задачі комівояжера

Календарний план виконання роботи:

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу.	16.10.2020	
2	Отримання літератури від наукового керівника.	09.11.2020	
3.	Огляд літератури за темою роботи.	20.03.2021	
4.	Аналіз літератури за темою роботи.	01.04.2021	
5.	Дослідження результатів отриманих в літературі.	15.04.2021	
6.	Оформлення курсової роботи.	15.05.2021	
7.	Створення презентації до курсової роботи.	16.05.2021	
8.	Захист курсової роботи.	20.05.2021	

Зміст

1	Вступ	4
1.1	Актуальність	4
1.2	Постановка задачі	5
2	Розв'язання задачі	6
2.1	Математична модель задачі комівояжера	6
2.2	Алгоритм гілок та меж	8
2.3	Вирішення задачі в Excel	31
2.4	Алгоритм мурашиної колонії	35
3	Висновки	39
4	Список літератури	40

1 Вступ

1.1 Актуальність

Дана робота присвячена темі “Задача про комівояжера”. Ця задача полягає в тому, щоб визначити найкоротший або найдешевший маршрут між деякими точками на карті(містами).

Однією з перших згадок про цю задачу є задача Уільяма Гамільтона в 19 сторіччі, яка полягала в тому, щоб знайти маршрути на графі, на якому знаходилося 20 вузлів. Невдовзі Хасслер Уїтні запропонував так звану задачу мандрівного торговця(задачу комівояжера). Вона заключалася в тому, що існував деякий торговець, в якого була обмежена кількість грошей і йому було необхідно прокласти найдешевший маршрут між деякими містами на мапі. Можна побачити, що та сама задачка Уільяма Гамільтона легко перетворюється в задачу комівояжера, якщо кожному ребру графа дати деяку числову характеристику. Це може бути кілометраж, вартість шляху або кількість використаного палива. Одним з прикладів першого використання цієї задачі на практиці є оптимізація виробництва на деякому заводі в кінці 20го сторіччя, де вдалося збільшити об’єм виробництва на 15%, залишивши обладнання і кількість працівників незмінними. На даний момент задача комівояжера є широко розповсюдженою у всьому світі. Починаючи з того, що людям необхідно їздити у відрядження, подорожі і тд, а ніхто не хоче витратити зайвий час та гроші на відвідування одного й того ж самого місця декілька разів, відповідно, доводиться планувати найвигідніші, з точки зору часу та грошей, маршрути. Закінчуючи плануванням авіамаршрутів, перевезенням великогабаритних грузів з місця на місце тощо. Адже у великому бізнесі неправильно прорахований маршрут може навіть призвести до краху усієї компанії. Виходячи з усього написаного вище, можна зробити заключення, що задача комівояжера вартує того, аби бути розгляненою нами сьогодні.

1.2 Постановка задачі

Отже розглянемо задачу про комівояжера. Нехай ми маємо n міст, відстань(вартість проїзду, кількість пального) між якими нам відома. Нашому комівояжеру необхідно пройти кожне місто лише один раз і повернутися туди, звідки він почав. При цьому треба зробити це найбільш оптимальним способом – затрачена відстань(вартість проїзду, кількість пального) має бути мінімальною. Можна представити міста як вершини графа, а лінії, які з'єднують ці міста, представити дугами графа. Тоді розв'язання задачі комівояжера зводиться до знаходження гамільтонового циклу на цьому графі.

Звичайно, перший спосіб розв'язання цієї проблеми, який приходить в голову, це просто взяти усі $n!$ Можливих перестановок вершин повного графу, порахувати довжину маршруту для кожної підстановки і вибрати найкоротший варіант. Але на практиці це дуже і дуже затратно. Давайте самі це перевіримо. Давайте розглянемо випадок, де у нас є 3 міста на мапі. Щоб знайти єдиний маршрут, спочатку потрібно вибрати початкове місце з трьох можливих місць на карті. Далі у нас буде вибір з 2 міст для другого місця, тоді нарешті залишиться лише 1 місто, яке потрібно вибрати, щоб завершити наш маршрут. Це означало б, що загалом можна вибрати $3 \times 2 \times 1$ різні маршрути. Це означає, що для цього прикладу є лише 6 різних маршрутів на вибір. Тож для цього випадку лише з 3 місць досить тривіально розрахувати кожен із цих 6 маршрутів і знайти найкоротший. Кількість можливих маршрутів є фактором кількості місць, які потрібно відвідати, а проблема факторіалів полягає в тому, що вони зростають у розмірі надзвичайно швидко! А тепер розглянемо більш складний випадок. Нехай нам необхідно знайти найкоротший шлях для 20 міст на мапі. Використовуючи факторіал, нам доведеться оцінити 2432902008176640000 різних маршрутів! Навіть при сучасних обчислювальних потужностях це надзвичайно недоцільно, а для ще більших проблем це майже неможливо.

2 Розв'язання задачі

2.1 Математична модель задачі комівояжера

Можемо записати математичну модель для задачі комівояжера, ввівши булеві змінні $x_{ij} = 1$, якщо відбувається переїзд з міста i в місто j , та $x_{ij} = 0$, якщо переїзд не відбувається. Тепер маємо можливість записати систему обмежень і цільову функцію:

$$\sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min$$

$$\sum_{j=1}^n x_{ij} = 1 (i = \overline{1, n})$$

$$\sum_{i=1}^n x_{ij} = 1 (j = \overline{1, n})$$

c_{ij} – матриця, де знаходяться відстані між усіма містами, $i, j = \overline{1, n}$.

Маючи ці обмеження можемо стверджувати, що кожна змінна x_{ij} рівна нулю або одиниці. А також, що комівояжер відвідає кожне місто, прибувши до нього один раз, і відбувши його один раз. Але цих обмежень недостатньо для повної постановки задачі. Оскільки нам також необхідне обмеження, яке буде забезпечувати зв'язність кінцевого циклу. Для цього введемо ще одне обмеження:

$$u_i - u_j + nx_{ij} \leq n - 1; i, j = \overline{1, n}; i \neq j$$

u_i – змінна, яка набуває значення номеру кроку, на якому було відвідане i -те місто.

2.2 Алгоритм гілок та меж

Я буду вирішувати дану задачу за допомогою алгоритму гілок та меж(також цей метод називають алгоритмом Літтля)

Спочатку теоретично покроково сформуємо метод Літтля:

1. Проводимо операцію редукції по рядкам – знайдемо найменший елемент $d_{\min}$ в кожному рядку і віднімемо його від усіх елементів відповідного рядка. Нижньою границею в цьому випадку є: $H = \sum d_{\min}$
2. Проводимо операцію редукції по стовпцям – знайдемо найменший елемент $d_{\min}$ в кожному стовпці і віднімемо його від усіх елементів відповідного стовпця. Нижньою границею в цьому випадку є:
 $H = H + \sum d_{\min}$
3. Константа зведення H являє собою нижню границю множини можливих гамільтонових контурів.
4. Замінімо нулі в нашій матриці на знак ∞ і знайдемо суму мінімальних елементів рядка і стовпця, відповідних цьому нулю.
5. Виберемо дугу (i, j) , степінь нульового елемента для якої набуває максимального значення.
6. Розіб'ємо множину гамільтонових контурів на дві підмножини: в першій будуть ті, що містять нашу дугу (i, j) , а в другій ті, що не містять (i^*, j^*) .
7. Зведемо матриці гамільтонових контурів з пошуком констант зведення $H(i, j)$ та $H(i^*, j^*)$.
8. Порівняємо нижні границі підмножини гамільтонових контурів $H(i, j)$ та $H(i^*, j^*)$. Тепер якщо $H(i^*, j^*) > H(i, j)$ то в першу чергу подальшому розгалуженню належить множина (i, j) . Інакше ж множина (i^*, j^*) належить подальшому розбиттю.
9. Якщо отримуємо матрицю 2×2 в результаті розгалужень, то визначаємо гамільтонів контур і його довжину.
10. Порівняємо отриману довжину гамільтонового контуру з нижніми межами обірваних гілок. Задача вважається вирішеною, якщо довжина

контур не є більшою за нижні межі. Якщо ж ні, то розвиваємо гілки підмножин з нижньою межею, яка менша отриманого контуру, до тих пір, поки не отримаємо маршрут з меншою довжиною.

