УДК 519.866

М.Зайченко
Тернопільський національний технічний університет імені Івана Пулюя, Україна
Науковий керівник: Н.Гарматій , канд.екон.наук, доц.

ПРАКТИЧНІ АСПЕКТИ ЗАСТОСУВАННЯ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ СИСТЕМ МАСОВОГО ОБСЛУГОВУВАННЯ НА ПРИКЛАДІ РОЗРАХУНКУ ОПТИМАЛЬНОГО МАРШРУТУ ДОСТАВКИ ПИТНОЇ ВОДИ ВІД ЛОГІСТИЧНОГО ЦЕНТРУ ДО ЗАКЛАДІВ ОСВІТИ

M.Zaichenko
Ternopil Ivan Puluj National Technical University, Ukraine
Supervisor: N.Harmatiy, Ph.D, Assos Prof.

PRACTICAL ASPECTS OF APPLYING SIMULATION MODELING OF MASS SERVICE SYSTEMS: A CASE STUDY OF CALCULATING THE OPTIMAL DELIVERY ROUTE FOR DRINKING WATER FROM A LOGISTICS CENTER TO EDUCATIONAL INSTITUTIONS

Задача комівояжера або TSP (Travelling Salesman Problem), є класичною проблемою в області комбінаторної оптимізації та теоретичної інформатики. Завдання полягає в знаходженні найкоротшого можливого маршруту, який проходить через заданий набір точок (міст, локацій) лише один раз і повертається до початкової точки. На даний момент не існує одного алгоритму, який може розв'язати цю задачу за поліноміальний час для всіх можливих наборів даних. Це робить таку задачу важливою для дослідження в областях, пов'язаних з алгоритмічною теорією та оптимізацією.

Для розрахунку оптимального маршруту доставки питної води від складу до шкіл Дарницького району м. Києва, спершу було складено матрицю відстаней, це відображено на рисунку 1.

```
const distances = [
    [0, 12, 8, 23, 38, 56, 25, 39, 21],
    [12, 0, 15, 19, 31, 34, 15, 35, 10],
    [8, 15, 0, 28, 40, 47, 28, 43, 23],
    [23, 19, 28, 0, 25, 16, 10, 16, 8],
    [38, 21, 25, 25, 0, 23, 24, 37, 17],
    [56, 24, 36, 16, 23, 0, 23, 21, 18],
    [25, 13, 21, 10, 24, 23, 0, 27, 8],
    [39, 35, 43, 16, 37, 21, 27, 0, 25],
    [21, 10, 23, 8, 17, 18, 8, 25, 0]
];
```

Рис.1 Матриця відстаней

За допомогою мови програмування JavaScript було написано функцію для пошуку оптимального маршруту, код функції зображено на рисунку 2.

```
function findShortestPath(distances) {
    const visited = Array(distances.length).fill(false);
    let route = [0];
    visited[0] = true;

    for (let i = 0; i < distances.length - 1; i++) {
        let minDistance = Infinity;
        let nextPoint = -1;

        for (let j = 1; j < distances.length; j++) {
            if (!visited[j] && distances[route[i]][j] < minDistance) {
                minDistance = distances[route[i]][j];
                nextPoint = j;
            }
        }

        route.push(nextPoint);
        visited[nextPoint] = true;
    }

    route.push(0); // Повернення на склад
    return route;
}

const shortestPath = findShortestPath(distances);
```

Рис. 2 Функція пошуку оптимального маршруту (візуалізація)

Алгоритм функціонування визначається наступним чином:

1.Ініціація процедури. Процес розпочинається із визначення стартової точки, яка слугує логістичним центром. Ця точка включається до маршруту, а її статус у масиві visited визначається як "відвідана".

2. Визначення найближчого сусіда. Алгоритм продовжується аналізом дистанцій від поточної точки до інших невідвіданих точок. Цей крок включає порівняння відстаней з метою ідентифікації найближчої ще не відвіданої точки.

3. Оновлення маршруту. Після ідентифікації найближчої точки, її інтегрують у планований маршрут, одночасно оновлюючи її статус у масиві visited на "відвідана".

4.Ітераційний процес. Далі алгоритм повторює вищеописані кроки для нової поточної точки (тобто найближчого сусіда, що було визначено на попередньому кроці) і продовжується до того часу, поки всі точки не будуть відвідані

5. Заключний етап – повернення до логістичного центру. Останнім етапом є повернення до стартової точки, яка є логістичним центром, що сигналізує про завершення маршруту.

Результатом виконання функції є послідовність точок, яку описує шлях доставки питної води та повернення до логістичного центру. Результати виконання функції зображено на рисунку 3.

```
Найкоротший маршрут:  ▶ (10) [0, 2, 1, 8, 3, 6, 5, 7, 4, 0]
```

Рис. 3 Найкоротший маршрут доставки води

Перевага запропонованого алгоритму полягає у його здатності мінімізувати загальну пройдену відстань через систематичний вибір найближчих невідвіданих точок. Це сприяє зниженню часу доставки та паливної ефективності, особливо у сценаріях з великою кількістю розподільних точок. Такий підхід також спрощує планування маршруту, забезпечуючи легкість в адаптації до змінених умов або несподіваних перешкод на

маршруті. Крім того, алгоритм володіє високою гнучкістю, що дозволяє легко інтегрувати його у різні логістичні системи. Використання структурованого підходу до відстеження відвіданих точок (через масив visited) гарантує, що всі критичні локації будуть охоплені без повторних відвідувань, що оптимізує загальний час і витрати. Однак, слід відмітити, що алгоритм може бути не оптимальним для великих масштабів або складних маршрутів, де потрібно враховувати численні змінні та обмеження. Тим не менш, його ефективність у стандартних сценаріях робить його цінним інструментом у сфері логістики та розподілу.

Література

1. Analysis of travelling salesman problem. URL: https://iopscience.iop.org/article/10.1088/1757-899X/263/4/042085/pdf
2. Vanderbei R. J. Linear programming: Foundations and extensions. Springer, 2014. 414 p.