УДК 510, 681

В.О.Денисюк

АНАЛІЗ ЗАДАЧІ КОМІВОЯЖЕРА ЗА ДОПОМОГОЮ АЛГОРИТМІВ КОМП'ЮТЕРНОЇ ГРАФІКИ

Вінницький національний аграрний університет

Анотація

Тези містять обґрунтування актуальності розв'язання задачі комівояджера. Основна увага приділена аналізу предметної області задачі комівояжера. Практична цінність роботи полягає в обиранні напрямку досліджень. Для досягнення поставленої мети пропонується об'єднати сучасні напрямки розв'язання задачі комівояжера з сучасними програмними та апаратними засобами комп'ютерної графіки.

Ключові слова: задача комівояжера, комп'ютерна графіка, піксель, шейдер.

Abstract

Theses contain a substantiation of the urgency of solving the problem of a salesman. The main attention is paid to the analysis of the subject area of the salesman's task. The practical value of the work lies in choosing the direction of research. To achieve this goal, it is proposed to combine modern directions of solving the problem of the salesman with modern software and hardware of computer graphics.

Keywords: salesman task, computer graphics, pixel, shader.

Вступ

Задача про комівояжера - Travelling Salesman Problem (TSP), - є однією із найбільш поширених задач комбінаторної оптимізації, до яких можна звести і ряд інших проблем оптимізації [1, 2]. Класична задача комівояжера полягає у знаходженні найвигіднішого маршруту, що проходить через вказані міста хоча б по одному разу. В умовах завдання вказуються критерій вигідності маршруту (найкоротший, найдешевший, сукупний критерій тощо) і відповідні матриці відстаней, вартості тощо. Прикладом таких проблем є пошук оптимальних туристичних маршрутів, задача оптимізації шляху експедитора, задача трасування електронних плат, вибір оптимального маршруту обльоту безпілотних літальних апаратів у реальному часі та ін. [1, 2]. Для розв'язування наведених задач використовуються прикладні алгоритми комбінаторної оптимізації. Майже всі вони мають деяку кількість параметрів. Саме тому дослідження та вибір параметрів (коефіцієнтів) матриці відстаней для розв'язування задачі комівояжера є актуальним. Для цього необхідно дослідити підходи до обирання параметрів матриці відстаней. Зміст задачі TSP такий: існує множина міст $\boldsymbol{N}$, відстань між якими відома, необхідно визначити такий оптимальний маршрут, який проходить через усі $\boldsymbol{N}$ міст без повторного відвідування (зазвичай задано, що маршрут повинен проходити через кожне місто тільки один раз, в такому випадку розв'язок знаходиться серед гамільтонових циклів) [1, 2]. В умовах завдання вказуються критерій вигідності маршруту (найкоротший, найдешевший, сукупний критерій тощо) і відповідні матриці відстаней, вартості.

Постановка задачі

Проблема побудови оптимальних маршрутів через задану множину точок на площині чи у просторі виникає у багатьох сферах людської діяльності, наприклад, задачах планування та логістики; при виробництві друкованих плат; мінімізації рухів у робототехніці; аналізі структури ДНК та ін. За суттю, усі ці проблеми зводяться до розв'язування задачі комівояжера.

Для можливості застосування математичного апарату для розв'язання TSP, її слід представити у вигляді математичної моделі, а саме у вигляді моделі на графі, тобто, використовуючи вершини та ребра між ними. Таким чином, вершини графу відповідають містам, а ребра $\boldsymbol{(i, j)}$ між вершинами $\boldsymbol{i}$ та $\boldsymbol{j}$ - сполученням між цими містами. У відповідність кожному ребру $\boldsymbol{(i, j)}$ можна зіставити вагу $\boldsymbol{c_{ij} \geq 0}$, яку можна розуміти як, наприклад, відстань між містами, час або вартість подорожі. TSP описується математична так: між кожним містом відома відстань $\boldsymbol{C \in R^{N\times N}}$. Треба побудувати такий маршрут $\boldsymbol{X_* \in R^{N\times N}}$, який проходить через кожне місто тільки один раз, крім початкового міста, та має мінімальну довжину. Тобто потрібно знайти замкнений гамільтонів цикл мінімальної довжини. Отже, необхідно знайти (1) при обмеженнях (2-4)

$$\boldsymbol{X_* = \underset{x_{ij}\in X\in R^{N\times N}}{argmin} \sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij} \times x_{ij}}\ , \tag{1}$$

$$\boldsymbol{\sum_{i=1}^{N} x_{ij} = 1,\ \ j = \overline{1, N}}\ , \tag{2}$$

$$\boldsymbol{\sum_{j=1}^{N} x_{ij} = 1,\ \ i = \overline{1, N}}\ , \tag{3}$$

$$\boldsymbol{x_{ij}} \in \{\boldsymbol{0}, \boldsymbol{1}\}\ , \tag{4}$$

де $\boldsymbol{N}$ – кількість міст;