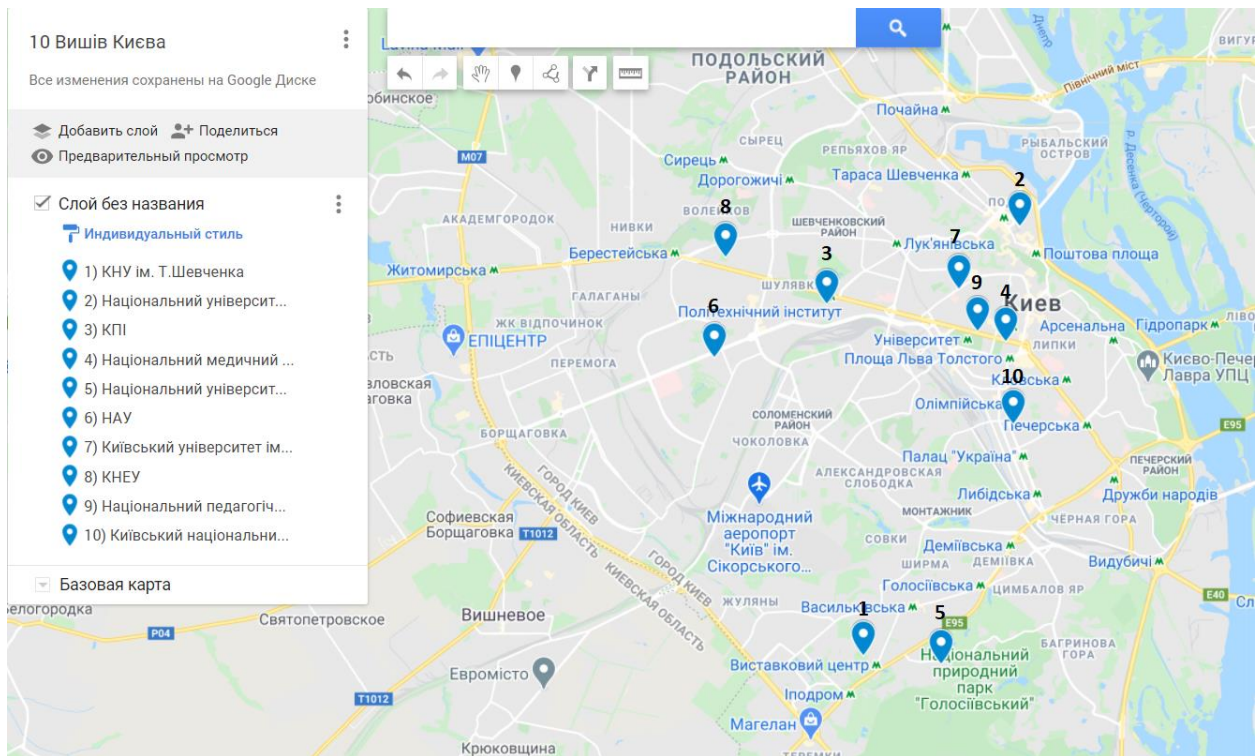
Тепер застосуємо даний алгоритм на практиці для вирішення конкретної задачі.

Нехай у нас на мапі відмічено 10 вишів Києва. Необхідно знайти найбільш вигідний маршрут між ними, відвідавши кожен лише один раз.

Ось перелік 10 Вишів, які я обрав для цієї роботи:

- 1 – Тараса Шевченка
- 2 – Могилянка
- 3 – КПІ
- 4 – Богомольця
- 5 – НУБіП
- 6 – НАУ
- 7 – Грінченка
- 8 – КНЕУ
- 9 – Драгоманова
- 10 – Лінгвістичний

За допомогою сервісу “Google Maps” нанесемо відмітки з місцезнаходженням кожного вишу на мапу, та визначимо відстань між ними.



Запишемо отримані дані в матрицю, де номер рядка чи стовбця – це номер відповідного Вишу. Отримаємо таку матрицю D:

i, j	1	2	3	4	5	6	7	8	9	10
1	∞	16	12	10	4	14	12	12	11	8
2	15	∞	6	3	14	10	3	8	4	12
3	11	6	∞	4	13	5	4	3	4	7
4	11	4	5	∞	9	11	2	7	1	3
5	4	15	12	9	∞	15	11	14	9	7
6	11	9	4	7	13	∞	6	4	6	8
7	11	3	4	2	11	9	∞	5	2	5
8	12	10	6	8	14	5	6	∞	10	10
9	11	4	4	2	10	9	1	6	∞	4
10	8	5	6	2	7	11	4	8	3	∞

Візьмемо будь-який маршрут. Наприклад:

$$X_0 = (1,2);(2,3);(3,4);(4,5);(5,6);(6,7);(7,8);(8,9);(9,10);(10,1)$$

$$\text{Відповідно } F(X_0) = 16 + 6 + 4 + 9 + 15 + 6 + 5 + 10 + 4 + 8 = 83$$

Для визначення нижньої границі скористаємося операцією редукції, для чого знайдемо мінімальний елемент в кожному рядку матриці D.

$$d_i = \min(j) d_{ij}$$

i, j	1	2	3	4	5	6	7	8	9	10	d_i
1	∞	16	12	10	4	14	12	12	11	8	4
2	15	∞	6	3	14	10	3	8	4	12	3
3	11	6	∞	4	13	5	4	3	4	7	3
4	11	4	5	∞	9	11	2	7	1	3	1
5	4	15	12	9	∞	15	11	14	9	7	4
6	11	9	4	7	13	∞	6	4	6	8	4
7	11	3	4	2	11	9	∞	5	2	5	2
8	12	10	6	8	14	5	6	∞	10	10	5
9	11	4	4	2	10	9	1	6	∞	4	1
10	8	5	6	2	7	11	4	8	3	∞	2

Після цього відніmemo d_i від елементів певного рядка. Таким чином в новій матриці матимемо принаймні один нуль в кожному рядку:

i, j	1	2	3	4	5	6	7	8	9	10
1	∞	12	8	6	0	10	8	8	7	4
2	12	∞	3	0	11	7	0	5	1	9
3	8	3	∞	1	10	2	1	0	1	4
4	11	3	4	∞	8	10	1	6	0	2
5	0	11	8	5	∞	11	7	10	5	3
6	7	5	0	3	9	∞	2	0	2	4
7	9	1	2	0	9	7	∞	3	0	3
8	7	5	1	3	9	0	1	∞	5	5
9	10	3	3	1	9	8	0	5	∞	3
10	6	4	4	0	5	9	2	6	1	∞

Таку ж саму операцію редукції проводимо по стовпцям, для чого знайдемо мінімальний елемент в кожному стовпці матриці D.

