УДК 004.9

ДОСЛІДЖЕННЯ ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ АЛГОРИТМІВ ЗНАХОДЖЕННЯ ОПТИМАЛЬНОГО ШЛЯХУ

Петришин Сергій, Решетник Вікторія

Вінницький національний технічний університет

Анотація

Розглянуто алгоритми, що знаходять найоптимальніший шлях з урахуванням особливостей прохідності місцевості. Описано методи, які умовно можна розділити на наступні класи: алгоритми пошуку найкоротшого шляху; алгоритми пошуку субоптимального шляху.

Ключові слова: пошук шляху, алгоритм Дейкстри, алгоритм А.*

Abstract

The algorithms providing one of the possible variants of a way from one point to another taking into account the peculiarities of the passability of the area are considered. Described methods that can be conventionally divided into the following classes: algorithms for finding the shortest path; algorithms for finding suboptimal paths.

*Keywords: path search, Dijkstra algorithm, A * algorithm.*

Вступ

При створенні симуляторів, що мають переміщувати різні типи об'єктів по великих територіях з урахуванням поточної обстановки, виникають проблеми з вибором алгоритму пошуку оптимального шляху, так як на його використання накладаються обмеження, викликані такими факторами як великий обсяг даних реальних карт місцевості; складність уявлення ландшафту та ін. Оптимальним рішенням може виявитися як пряма, так і ламана лінія.

Головна проблема задачі пошуку шляху заключається в тому, що не існує універсального алгоритму її вирішення.

Результати дослідження

Перебірні алгоритми по суті своїй є алгоритмами пошуку, як правило, пошуку оптимального рішення. При цьому рішення конструюється поступово. В цьому випадку зазвичай говорять про перебір вершин дерева варіантів [1]. Вершинами такого графа будуть проміжні або кінцеві варіанти, а ребра будуть вказувати шляхи побудови варіантів рішення.

Хвильовий алгоритм Лі належать до класу перебірних та складається з двох етапів:

- поширення хвилі;
- зворотній хід [2].

Поширення хвилі і є власне пошук в ширину, при якому клітини позначаються номером кроку методу, на якому клітина відвідується. При зворотному ході, починаючи з кінцевої вершини, йде відновлення шляху, по якому в неї потрапили шляхом включення в нього клітин з мінімальною позначкою. Особливістю алгоритму Лі є те, що відновлення починається з кінця (з початку воно часто неможливо) [3].

Граф представляє собою сукупність непустої множини вершин і ребер (наборів пар вершин). Дві вершини на графі сусідні, якщо вони з'єднуються спільним ребром. Шлях в неорієнтованому графі представляє собою послідовність вершин $P = (v1, v2, \dots, vn) \in V \times V \times \dots \times V$, таких, що vi суміжна з $vi + 1$ для $1 \leq i < n$. Такий шлях P називається шляхом довжиною n з вершини v_1 в v_n (i вказує на номер вершини шляху і не має ніякого відношення до нумерації вершин на графі) [4].

Для вирішення вказаної задачі можна використовувати алгоритм Дейкстри. Він знаходить найкоротшу відстань від однієї вершини графа до всіх інших. Алгоритм

Дейкстри придуманий нідерландським вченим Едсгер Дейкстрой в 1959 році [5].

Складність алгоритму складає $O(n^2 + m)$, де n – кількість вершин і m – кількість ребер.

Недоліком алгоритму є те, що він не завжди працює з ребрами при від'ємному значенні ребер [6].

Алгоритм Флойда-Уоршелла був опублікований Робертом Флойдом і Стівеном Уоршеллом в 1962 році. Він є легшим у реалізації програми у порівнянні з алгоритмом Дейкстри. Алгоритм динамічного програмування Флойда-Уоршелла служить для визначення найкоротших шляхів між усіма парами вершин графа. Даний алгоритм дуже простий у освоєнні, але у нього дуже низька швидкість роботи і в ньому не передбачено розрахунок ребер з негативними вагами.

Мета алгоритму - знаходження найкоротшого шляху між усіма вершинами графа [7].

Метод Флойда базується на тому факті, що в графі з позитивними вагами будь-якого неелементарного ребра (що містить більше 1 ребра), короткий шлях складається з інших найкоротших шляхів.

Дано матрицю, яка заповнюється вагами ребер для кожної з вершин. Після цього проводиться ітерація кожної вершини, де порівнюється вага ребер між вершинами А і Б, через вершину В. Найкоротшим шляхом між А і Б є мінімальне значенн з цих варіантів.

Цей алгоритм більш загальний порівняно з алгоритмом Дейкстри, так як він знаходить найкоротші шляхи між будь-якими двома вершинами графа.

Перегавою алгоритму є його простота реалізації, слабкість – в складності алгоритму. Вона дорівнює $O(n^3)$, де n - кількість вершин графа [8].