$\boldsymbol{x_{ij}}$ – наявність переходу від міста $\boldsymbol{i}$ до міста $\boldsymbol{j}$ у маршруті;

$\boldsymbol{X_*}$ – маршрут для обходу усіх міст;

$\boldsymbol{c_{ij}}$ – відстань (вага) переходу від міста $\boldsymbol{i}$ до міста $\boldsymbol{j}$ у маршруті.

Цільова функція визначається у (1) - знаходження такого маршруту, при якому його довжина буде мінімальною. Обмеження (2) та (3) вказує на умову відвідування міста тільки один раз.

TSP має багато модифікацій, а саме [2]: задача кур'єра, кластерна TSP, чорна та біла TSP, TSP з декількома комівояжерами, задача покриття для комівояжера, TSP з часовими вікнами. Задачу комівояжера за вхідними даними поділяють на симетричну та асиметричну, причому, симетрична TSP має більше поширення. Задача комівояжера є NP-повною [2]. У 1972 році Річард Карп довів NP-повноту задачі пошуку гамільтонових шляхів, із чого, через поліноміальну зводимість, випливала NP-повнота задачі комівояжера. На основі цих властивостей ним було наведено теоретичне обґрунтування складності пошуку розв'язків задачі на практиці. Отже, немає поліноміального алгоритму для розв’язування TSP. Тому для її розв’язування використовуються алгоритми для розв’язування задач комбінаторної оптимізації, Combinatorial Optimization Problem (COP). Характерною особливістю сьогодення є істотне зростання розмірності TSP (до мільйонів точок), а також наявність часткових задач зі специфічними властивостями. Це – динамічні транспортні задачі, особливістю яких є поява нових точок обслуговування в процесі реалізації заданого маршруту; системи за викликом (швидка допомога, кур’єрська, пожежна служби, таксі), системи з часовими вікнами, системи постачання та інші. Вони потребують розроблення спеціальних алгоритмів, які б забезпечували отримання якісних результатів в режимі реального часу [2]. Асиметрична задача комівояжера відрізняється тим, що ребра між вершинами можуть мати різну вагу в залежності від напряму, тобто, задача моделюється орієнтованим графом. Таким чином, окрім ваги ребер графа, слід також зважати і на те, в якому напрямку знаходяться ребра. Симетрична задача така: усі пари ребер між одними й тими самими вершинами мають однакову вагу, тобто, для ребра ***(i, j)*** ваги однакові $\boldsymbol{c_{ij}}$=$\boldsymbol{c_{ji}}$. Отже, всі маршрути мають однакову довжину в обидва напрямки. В симетричному випадку кількість можливих маршрутів вдвічі менша за асиметричний випадок. Симетрична задача моделюється неорієнтованим графом [1, 2].

Оскільки комівояжер в кожному із міст постає перед вибором наступного міста з тих, що він ще не відвідав, існує ***(n-1)!*** маршрутів для асиметричної та ***(n-1)!/2*** маршрутів для симетричної задачі комівояжера. З цього слідує, що розмір простору пошуку залежить над-експоненційно від кількості міст ***N***.

Симетричну задачу комівояжера називають метричною, якщо відносно довжин ребер виконується нерівність трикутника. Умовно кажучи, в таких задачах обхідні шляхи довші за прямі, тобто, ребро від вершини ***i*** до вершини ***j*** ніколи не довше за шлях через проміжну вершину ***k***.

Насправді, задача комівояжера у випадку реальних міст може бути як симетричною, так і асиметричною в залежності від тривалості або довжини маршрутів в залежності від напряму руху [1]. Не-метрична задача комівояжера може виникати, наприклад, у випадку мінімізації тривалості подорожі за наявності вибору транспортних засобів в різних напрямах. В такому випадку обхідний шлях літаком або морем може бути коротший за пряме сполучення автомобілем [1].