$$d_j = \min(i) d_{ij}$$

i, j	1	2	3	4	5	6	7	8	9	10
1	∞	12	8	6	0	10	8	8	7	4
2	12	∞	3	0	11	7	0	5	1	9
3	8	3	∞	1	10	2	1	0	1	4
4	11	3	4	∞	8	10	1	6	0	2
5	0	11	8	5	∞	11	7	10	5	3
6	7	5	0	3	9	∞	2	0	2	4

7	9	1	2	0	9	7	∞	3	0	3
8	7	5	1	3	9	0	1	∞	5	5
9	10	3	3	1	9	8	0	5	∞	3
10	6	3	4	0	5	9	2	6	1	∞
d_j	0	1	0	0	0	0	0	0	0	2

Після віднімання мінімальних елементів отримаємо повністю скорочену матрицю, в якій величини d_i і d_j називаються константами зведення.

i, j	1	2	3	4	5	6	7	8	9	10
1	∞	11	8	6	0	10	8	8	7	2
2	12	∞	3	0	11	7	0	5	1	7
3	8	2	∞	1	10	2	1	0	1	2
4	11	2	4	∞	8	10	1	6	0	0
5	0	10	8	5	∞	11	7	10	5	1
6	7	4	0	3	9	∞	2	0	2	2
7	9	0	2	0	9	7	∞	3	0	1
8	7	4	1	3	9	0	1	∞	5	3
9	10	2	3	1	9	8	0	5	∞	1
10	6	2	4	0	5	9	2	6	1	∞

За допомогою суми констант зведення визначимо нижню границю H:

$$H = \sum d_i + \sum d_j$$

$$H = 4 + 3 + 3 + 1 + 4 + 4 + 2 + 5 + 1 + 2 + 0 + 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 2 = 32$$

Елементи матриці d_{ij} відповідають відстані від пункта i до пункта j . Оскільки в матриці n міст, D є матрицею $n \times n$ з невід'ємними коефіцієнтами $d_{ij} \geq 0$. Кожен можливий маршрут являє собою цикл, по якому комівояжер відвідує місто лише один раз і повертається в початкове місто. Довжину маршруту можна порахувати по формулі: $F(M_k) = \sum d_{ij}$. Причому кожний рядок і стовбець входять в маршрут тільки один раз з елементом d_{ij} .

Крок №1

Визначимо ребро гілкування і розіб'ємо всю множину маршрутів відносно цього ребра на дві підмножини (i, j) та (i^*, j^*) . Відповідно заміняємо усі клітинки матриці з нульовими елементами на ∞ і визначаємо для них суму констант зведення(можна побачити в дужках).

i, j	1	2	3	4	5	6	7	8	9	10	d_i
1	∞	11	8	6	0(7)	10	8	8	7	2	2
2	12	∞	3	0(0)	11	7	0(0)	5	1	7	0
3	8	2	∞	1	10	2	1	0(1)	1	2	1
4	11	2	4	∞	8	10	1	6	0(0)	0(1)	0
5	0(7)	10	8	5	∞	11	7	10	5	1	1
6	7	4	0(1)	3	9	∞	2	0(0)	2	2	0
7	9	0(2)	2	0(0)	9	7	∞	3	0(0)	1	0
8	7	4	1	3	9	0(3)	1	∞	5	3	1
9	10	2	3	1	9	8	0(1)	5	∞	1	1
10	6	2	4	0(1)	5	9	2	6	1	∞	1
d_j	6	2	1	0	5	2	0	0	0	1	0

$d(1,5) = 2 + 5 = 7$; $d(2,4) = 0 + 0 = 0$; $d(2,7) = 0 + 0 = 0$; $d(3,8) = 1 + 0 = 1$; $d(4,9) = 0 + 0 = 0$; $d(4,10) = 0 + 1 = 1$; $d(5,1) = 1 + 6 = 7$; $d(6,3) = 0 + 1 = 1$; $d(6,8) = 0 + 0 = 0$; $d(7,2) = 0 + 2 = 2$; $d(7,4) = 0 + 0 = 0$; $d(7,9) = 0 + 0 = 0$; $d(8,6) = 1 + 2 = 3$; $d(9,7) = 1 + 0 = 1$; $d(10,4) = 1 + 0 = 1$

Найбільша сума констант зведення дорівнює $(2+5) = 7$ для ребра $(1, 5)$.

Відповідно множина розбивається на дві підмножини $(1,5)$ і $(1^*,5^*)$.

Замінюємо елемент $d_{15} = 0$ на ∞ , тим самим провівши виключення ребра $(1,5)$.

Після чого робимо зведення матриці відстаней для отриманої підмножини $(1^*,5^*)$, в результаті отримаємо скорочену матрицю.

i, j	1	2	3	4	5	6	7	8	9	10	d_i
1	∞	11	8	6	∞	10	8	8	7	2	2
2	12	∞	3	0	11	7	0	5	1	7	0
3	8	2	∞	1	10	2	1	0	1	2	0
4	11	2	4	∞	8	10	1	6	0	0	0
5	0	10	8	5	∞	11	7	10	5	1	0
6	7	4	0	3	9	∞	2	0	2	2	0
7	9	0	2	0	9	7	∞	3	0	1	0
8	7	4	1	3	9	0	1	∞	5	3	0
9	10	2	3	1	9	8	0	5	∞	1	0
10	6	2	4	0	5	9	2	6	1	∞	0
d_j	0	0	0	0	5	0	0	0	0	0	7

Нижня границя гамільтонових циклів цієї підмножини: $H(1^*,5^*) = 32+7=39$.

Включення ребра (1,5) проводиться шляхом виключення всіх елементів першого рядка і п'ятого стовбця, в якій елемент d_{51} замінюємо на ∞ , щоб виключити можливість утворення негамільтонового циклу. В результаті отримаємо іншу скорочену матрицю, яка підлягає операції зведення. Після зведення скорочена матриця буде виглядати так:

i, j	1	2	3	4	6	7	8	9	10	d_i
2	12	∞	3	0	7	0	5	1	7	0
3	8	2	∞	1	2	1	0	1	2	0
4	11	2	4	∞	10	1	6	0	0	0
5	∞	10	8	5	11	7	10	5	1	1
6	7	4	0	3	∞	2	0	2	2	0
7	9	0	2	0	7	∞	3	0	1	0
8	7	4	1	3	0	1	∞	5	3	0
9	10	2	3	1	8	0	5	∞	1	0
10	6	2	4	0	9	2	6	1	∞	0
d_j	6	0	0	0	0	0	0	0	0	7

Сума констант зведення скороченої матриці: $\sum d_i + \sum d_j = 7$

Нижня границя підмножини (1,5) дорівнює: $H(1,5) = 32+7 = 39 \leq 39$

Оскільки нижні границі підмножини (1,5) і (1*,5*) рівні, то включаємо ребро (1,5) в маршрут з новою границею $H = 39$.

