

Поліщук О.С., студентка 1 курсу спеціальності 122 «Комп'ютерні науки»

Ніколюк П. К., д-р фіз.-мат. наук, професор кафедри інформаційних технологій

ЗАСТОСУВАННЯ АЛГОРИТМУ ДЕЙКСТРИ ДЛЯ ПОШУКУ ОПТИМАЛЬНОГО МАРШРУТУ

Донецький національний університет імені Василя Стуса, м. Вінниця

Алгоритм голландського вченого Едсгера Дейкстри знаходить всі найкоротші шляхи з однієї наперед заданої вершини графа до всіх інших. З його допомогою, при наявності всієї необхідної інформації, можна, наприклад, дізнатися яку послідовність доріг краще використовувати, щоб дістатися з одного міста до кожного з багатьох інших, або в які країни вигідніше експортувати нафту тощо.

Мінусом даного методу є неможливість обробки графів, в яких є ребра з негативною вагою. Якщо, наприклад, деяка система передбачає збиткові для фірми маршрути, то для роботи з нею варто скористатися іншим алгоритмом [1].

Розглянемо основні кроки алгоритму Дейкстри:

- Крок 1: Ініціалізація

Алгоритм починає роботу з визначення початкової вершини і призначення їй ваги 0. Усі інші вершини отримують початкову вагу "нескінченність" або дуже велике значення. Також створюється список відвіданих вершин, початковий список найкоротших шляхів і список попередніх вершин.

- Крок 2: Обхід сусідніх вершин

Алгоритм обходить всі сусідні вершини по черзі та оновлює їх ваги, якщо знайдено коротший шлях. Якщо вага поточного шляху та вага поточної вершини разом менше, ніж вага сусідньої вершини, то вага оновлюється і зберігається, як новий найкоротший шлях.

- Крок 3: Позначення вершини як відвіданої

Після оновлення ваги всіх сусідніх вершин поточної вершини, поточна вершина позначається як відвідана і додається до списку відвіданих вершин.

- Крок 4: Вибір наступної вершини

З наявних невідвіданих вершин обирається вершина з найменшою вагою, яка стала доступною через вже відвідані вершини. Ця вершина стає новою поточною вершиною, і процес повторюється з кроку 2.

- Крок 5: Повторення кроків 2-4

Процес обходу сусідніх вершин, оновлення ваг та позначення вершин як відвіданих повторюється до тих пір, поки всі вершини не будуть відвідані або буде знайдений найкоротший шлях до цільової вершини.

- Крок 6: Визначення найкоротшого шляху

Після того, як усі вершини будуть відвідані, алгоритм Дейкстри визначає найкоротший шлях до кожної вершини від початкової вершини. Цей шлях можна відновити, переходячи від цільової вершини до початкової за допомогою списку попередніх вершин [2].

На рис. 1 зображена послідовність кроків пошуку найкоротших шляхів для всіх вершин з першої для конкретного графа [1].

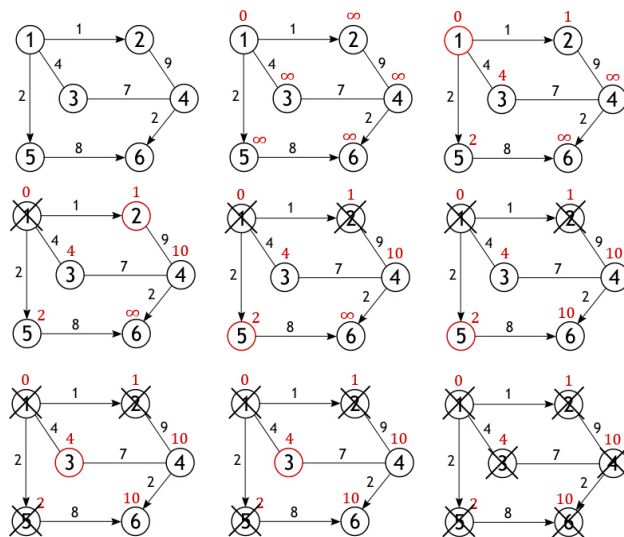


Рисунок 1- Алгоритм Дейкстри

Застосування алгоритму Дейкстри також можна знайти в логістичних системах, де необхідно знаходити оптимальні маршрути для доставки товарів або вантажу. Алгоритм допомагає визначити найефективніші шляхи для мінімізації витрат на транспорт та зниження часу доставки.