Для розв’язування TSP використовують різні алгоритми комбінаторної оптимізації. Вони поділяються на точні та наближенні. Точні алгоритми – це алгоритми, які за скінчений час гарантовано повертають оптимальний розв'язок, або роблять висновок що його не існує, якщо задача нерозв'язна. Ці алгоритми можна поділити на 2 класи: загальні та спеціальні методи. Загальні методи можуть бути використані для широкого кола задач. Спеціальні будується для конкретних задач з урахуванням їх специфіки, як наприклад, метод Балаша для розв'язування лінійної задачі про призначення. До точних алгоритмів відносять [2]: повний перебір (вичерпний пошук); метод гілок та меж, Branch-and-Bound Method (BBM); метод відтинань (алгоритм Гоморі); послідовний аналіз та відсіювання варіантів (послідовний аналіз варіантів, «київський віник»); динамічне програмування (метод Белмана). Але ці алгоритми є трудомісткими та потребують дуже великої кількості операцій при збільшені кількості міст в TSP. Так, наприклад, якщо для задачі комівояжера з $\boldsymbol{n}$ міст асимптотична складність повного перебору складає $\boldsymbol{O(n!)}$, практично такі ж оцінки мають інші точні алгоритми. Це потребує великої кількості часу для виконання, що у більшості випадків є неприйнятним на практиці. Існує оцінка залежності гіпотетичного часу роботи алгоритмів від розмірності задачі, де робиться припущення, що один елементарний крок виконується одну наносекунду. Найбільш популярними серед алгоритмів розв'язування СОР є наближені алгоритми комбінаторної оптимізації. Вони є поширеними тому що: СОР має зазвичай велику кількість локальних екстремумів; дані можуть задаватись з певними похибками; наближені методи дозволяють створювати алгоритми, які можуть розв'язувати не одну задачу, а цілий клас задач близьких за постановкою оптимізаційних задач [2].

Висновки

Пропонується розглядати задачу про комівояжера як рух умовного об'єкта (маркера, точки, пікселя) по дискретному координатному простору (ДКП), обмеженого своїми максимальними параметрами довжини та ширини. Можна також цей ДКП вважати пласкою мапою в прямокутній системі координат (декартовій системі координат) с розмірами по горизонталі та вертикалі як в значеннях GPS-координат, так і в значеннях у км [3]. Також мапа повинна бути такою аби повністю вбирати в себе маршрут комівояжера. Вважаємо, що горизонтальні координати вимірюємо за віссю *OX*, а вертикальні – за віссю *OY*, де *O* – точка початку цієї системи координат та має впорядковану пару координат $(\boldsymbol{X_O}, \boldsymbol{Y_O})$ - ***(0, 0)***. Координати точок у ДКП визначаємо як положення ортогональних проекцій точок на ці дві осі, що задаються як знакові відстані від початку координат. Реальні значення координат об'єктів, між якими переміщується комівояжер, завжди можна перерахувати у абсолютні значення координат обраного ДКП. Таким чином, комівояжер переміщується між точками ДКП $(\boldsymbol{X_i}, \boldsymbol{Y_i})$, де $\boldsymbol{i} \in \{1, 2, 3, \dots, N\}$ та є натуральним числом, $\boldsymbol{N}$ – максимальна кількість точок подорожі комівояжера. Отже, цілком слушно використати можливості алгоритмів комп'ютерної та машинної графіки для визначення відстаней між точками $\boldsymbol{i}$ $(\boldsymbol{X_i}, \boldsymbol{Y_i})$ та $\boldsymbol{j}$ $(\boldsymbol{X_j}, \boldsymbol{Y_j})$**,** як довжини ребер графу, що відповідає TSP. Причому, якщо зробити відповідну конвеєризацію та розпаралелювання обчислень, то з тією чи іншою глибиною передбачення можливо побудувати прийнятний маршрут TSP. Для розв'язання таких задач цілком підійде більшість графічних акселераторів, наприклад, GeForce, Radeon, з програмуванням наявних в них шейдерів [3].

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Задача комівояжера. https://uk.wikipedia.org/wiki/Задача_ комівояжера.
2. Денисюк В.О. Обирання критеріїв для задачі комівояжера. *The scientific heritage.* https://www.tsh-journal.com/wp-content/uploads/2020/09/VOL-8-No-46-46-2020.pdf
3. Денисюк В.О. Вибір графічного процесору для систем комп'ютерної графіки. *The scientific heritage*. https://www.tsh-journal.com/wp-content/uploads/2020/09/VOL-1-No-47-47-2020.pdf

Денисюк Валерій Олександрович - к.т.н, доцент, доцент кафедри комп'ютерних наук та економічної кібернетики ВНАУ, Вінницький національний аграрний університет, м. Вінниця, vad64@i.ua

***Denisyuk Valeriy O.* -** Ph.D., Associate Professor, Associate Professor of the Department of Computer Science and Economic Cybernetics, VNAU, Vinnytsia National Agrarian University, Vinnytsia, vad64@i.ua