Крок №2

Визначимо ребро гілкування і розіб'ємо всю множину маршрутів відносно цього ребра на дві підмножини (i, j) та (i*, j*). Відповідно заміняємо усі

клітинки матриці з нульовими елементами на ∞ і визначаємо для них суму констант зведення(можна побачити в дужках).

i, j	1	2	3	4	6	7	8	9	10	d_i
2	6	∞	3	0(0)	7	0(0)	5	1	7	0
3	2	2	∞	1	2	1	0(1)	1	2	1
4	4	2	4	∞	10	1	6	0(0)	0(0)	0
5	∞	9	7	4	10	6	9	4	0(4)	4
6	1	4	0(1)	3	∞	2	0(0)	2	2	0
7	3	0(2)	2	0(0)	7	∞	3	0(0)	1	0
8	1	4	1	3	0(3)	1	∞	5	3	1
9	4	2	3	1	8	0(1)	5	∞	1	1
10	0(1)	2	4	0(0)	9	2	6	1	∞	0
d_j	1	2	1	0	2	0	0	0	0	0

$d(2,4) = 0 + 0 = 0$; $d(2,7) = 0 + 0 = 0$; $d(3,8) = 1 + 0 = 1$; $d(4,9) = 0 + 0 = 0$; $d(4,10) = 0 + 0 = 0$; $d(5,10) = 4 + 0 = 4$; $d(6,3) = 0 + 1 = 1$; $d(6,8) = 0 + 0 = 0$; $d(7,2) = 0 + 2 = 2$; $d(7,4) = 0 + 0 = 0$; $d(7,9) = 0 + 0 = 0$; $d(8,6) = 1 + 2 = 3$; $d(9,7) = 1 + 0 = 1$; $d(10,1) = 0 + 1 = 1$; $d(10,4) = 0 + 0 = 0$;

Найбільша сума констант зведення дорівнює $(4+0) = 4$ для ребра $(5, 10)$.

Відповідно множина розбивається на дві підмножини $(5,10)$ і $(5^*,10^*)$.

Замінюємо елемент $d_{510} = 0$ на ∞ , тим самим провівши виключення ребра $(5,10)$.

Після чого робимо зведення матриці відстаней для отриманої підмножини $(5^*,10^*)$, в результаті отримаємо скорочену матрицю.

i, j	1	2	3	4	6	7	8	9	10	d_i
------	---	---	---	---	---	---	---	---	----	-------

2	6	∞	3	0	7	0	5	1	7	0
3	2	2	∞	1	2	1	0	1	2	0
4	4	2	4	∞	10	1	6	0	0	0
5	∞	9	7	4	10	6	9	4	∞	4
6	1	4	0	3	∞	2	0	2	2	0
7	3	0	2	0	7	∞	3	0	1	0
8	1	4	1	3	0	1	∞	5	3	0
9	4	2	3	1	8	0	5	∞	1	0
10	0	2	4	0	9	2	6	1	∞	0
d_j	0	0	0	0	0	0	0	0	0	4

Нижня границя гамільтонових циклів цієї підмножини: $H(5^*, 10^*) = 39 + 4 = 43$.

Включення ребра (5,10) проводиться шляхом виключення всіх елементів п'ятого рядка і десятого стовбця, в якій елемент d_{105} замінюємо на ∞ , щоб виключити можливість утворення негамільтонового циклу. В результаті отримаємо іншу скорочену матрицю, яка підлягає операції зведення. Після зведення скорочена матриця буде виглядати так:

i, j	1	2	3	4	6	7	8	9	d_i
2	6	∞	3	0	7	0	5	1	0
3	2	2	∞	1	2	1	0	1	0
4	4	2	4	∞	10	1	6	0	0
6	1	4	0	3	∞	2	0	2	0
7	3	0	2	0	7	∞	3	0	0

8	1	4	1	3	0	1	∞	5	0
9	4	2	3	1	8	0	5	∞	0
10	0	2	4	0	9	2	6	1	0
d_j	0	0	0	0	0	0	0	0	0

Сума констант зведення скороченої матриці: $\sum d_i + \sum d_j = 0$

Нижня границя підмножини (5,10) дорівнює: $H(5,10) = 39+0 = 39 \leq 43$

Щоб виключити підцикли, заборонимо перехід (10,1).

Оскільки нижня границя підмножини(5,10) менша за $(5^*, 10^*)$, то ребро (5,10) включаємо в маршрут з новою границею $H = 39$

Крок №3

Визначимо ребро гілкування і розіб'ємо всю множину маршрутів відносно цього ребра на дві підмножини (i, j) та (i^*, j^*) . Відповідно заміняємо усі клітинки матриці з нульовими елементами на ∞ і визначаємо для них суму констант зведення(можна побачити в дужках).

i, j	1	2	3	4	6	7	8	9	d_i
2	5	∞	3	0(0)	7	0(0)	5	1	0
3	1	2	∞	1	2	1	0(1)	1	1
4	3	2	4	∞	10	1	6	0(1)	1
6	0(0)	4	0(1)	3	∞	2	0(0)	2	0
7	2	0(2)	2	0(0)	7	∞	3	0(0)	0
8	0(0)	4	1	3	0(2)	1	∞	5	0

9	3	2	3	1	8	0(1)	5	∞	1
10	∞	2	4	0(1)	9	2	6	1	1
d_j	0	2	1	0	2	0	0	0	0

$d(2,4) = 0 + 0 = 0$; $d(2,7) = 0 + 0 = 0$; $d(3,8) = 1 + 0 = 1$; $d(4,9) = 1 + 0 = 1$; $d(6,1) = 0 + 0 = 0$; $d(6,3) = 0 + 1 = 1$; $d(6,8) = 0 + 0 = 0$; $d(7,2) = 0 + 2 = 2$; $d(7,4) = 0 + 0 = 0$; $d(7,9) = 0 + 0 = 0$; $d(8,1) = 0 + 0 = 0$; $d(8,6) = 0 + 2 = 2$; $d(9,7) = 1 + 0 = 1$; $d(10,4) = 1 + 0 = 1$;

Найбільша сума констант зведення дорівнює $(0+2) = 2$ для ребра $(7, 2)$.

Відповідно множина розбивається на дві підмножини $(7,2)$ і $(7^*, 2^*)$.

Замінюємо елемент $d_{72} = 0$ на ∞ , тим самим провівши виключення ребра $(7,2)$.

Після чого робимо зведення матриці відстаней для отриманої підмножини $(7^*, 2^*)$, в результаті отримаємо скорочену матрицю.

i, j	1	2	3	4	6	7	8	9	d_i
2	5	∞	3	0	7	0	5	1	0
3	1	2	∞	1	2	1	0	1	0
4	3	2	4	∞	10	1	6	0	0
6	0	4	0	3	∞	2	0	2	0
7	2	∞	2	0	7	∞	3	0	0
8	0	4	1	3	0	1	∞	5	0
9	3	2	3	1	8	0	5	∞	0
10	∞	2	4	0	9	2	6	1	0
d_j	0	2	0	0	0	0	0	0	2

Нижня границя гамільтонових циклів цієї підмножини: $H(7^*, 2^*) = 39 + 2 = 41$.

Включення ребра $(7, 2)$ проводиться шляхом виключення всіх елементів сьомого рядка і другого стовбця, в якій елемент d_{27} замінюємо на ∞ , щоб виключити можливість утворення негамільтонового циклу. В результаті отримаємо іншу скорочену матрицю, яка підлягає операції зведення. Після зведення скорочена матриця буде виглядати так:

i, j	1	3	4	6	7	8	9	d_i
2	5	3	0	7	0	5	1	0
3	1	∞	1	2	1	0	1	0
4	3	4	∞	10	1	6	0	0
6	0	0	3	∞	2	0	2	0
8	0	1	3	0	1	∞	5	0
9	3	3	1	8	0	5	∞	0
10	∞	4	0	9	2	6	1	0
d_j	0	0	0	0	0	0	0	0

Сума констант зведення скороченої матриці: $\sum d_i + \sum d_j = 0$

Нижня границя підмножини $(7, 2)$ дорівнює: $H(7, 2) = 39 + 0 = 39 \leq 41$

Щоб виключити підцикли, заборонимо перехід $(10, 1)$.