Алгоритм бектрекінгу був розроблений багатьма дослідниками ще до формального опису. Метою цього алгоритму не є знаходження найкоротшого шляху, а знаходження всіх можливих варіантів шляхів від вершини А до вершини Б.

Складність алгоритму $O(n * m)$, де n - кількість вершин графа. Алгоритм ефективний тим, що всі можливі варіанти можуть бути збережені в структурах даних і легко можуть бути використані в подальших розрахунках у програмі [9].

Перебірні алгоритми по своїй суті є алгоритмами пошуку, як правило, пошуку оптимального рішення. При цьому рішення конструюється поступово. В цьому випадку звичайно говорять про перебір вершин дерева варіантів. Для прискорення перебору з поверненням обчислень завжди намагаються організувати так, щоб була можливість якомога відмінити недоречні варіанти [10].

Алгоритм пошуку А* належить до евристичних алгоритмів пошуку. Він використовується для пошуку найкоротшого шляху між двома вершинами графу з додатніми вагами ребер. Алгоритм використовує допоміжну функцію (евристику), аби скеровувати напрям пошуку та скорочувати його тривалість [11]. Основним недоліком алгоритму А* є потреба в пам'яті для збереження всіх відомих та досліджених вершин. Через це алгортим А* непридатний для багатьох задач [12].

Однією з основних проблем алгоритму А * є те, що одержані за його допомогою шляхи не виглядають реалістичними. Ця проблема вирішується в алгоритмі Theta*. Ключовий момент, який відрізняє ці два алгоритми, в тому, що Theta * дозволяє в якості «предка» для кожної вершини вибрати будь-яку вершину, на відміну від А*, де «предком» може бути тільки найближчий видимий сусід. Якщо потрібно отримати найбільш реалістичний субоптимальний шлях, рекомендується використовувати алгоритм Theta* [13].

Висновки

Було розглянуто алгоритми, які використовуються для знаходження оптимального шляху на географічних картах, а саме: алгоритм Дейкстри, алгоритм Флойда-Уоршелла, алгоритм бектрекінгу, алгоритм А* та алгоритм Theta*. Також було проведено короткий

огляд перебірних алгоритмів, але вони не підходять для вирішення поставленої задачі через великі затрати пам'яті.

У загальному випадку необхідно розробляти систему алгоритмів. Так як витрати на звернення на ландшафт являються критичними, то будемо використовувати наступну комбінацію алгоритмів:

- алгоритм А * для отримання маршруту;
- видалимо точки, що лежать на одній прямій;
- застосуємо алгоритм перевірки наявності шляху по прямій.

Список використаних джерел:

1. Cook W. J. In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation/ W. J. Cook. – Princeton University Press, 2012. – 254 p.

2. Applegate D.L., Bixby R.E., Chvátal V., Cook W.J. The Traveling Salesman Problem. Princeton University Press, 2007. – 347 p.

3. Ананий Левитин, Игорь Красиков. Алгоритмы. Введение в разработку и анализ/ Левитин А.., Красиков И .– Вильямс, 2006. – 576 с. – ISBN 5-8459-0987-2, 0-201-74395-7.

4. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы. Построение и анализ. Москва, Вильямс, 2005.

5. Botea A., Muller M., Schaeffer J. Near Optimal Hierarchical Path-Finding. Journal of Game Development, 2004, vol. 1, issue 1, pp. 7–28.

6. Daniel K., Nash A., Koenig S., Felner A. Theta*: Any-Angle Path Planning on Grids. Journal of Artificial Intelligence Research, 2010, vol. 39, pp. 533–579.

7. Коробицын В.В., Фролова Ю.В. Моделирование автомобильного потока на перекрестке и определение оптимальных параметров управления сигналами светофора: журнал. Выпуск №2(14), журнал «Математические структуры и моделирование» — 2004 г., — вып. 14, — с. 34—36

8. Седжвик Р. Алгоритмы на Java: учеб. пособие. Вильямс, 2012. — 575 с.

9. Сундетов р.н., есламгалиев т.н. обоснование выбора алгоритма поиска оптимального пути с расчетом показателей светофора в городской сети // естественные и математические науки в современном мире: сб. Ст. По матер. XXX междунар. Науч.-практ. Конф. № 5(29). – новосибирск: сибак, 2015.

10.Семакин, И.Г. Основы алгоритмизации и программирования: Учебник для студ. учреждений сред. проф. образования / И.Г. Семакин, А.П. Шестаков. - М.: ИЦ Академия, 2013. - 304 с.

11.Крупский, В.Н. Математическая логика и теория алгоритмов: Учебное пособие для студентов учреждений высшего проф. образования / В.Н. Крупский, В.Е. Плиско. - М.: ИЦ Академия, 2013. - 416 с.

12.Игошин, В.И. Теория алгоритмов: Учебное пособие / В.И. Игошин. - М.: ИНФРА-М, 2013. - 318 с.

13.Канцедал, С.А. Алгоритмизация и программирование : Учебное пособие / С.А. Канцедал. - М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2013. - 352 с.