У сфері телекомунікацій алгоритм Дейкстри застосовується для оптимізації маршрутизації даних у комп'ютерних мережах. Він дозволяє знаходити найшвидші шляхи для передачі пакетів даних між різними вузлами мережі, що забезпечує ефективне використання ресурсів мережі та знижує затримки у передачі даних.

Однак, важливо враховувати, що алгоритм Дейкстри працює ефективно лише для графів з невід'ємними вагами ребер. У випадку, коли граф містить від'ємні ваги або цикли, більш складні алгоритми, такі як алгоритм Беллмана-Форда або алгоритм Флойда-Воршелла, можуть бути більш підходящими [2].

Отже, алгоритм Дейкстри є потужним і ефективним інструментом для пошуку оптимального маршруту у графах з невід'ємними вагами ребер. Застосування цього алгоритму в різних галузях, включаючи транспортні системи, мережі зв'язку, GPS-навігацію та інтернет-маршрутизацію, дозволяє покращити ефективність і точність маршрутизації, знизити час та витрати на

переміщення, а також покращити загальний досвід користувачів. Застосування алгоритму Дейкстри може сприяти оптимізації різних процесів, що пов'язані з маршрутизацією, транспортом та доставкою, що робить його незамінним інструментом у сучасному світі.

Важливо також зазначити, що розвиток технологій та збільшення обчислювальної потужності дозволяють застосовувати більш складні алгоритми, які можуть враховувати додаткові фактори та обмеження, такі як трафік, густота населення, обмеження швидкості тощо. Однак, алгоритм Дейкстри залишається основою та одним з найпоширеніших методів для пошуку оптимального маршруту.

У підсумку, застосування алгоритму Дейкстри для пошуку оптимального маршруту має широкі можливості у різних галузях, що потребують ефективної маршрутизації. Його використання дозволяє забезпечити швидкий та ефективний пошук найкоротшого шляху, що впливає на покращення процесів транспортування, маршрутизації даних, навігації та інших систем, які вимагають оптимальних маршрутів. Алгоритм Дейкстри залишається одним з найефективніших та популярних методів для вирішення цих завдань [3].

Список літератури.

1. *Labeling Algorithm for Shortest Paths on Road Networks.* / [Abraham I., Delling D., Goldberg A., Werneck R.]. - Philadelphia.- *Symposium on Experimental Algorithms, 2011.* — pp. 230-241.
2. Wikipedia. "Dijkstra's algorithm." Wikipedia, *The Free Encyclopedia.* [Online]. Available: https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm
3. GeeksforGeeks. "Dijkstra's Algorithm for Finding Shortest Path." [Online]. Available: <https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/>

УДК 004.6

*Проценко А. С., студентка I курсу
спеціальності 122 «Комп'ютерні науки»
Науковий керівник:
Гончар В. М., асистент кафедри
інформаційних технологій*

АЛГОРИТМИ ЗНАХОДЖЕННЯ НАЙБІЛЬШОГО ПІДГРАФА З НЕЗАЛЕЖНОЮ МНОЖИНОЮ РЕБЕР

Донецький національний університет імені Василя Стуса, м. Вінниця

У наш час графі є потужним інструментом для моделювання різноманітних систем. Знаходження найбільшого підграфа з незалежною множиною ребер має багато прикладних застосувань, наприклад, в оптимізації мережі транспорту або при проектуванні системи зв'язку.

Далі розглянемо деякі алгоритми для знаходження найбільшого підграфа з незалежною множиною ребер.