Оскільки нижня границя підмножини $(7, 2)$ менша за $(7^*, 2^*)$, то ребро $(7, 2)$ включаємо в маршрут з новою границею $H = 39$

Крок №4

Визначимо ребро гілкування і розіб'ємо всю множину маршрутів відносно цього ребра на дві підмножини (i, j) та (i^*, j^*) . Відповідно заміняємо усі клітинки матриці з нульовими елементами на ∞ і визначаємо для них суму констант зведення(можна побачити в дужках).

i, j	1	3	4	6	7	8	9	d_i
2	5	3	0(1)	7	∞	5	1	1
3	1	∞	1	2	1	0(1)	1	1
4	3	4	∞	10	1	6	0(2)	1
6	0(0)	0(1)	3	∞	2	0(0)	2	0
8	0(0)	1	3	0(2)	1	∞	5	0
9	3	3	1	8	0(2)	5	∞	1
10	∞	4	0(1)	9	2	6	1	1
d_j	0	1	0	2	1	0	1	0

$d(2,4) = 1 + 0 = 1$; $d(3,8) = 1 + 0 = 1$; $d(4,9) = 1 + 1 = 2$; $d(6,1) = 0 + 0 = 0$; $d(6,3) = 0 + 1 = 1$; $d(6,8) = 0 + 0 = 0$; $d(8,1) = 0 + 0 = 0$; $d(8,6) = 0 + 2 = 2$; $d(9,7) = 1 + 1 = 2$; $d(10,4) = 1 + 0 = 1$;

Найбільша сума констант зведення дорівнює $(1+1) = 2$ для ребра $(4, 9)$.

Відповідно множина розбивається на дві підмножини $(4,9)$ і $(4^*, 9^*)$.

Замінюємо елемент $d_{49} = 0$ на ∞ , тим самим провівши виключення ребра $(4,9)$.

Після чого робимо зведення матриці відстаней для отриманої підмножини $(4^*, 9^*)$, в результаті отримаємо скорочену матрицю.

i, j	1	3	4	6	7	8	9	d_i
2	5	3	0	7	∞	5	1	0

3	1	∞	1	2	1	0	1	0
4	3	4	∞	10	1	6	∞	1
6	0	0	3	∞	2	0	2	0
8	0	1	3	0	1	∞	5	0
9	3	3	1	8	0	5	∞	0
10	∞	4	0	9	2	6	1	0
d_j	0	0	0	0	0	0	1	2

Нижня границя гамільтонових циклів цієї підмножини: $H(4^*, 9^*) = 39 + 2 = 41$.

Включення ребра (4,9) проводиться шляхом виключення всіх елементів четвертого рядка і дев'ятого стовбця, в якій елемент d_{94} замінюємо на ∞ , щоб виключити можливість утворення негамільтонового циклу. В результаті отримаємо іншу скорочену матрицю, яка підлягає операції зведення. Після зведення скорочена матриця буде виглядати так:

i, j	1	3	4	6	7	8	d_i
2	5	3	0	7	∞	5	0
3	1	∞	1	2	1	0	0
6	0	0	3	∞	2	0	0
8	0	1	3	0	1	∞	0
9	3	3	1	8	0	5	0
10	∞	4	0	9	2	6	0
d_j	0	0	0	0	0	0	0

Сума констант зведення скороченої матриці: $\sum d_i + \sum d_j = 0$

Нижня границя підмножини (4,9) дорівнює: $H(4,9) = 39+0 = 39 \leq 41$

Щоб виключити підцикли, заборонимо перехід (10,1).

Оскільки нижня границя підмножини(4,9) менша за $(4^*,9^*)$, то ребро (4,9) включаємо в маршрут з новою границею $H = 39$

Крок №5

Визначимо ребро гілкування і розіб'ємо всю множину маршрутів відносно цього ребра на дві підмножини (i, j) та (i^*, j^*) . Відповідно заміняємо усі клітинки матриці з нульовими елементами на ∞ і визначаємо для них суму констант зведення(можна побачити в дужках).

i, j	1	3	4	6	7	8	d_i
2	5	3	0(3)	7	∞	5	3
3	1	∞	1	2	1	0(1)	1
6	0(0)	0(1)	3	∞	2	0(0)	0
8	0(0)	1	3	0(2)	1	∞	0
9	3	3	∞	8	0(4)	5	3
10	∞	4	0(2)	9	2	6	2
d_j	0	1	0	2	1	0	0

$d(2,4) = 3 + 0 = 3$; $d(3,8) = 1 + 0 = 1$; $d(6,1) = 0 + 0 = 0$; $d(6,3) = 0 + 1 = 1$; $d(6,8) = 0 + 0 = 0$; $d(8,1) = 0 + 0 = 0$; $d(8,6) = 0 + 2 = 2$; $d(9,7) = 3 + 1 = 4$; $d(10,4) = 2 + 0 = 2$;

Найбільша сума констант зведення дорівнює $(3+1) = 4$ для ребра (9, 7).

Відповідно множина розбивається на дві підмножини (9,7) і $(9^*,7^*)$.

Замінюємо елемент $d_{97} = 0$ на ∞ , тим самим провівши виключення ребра(4,9).

Після чого робимо зведення матриці відстаней для отриманої підмножини $(9^*, 7^*)$, в результаті отримаємо скорочену матрицю.

i, j	1	3	4	6	7	8	d_i
2	5	3	0	7	∞	5	0
3	1	∞	1	2	1	0	0
6	0	0	3	∞	2	0	0
8	0	1	3	0	1	∞	0
9	3	3	∞	8	∞	5	3
10	∞	4	0	9	2	6	0
d_j	0	0	0	0	1	0	4

Нижня границя гамільтонових циклів цієї підмножини: $H(9^*, 7^*) = 39 + 4 = 43$.

Включення ребра $(9, 7)$ проводиться шляхом виключення всіх елементів дев'ятого рядка і сьомого стовбця, в якій елемент d_{79} замінюємо на ∞ , щоб виключити можливість утворення негамільтонового циклу. В результаті отримаємо іншу скорочену матрицю, яка підлягає операції зведення. Після зведення скорочена матриця буде виглядати так:

i, j	1	3	4	6	8	d_i
2	5	3	0	7	5	0
3	1	∞	1	2	0	0
6	0	0	3	∞	0	0
8	0	1	3	0	∞	0

10	∞	4	0	9	6	0
d_j	0	0	0	0	0	0

Сума констант зведення скороченої матриці: $\sum d_i + \sum d_j = 0$

Нижня границя підмножини (9,7) дорівнює: $H(9,7) = 39+0 = 39 \leq 43$

Щоб виключити підцикли, заборонимо наступні переходи (10,1),(2,4),(2,9).

Оскільки нижня границя підмножини(9,7) менша за $(9^*,7^*)$, то ребро (9,7) включаємо в маршрут з новою границею $H = 39$

Крок №6

Визначимо ребро гілкування і розіб'ємо всю множину маршрутів відносно цього ребра на дві підмножини (i, j) та (i^*, j^*) . Відповідно заміняємо усі клітинки матриці з нульовими елементами на ∞ і визначаємо для них суму констант зведення(можна побачити в дужках).

i, j	1	3	4	6	8	d_i
2	2	0(2)	∞	4	2	2
3	1	∞	1	2	0(1)	1
6	0(0)	0(0)	3	∞	0(0)	0
8	0(0)	1	3	0(2)	∞	0
10	∞	4	0(5)	9	6	4
d_j	0	0	1	2	0	0

$d(2,3) = 2 + 0 = 2$; $d(3,8) = 1 + 0 = 1$; $d(6,1) = 0 + 0 = 0$; $d(6,3) = 0 + 0 = 0$; $d(6,8) = 0 + 0 = 0$; $d(8,1) = 0 + 0 = 0$; $d(8,6) = 0 + 2 = 2$; $d(10,4) = 4 + 1 = 5$;

Найбільша сума констант зведення дорівнює $(4+1) = 5$ для ребра $(10, 4)$.

Відповідно множина розбивається на дві підмножини $(10,4)$ і $(10^*,4^*)$.

Замінюємо елемент $d_{104} = 0$ на ∞ , тим самим провівши виключення ребра $(10,4)$.

Після чого робимо зведення матриці відстаней для отриманої підмножини $(10^*,4^*)$, в результаті отримаємо скорочену матрицю.

i, j	1	3	4	6	8	d_i
2	2	0	∞	4	2	0
3	1	∞	1	2	0	0
6	0	0	3	∞	0	0
8	0	1	3	0	∞	0
10	∞	4	∞	9	6	4
d_j	0	0	1	0	0	5

Нижня границя гамільтонових циклів цієї підмножини: $H(10^*,4^*) = 39+5=44$.

Включення ребра $(10,4)$ проводиться шляхом виключення всіх елементів десятого рядка і четвертого стовбця, в якій елемент d_{410} замінюємо на ∞ , щоб виключити можливість утворення негамільтонового циклу. В результаті отримаємо іншу скорочену матрицю, яка підлягає операції зведення. Після зведення скорочена матриця буде виглядати так:

i, j	1	3	6	8	d_i
2	2	0	4	2	0
3	1	∞	2	0	0
6	0	0	∞	0	0
8	0	1	0	∞	0

d_j	0	0	0	0	0
-------	---	---	---	---	---

Сума констант зведення скороченої матриці: $\sum d_i + \sum d_j = 0$

Нижня границя підмножини (10,4) дорівнює: $H(10,4) = 39+0 = 39 \leq 44$

Щоб виключити підцикли, заборонимо наступні переходи

(2,1),(2,5),(2,9),(2,4),(2,10)

Оскільки нижня границя підмножини(10,4) менша за $(10^*,4^*)$, то ребро (10,4) включаємо в маршрут з новою границею $H = 39$

Крок №7

Визначимо ребро гілкування і розіб'ємо всю множину маршрутів відносно цього ребра на дві підмножини (i, j) та (i^*, j^*) . Відповідно заміняємо усі клітинки матриці з нульовими елементами на ∞ і визначаємо для них суму констант зведення(можна побачити в дужках).

i, j	1	3	6	8	d_i
2	∞	0(2)	4	2	2
3	1	∞	2	0(1)	1
6	0(0)	0(0)	∞	0(0)	0
8	0(0)	1	0(2)	∞	0
d_j	0	0	2	0	0

$d(2,3) = 2 + 0 = 2$; $d(3,8) = 1 + 0 = 1$; $d(6,1) = 0 + 0 = 0$; $d(6,3) = 0 + 0 = 0$; $d(6,8) = 0 + 0 = 0$; $d(8,1) = 0 + 0 = 0$; $d(8,6) = 0 + 2 = 2$;

Найбільша сума констант зведення дорівнює $(2+0) = 2$ для ребра (2, 3).

Відповідно множина розбивається на дві підмножини (2,3) і $(2^*,3^*)$.

Замінюємо елемент $d_{23} = 0$ на ∞ , тим самим провівши виключення ребра(2,3).

Після чого робимо зведення матриці відстаней для отриманої підмножини $(2^*, 3^*)$, в результаті отримаємо скорочену матрицю.

i, j	1	3	6	8	d_i
2	∞	∞	4	2	2
3	1	∞	2	0	0
6	0	0	∞	0	0
8	0	1	0	∞	0
d_j	0	0	0	0	2

Нижня границя гамільтонових циклів цієї підмножини: $H(2^*, 3^*) = 39 + 2 = 41$.

Включення ребра (2,3) проводиться шляхом виключення всіх елементів другого рядка і третього стовбця, в якій елемент d_{32} замінюємо на ∞ , щоб виключити можливість утворення негамільтонового циклу. В результаті отримаємо іншу скорочену матрицю, яка підлягає операції зведення. Після зведення скорочена матриця буде виглядати так:

i, j	1	6	8	d_i
3	1	2	0	0
6	0	∞	0	0
8	0	0	∞	0
d_j	0	0	0	0

Сума констант зведення скороченої матриці: $\sum d_i + \sum d_j = 0$

Нижня границя підмножини (2,3) дорівнює: $H(2,3) = 39 + 0 = 39 \leq 41$

Щоб виключити підцикли, заборонимо наступні переходи

$(3,1), (3,5), (3,9), (3,4), (3,10), (3,7)$

Оскільки нижня границя підмножини $(2,3)$ менша за $(2^*, 3^*)$, то ребро $(2,3)$ включаємо в маршрут з новою границею $H = 39$

Крок №8

Визначимо ребро гілкування і розіб'ємо всю множину маршрутів відносно цього ребра на дві підмножини (i, j) та (i^*, j^*) . Відповідно заміняємо усі клітинки матриці з нульовими елементами на ∞ і визначаємо для них суму констант зведення (можна побачити в дужках).

i, j	1	6	8	d_i
3	∞	2	0(2)	2
6	0(0)	∞	0(0)	0
8	0(0)	0(2)	∞	0
d_j	0	2	0	0

$d(3,8) = 2 + 0 = 2$; $d(6,1) = 0 + 0 = 0$; $d(6,8) = 0 + 0 = 0$; $d(8,1) = 0 + 0 = 0$; $d(8,6) = 0 + 2 = 2$;

Найбільша сума констант зведення дорівнює $(2+0) = 2$ для ребра $(3, 8)$.

Відповідно множина розбивається на дві підмножини $(3,8)$ і $(3^*, 8^*)$.

Замінюємо елемент $d_{38} = 0$ на ∞ , тим самим провівши виключення ребра $(3,8)$.

Після чого робимо зведення матриці відстаней для отриманої підмножини $(3^*, 8^*)$, в результаті отримаємо скорочену матрицю.

i, j	1	6	8	d_i
3	∞	2	∞	2

6	0	∞	0	0
8	0	0	∞	0
d_j	0	0	0	2

Нижня границя гамільтонових циклів цієї підмножини: $H(3^*, 8^*) = 39 + 2 = 41$.

Включення ребра (3,8) проводиться шляхом виключення всіх елементів третього рядка і восьмого стовбця, в якій елемент d_{83} замінюємо на ∞ , щоб виключити можливість утворення негамільтонового циклу. В результаті отримаємо іншу скорочену матрицю, яка підлягає операції зведення. Після зведення скорочена матриця буде виглядати так:

i, j	1	6	d_i
6	0	∞	0
8	0	0	0
d_j	0	0	0

Сума констант зведення скороченої матриці: $\sum d_i + \sum d_j = 0$

Нижня границя підмножини (3,8) дорівнює: $H(3,8) = 39 + 0 = 39 \leq 41$

Оскільки нижня границя підмножини (3,8) менша за $(3^*, 8^*)$, то ребро (3,8) включаємо в маршрут з новою границею $H = 39$

У відповідності до цієї матриці включаємо ребра (6,1) і (8,6) в гамільтонів маршрут. Отже отримуємо гамільтонів цикл:

(1,5),(5,10),(10,4),(4,9),(9,7),(7,2),(2,3),(3,8),(8,6),(6,1)

Довжина маршруту дорівнює $F(M_k) = 43$.

2.3 Вирішення задачі в Excel

1. Задамо матрицю відстаней між містами

	A	B	C	D	E	F	G	H	I	J	K
1		1	2	3	4	5	6	7	8	9	10
2	1	1000000	16	12	10	4	14	12	12	11	8
3	2	15	1000000	6	3	14	10	3	8	4	12
4	3	11	6	1000000	4	13	5	4	3	4	7
5	4	11	4	5	1000000	9	11	2	7	1	3
6	5	4	15	12	9	1000000	15	11	14	9	7
7	6	11	9	4	7	13	1000000	6	4	6	8
8	7	11	3	4	2	11	9	1000000	5	2	5
9	8	12	10	6	8	14	5	6	1000000	10	10
10	9	11	4	4	2	10	9	1	6	1000000	4
11	10	8	5	6	2	7	11	4	8	3	1000000

2. Задамо обмеження на рішення задачі

	A	B	C	D	E	F	G	H	I	J	K	L
13		1	2	3	4	5	6	7	8	9	10	
14	1											0
15	2											0
16	3											0
17	4											0
18	5											0
19	6											0
20	7											0
21	8											0
22	9											0
23	10											0
24		0	0	0	0	0	0	0	0	0	0	0

3. Задамо цільову функцію

	A	B	C	D	E
25	Ціл. Фун	=СУММПРОИЗВ(B2:K11;B14:K23)			

4. Задамо клітинки для додаткових змінних

	A	B	C	D	E	F	G	H	I	J	K
26	Дод. Змін.										

5. Задамо умови для додаткових змінних

	A	B	C	D	E	F	G	H	I	J
28		u2	u3	u4	u5	u6	u7	u8	u9	u10
29	u2	0	0	0	0	0	0	0	0	0
30	u3	0	0	0	0	0	0	0	0	0
31	u4	0	0	0	0	0	0	0	0	0
32	u5	0	0	0	0	0	0	0	0	0
33	u6	0	0	0	0	0	0	0	0	0
34	u7	0	0	0	0	0	0	0	0	0
35	u8	0	0	0	0	0	0	0	0	0
36	u9	0	0	0	0	0	0	0	0	0
37	u10	0	0	0	0	0	0	0	0	0

6. Перейдімо в клітинку B25(значення цільової функції)

7. Необхідно натиснути “Дані → Пошук рішення”. Отримасмо діалогове вікно:

Параметры поиска решения

Оптимизировать целевую функцию:

\$B\$25

↑

До:

☒ Максимум
☐ Минимум
☐ Значения:

0

Изменяя ячейки переменных:

↑

В соответствии с ограничениями:

Добавить

Изменить

Удалить

Сбросить

Загрузить/сохранить

☒ Сделать переменные без ограничений неотрицательными

Выберите метод решения:

Поиск решения нелинейных задач методом ОПГ

▼

Параметры

Метод решения

Для гладких нелинейных задач используйте поиск решения нелинейных задач методом ОПГ, для линейных задач - поиск решения линейных задач симплекс-методом, а для негладких задач - эволюционный поиск решения.

Справка

Найти решение

Закрыть

В ньому необхідно:

А) Вибрати метод вирішення симплекс методом

Выберите метод решения: Поиск решения лин. задач симплекс-методом

Б) Поставити оптимізацію цільової функції до мінімуму

Оптимизировать целевую функцию: \$B\$25

До: ☐ Максимум ☒ Минимум ☐ Значения: 0

В) В полі “змінюючи клітинки змінних” задати наступні параметри:

B14:K23(пункт 2); C26:K26(пункт 4)

Изменяя ячейки переменных: \$B\$14:\$K\$23;\$C\$26:\$K\$26

Г) В полі “Обмеження” задамо необхідні обмеження: B14:K23 – цілі; B24:K24 = 1; B29:J37 <= 8; L14:L23 = 1

В соответствии с ограничениями:

\$B\$14:\$K\$23 = целое
 \$B\$24:\$K\$24 = 1
 \$B\$29:\$J\$37 <= 8
 \$L\$14:\$L\$23 = 1

Добавить
 Изменить
 Удалить
 Сбросить
 Загрузить/сохранить

Д) Натиснути кнопку “Розрахувати”

8. А) Отримали значення цільової функції 43 (як і за методом гілок та меж)

	А	В
25	Ціл. Фун	43

Б) Отримали оптимальний шлях

	A	B	C	D	E	F	G	H	I	J	K	L
13		1	2	3	4	5	6	7	8	9	10	
14	1	0	0	0	0	1	0	0	0	0	0	1
15	2	0	0	1	0	0	0	0	0	0	0	1
16	3	0	0	0	0	0	0	0	1	0	0	1
17	4	0	0	0	0	0	0	0	0	1	0	1
18	5	0	0	0	0	0	0	0	0	0	1	1
19	6	1	0	0	0	0	0	0	0	0	0	1
20	7	0	1	0	0	0	0	0	0	0	0	1
21	8	0	0	0	0	0	1	0	0	0	0	1
22	9	0	0	0	0	0	0	1	0	0	0	1
23	10	0	0	0	1	0	0	0	0	0	0	1
24		1	1	1	1	1	1	1	1	1	1	

В) Ось таких значень набувають додаткові змінні

	A	B	C	D	E	F	G	H	I	J	K
26	Дод. Змін.		5	6	2	0	8	4	7	3	1

Г) Ось таких значень набувають умови для додаткових змінних

	A	B	C	D	E	F	G	H	I	J
28		u2	u3	u4	u5	u6	u7	u8	u9	u10
29	u2	0	8	3	5	-3	1	-2	2	4
30	u3	1	0	4	6	-2	2	8	3	5
31	u4	-3	-4	0	2	-6	-2	-5	8	1
32	u5	-5	-6	-2	0	-8	-4	-7	-3	8
33	u6	3	2	6	8	0	4	1	5	7
34	u7	8	-2	2	4	-4	0	-3	1	3
35	u8	2	1	5	7	8	3	0	4	6
36	u9	-2	-3	1	3	-5	8	-4	0	2
37	u10	-4	-5	8	1	-7	-3	-6	-2	0

2.4 Алгоритм мурашиної колонії

Запропонована в 1992 році Марко Доріго, оптимізація колонії мурашок (АСО) - це оптимізаційна техніка, натхненна поведінкою мурашок, що шукають їжу, при пошуку шляху. АСО також є підмножиною ройового інтелекту - техніки вирішення проблем за допомогою децентралізованої колективної поведінки для виведення штучного інтелекту. Типовими додатками оптимізації колонії мурашок є комбінаторні завдання оптимізації, такі як проблема “продавця”, але вона також може бути використана для вирішення різних завдань планування та маршрутизації.

Однією з переваг алгоритмів оптимізації колоній мурашок перед іншими алгоритмами оптимізації є їх здатність адаптуватися до динамічних середовищ - особливість, яка робить його чудовим для таких додатків, як мережева маршрутизація, де, швидше за все, часто виникають зміни, щоб їх адаптувати. В цій розрахунково-графічній роботі ми розглянемо застосування алгоритму АСО до проблеми “продавця” (TSP).

Щоб повністю зрозуміти оптимізацію колонії мурашок, важливо спочатку зрозуміти природну поведінку мурах, яка спочатку надихнула алгоритм.

При пошуку їжі мурахи дотримуються відносно основного набору правил. Ці правила, хоча і прості, дозволяють мурахам спілкуватися та спільно оптимізувати свої шляхи до джерел їжі. Однією з ключових характеристик усіх цих правил є використання феромонних стежок. Феромонові стежки - це, по суті, те, що мурахи використовують для спілкування з іншими мурахами про знайдене джерело їжі та про те, як до нього дістатися. Коли інші мурахи натрапляють на феромонові стежки, вони, як правило, можуть розраховувати знайти їжу, якщо вирішать їй слідувати. Однак мурахи не йдуть по кожному виявленому феромоновому сліду. Залежно від сили феромонного сліду, мураха може вирішити піти іншим шляхом, або, можливо, абсолютно випадковим шляхом, на

якому немає феромону. Однак у середньому, чим сильніший феромоновий слід, тим більше шансів у мурахи його взяти.

З часом феромонові стежки поступово випаровуватимуться, якщо не будуть посилені іншими мурахами. Це означає, що феромонові стежки, які більше не ведуть до джерела їжі, врешті-решт перестануть використовуватися, стимулюючи мурах шукати нові шляхи та нові джерела їжі.

Основна частина алгоритму оптимізації колонії мурах складається лише з декількох кроків.

По-перше, кожен мураха в колонії конструює розчин на основі попередньо відкладених феромонових стежок. Далі мурахи прокладуть феромонові стежки на компоненти обраного ними розчину, залежно від якості розчину. У прикладі проблеми “продавця”, що подорожує, це були б краї (або шляхи між містами). Нарешті, після того, як усі мурахи закінчили побудову розчину і проклали свої феромонні стежки, феромон випаровується з кожного компонента залежно від швидкості випаровування феромону. Потім ці кроки виконуються стільки разів, скільки потрібно для створення оптимального рішення.

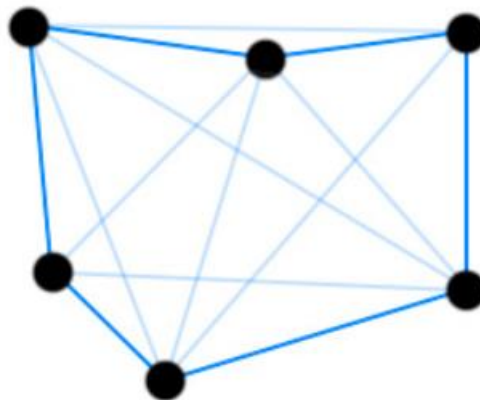
Як уже згадувалося раніше, мураха часто буде слідувати найсильнішим феромоновим слідом при розробці розчину. Однак, щоб мураха розглядала рішення, крім найкращих на даний момент, у процесі прийняття рішення потрібна невелика кількість випадковості. На додаток до цього, обчислюється також евристична цінність, яка розглядається як допомога для спрямування процесу пошуку на найкращі рішення. У прикладі проблеми мандрівного продавця цією евристикою, як правило, буде довжина шляху між містом, яке розглядається - чим коротший шлях, тим більша ймовірність, що його вибере мураха.

$$p_{ij}^k = \frac{[t_{ij}]^\alpha \cdot [n_{ij}]^\beta}{\sum_{l \in N_i^k} [t_{il}]^\alpha \cdot [n_{il}]^\beta}$$

Це рівняння обчислює ймовірність вибору однієї складової рішення. Тут t_{ij} позначає кількість феромону в компоненті між станами i та j , а n_{ij} позначає його евристичну цінність. α та β - обидва параметри, що використовуються для контролю важливості феромонного сліду та евристичної інформації під час вибору компонента.

У випадку проблеми продавця держава представляла б на графіку єдине місто. Тут мураха обирає наступне місто залежно від відстані до наступного міста та кількості феромону на шляху між двома містами.

Локальний процес оновлення феромонів застосовується кожного разу, коли мураха успішно створює розчин. Цей крок імітує спосіб прокладання мурахами феромонових стежок після пошуку їжі в природі. Як ви пам'ятаєте раніше, в природі кращі шляхи набувають більше феромону завдяки тому, що мурахи можуть швидше їх пройти. В АСО ця характеристика відтворюється шляхом варіювання кількості феромону, що осідає на компоненті, з урахуванням того, наскільки добре оцінений готовий розчин. Якщо для прикладу взяти проблему продавця, який подорожує, феромон буде осідати на стежках, якими пройшов мураха між містами, залежно від загальної дистанції туру.



На цьому зображенні показані феромонові стежки, прокладені на шляхах між містами, що є типовою проблемою “продавця”.

$$\Delta\tau_{ij}^k = \begin{cases} Q/C^k \\ 0 \end{cases}$$

Випадок Q/C^k виконується, якщо компонента (i, j) була використана мурахою

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k$$

Тут C_k визначається як загальна вартість рішення, у прикладі проблеми продавця це буде тривалість туру. Q - це просто параметр для регулювання кількості осадженого феромону, як правило, він встановлюється рівним 1. Ми підсумовуємо Q / C_k для кожного розчину, який використовував компонент (i, j) , тоді це значення стає кількістю феромону, який потрібно осадити на компонент (i, j) .

Глобальне оновлення феромону - це стадія, на якій феромон випаровується з компонентів. Він застосовується після кожної ітерації алгоритму, коли кожен мураха успішно побудував шлях і застосував правило локального оновлення.

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij}$$

Де τ_{ij} - феромон компонента між станом i та j та ρ - параметр, що використовується для зміни рівня випаровування.

3 Висновки

У ході виконання курсової роботи було розглянуто задачу про комівояжера. Було визначено навіщо необхідно знати вирішення даної проблеми. Було реалізовано розв'язок за допомогою методу гілок та меж. А також за допомогою Excel. Також було розглянуто як працює алгоритм мурашиної колонії. Суть підходу полягає в аналізі та використанні моделі поведінки мурах, що шукають шляхи від колонії до джерела живлення, і являє собою метаевристичну оптимізацію. Практичне значення одержаних результатів допоможе мандрівникам та компаніям у пошуку найкоротшого шаршруту.

4 Список літератури

1. M. Dorigo, V. Maniezzo, A. Coloni, "The Ant System: Optimization by a colony of cooperating agents" / *Man, and Cybernetics-Part B*, стор. 29-41
2. L. M. Gambardella, M. Dorigo, "Ant-Q: A Reinforcement Learning Approach to the Traveling Salesman Problem" / [журнал] *Morgan Kaufmann* 1995 p.
3. T. Stützle, H. Hoos, "MAX-MIN Ant System and local search for the traveling salesman problem" / *International Conference on Evolutionary Computation*, 1997 p.
4. M. Dorigo, S. Member and L. M. Gambardella "Ant colony system: a cooperative learning approach to the TSP" / *TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, VOL. 1, NO. 1, APRIL 1997
<http://people.idsia.ch/~luca/acs-ec97.pdf>
5. Bernd Bullnheimer, Richard F. Hartl, Christine Strauß, "A new rank based version of the Ant System. A computational study" / *Adaptive Information Systems and Modelling in Economics and Management Science*, 1997 p.
6. Штовба С. Д. "Муравьиные алгоритмы" // [журнал] *Exponenta Pro*. 2003. No 4. С.70-75
http://www.serhiy-shtovba.narod.ru/doc/Shtovba_Ant_Algorithms_ExponentaPro_2003_3.pdf
7. M. Dorigo, D. Corde, F. Glover "New Ideas in Optimization" McGraw-Hill 1999 p.
8. M. Dorigo, "Optimization, Learning and Natural Algorithms", [дисертація] 1992p.
9. Ашманов С.А. "Линейное программирование". М.:Наука, 1981.
10. Ершов Н.М. "Муравьиные алгоритмы" [Лекція] 2011 p.
http://hpc-education.ru/files/lectures/2011/ershov/ershov_2011_lectures10.pdf
11. А. Сухарев, А.Тимохов, В. Федоров "Курс Методов Оптимизации", 1986.