

С. С. Бульба<sup>1</sup>, О. І. Соловйова<sup>2</sup>, Ю. О. Семеренко<sup>2</sup>

<sup>1</sup> Національний технічний університет «Харківський політехнічний інститут», Харків, Україна

<sup>2</sup> Харківський національний університет Повітряних Сил імені Івана Кожедуба, Харків, Україна

## ДОСЛІДЖЕННЯ АЛГОРИТМІВ ПОШУКУ ОПТИМАЛЬНОГО ШЛЯХУ

**Анотація.** Обширні області застосування алгоритмів пошуку оптимального шляху призводить до необхідності ретельного їх дослідження. Метою даної роботи є проведення дослідження існуючих методів пошуку оптимального шляху в залежності від особливостей вхідних даних та цілей оптимізації. У статті розглядаються такі види алгоритмів як: алгоритми на основі графів та на основі евристики. У роботі було проведений аналіз сфери застосування наведених алгоритмів. Визначені переваги та недоліки використання алгоритмів в залежності від області використання. Досліджено складності застосування в залежності від обчислювальних потужностей та часу обробки вхідних даних. Представлені дослідження дають змогу отримати інформацію про достовірність результатів роботи алгоритмів що розглядаються в залежності від вхідної сукупності параметрів для обчислення.

**Ключові слова:** алгоритм, програмування, розподіл ресурсів, граф, оптимальний шлях.

### Вступ

У сучасному світі все більше виникає необхідність в побудові різних систем та засобів підвищення якості та комфорту життя людини. З такими викликами все частіше зустрічаються в області інформаційних технологій, а отже необхідно досліджувати існуючі та розробляти нові методи вирішення задач що виникають.

На даний момент більшість користувачів обчислювальних машин не можуть уявити своє існування без навігації на картах, швидкої передачі даних через інтернет, швидкої та дешевої доставки товарів. Також, швидко розвиваються такі напрямлення як: системи безпілотного керування автомобілем, навігація персонажів в відеоіграх, планування маршруту руху для роботів.

В усіх розглянутих сферах, для пошуку ефективного руху та доставки використовують методи та алгоритми пошуку оптимального шляху [1–10].

Алгоритми пошуку оптимального шляху класифікуються за такими критеріями як: тип вхідних даних, та принципом роботи. В залежності від критерію алгоритми можна розподілити на такі групи як:

- алгоритми на основі графів;
- алгоритми на основі штучного інтелекту;
- алгоритмів на основі евристик;
- алгоритми на основі випадкових методів;
- алгоритми на основі комбінаторної організації;
- алгоритми на основі нейронних мереж.

Кожний з наведених видів алгоритмів призначений для вирішення певної задачі, а отже необхідно розуміти коли і який потрібно використовувати щоб отримати оптимальний, або наближений до оптимального результат при мінімальних витратах на обчислювальну потужність та час виконання [11].

Розуміння основних переваг та недоліків роботи алгоритмів що розглядаються дає змогу розробникам та програмістам проектувати швидкі та зручні системи що вирішують поставлену перед ними задачу.

**Мета роботи** – дослідження існуючих методів пошуку оптимального шляху в залежності від особливостей вхідних даних та цілей оптимізації.

Дослідження спрямоване на виявлення оптимальних стратегій використання алгоритмів пошуку шляху в залежності від області використання.

### 1. Алгоритми на основі графів

До алгоритмів даного типу відносяться ті, які для побудови шляху використовують такі вхідні дані як граф. Граф  $G$  це впорядкована пара

$$G = (V, E), \quad (1)$$

де  $V$  – непорожня множина вузлів та вершин,  $E$  – множина пар вершин що називаються ребрами.

Вершини  $u$  та  $v$  називаються кінцевими вершинами (або просто кінцями) ребра  $e = \{u, v\}$ . Ребро, своєю чергою, з'єднує ці вершини. Дві кінцеві вершини того самого ребра називаються суміжними

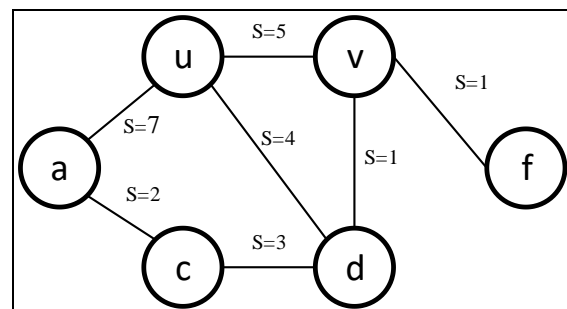


Рис. 1. Ненаправлений граф

Існують різні постановки задачі про найкоротший шлях:

1. Завдання про найкоротший шлях до заданого пункту призначення. Потрібно знайти найкоротший шлях у задану вершину призначення  $t$ , який починається у кожній з вершин графа (крім  $t$ ). Змінюючи напрям кожного ребра, що належить графу, це завдання можна вести до завдання про єдину вихідну вершину (в якій здійснюється пошук найкоротшого шляху із заданої вершини у всі інші).

2. Завдання про найкоротший шлях між заданою парою вершин. Потрібно знайти найкоротший шлях із заданої вершини  $u$  задану вершину  $v$ .

3. Завдання про найкоротший шлях між усіма парами вершин. Потрібно знайти найкоротший шлях

з кожної вершини  $u$  до кожної вершини  $v$ . Це завдання теж можна вирішити за допомогою алгоритму, призначеного для вирішення задачі про одну вихідну вершину, проте зазвичай вона вирішується швидше.

Через різну постановку завдань існує велика кількість алгоритмів для їх розв'язання, до них відносяться:

- Алгоритм Дейкстри;
- Алгоритм А\*;
- Алгоритм Беллмана-Форда;
- Алгоритм Флойда-Уоршелла.

## 2. Алгоритм Дейкстри

Алгоритм пошуку найкоротших шляхів взваженого графу від одного з його вузлів до всіх інших. Даний алгоритм застосовується лише на графах з невід'ємними вагами ребра.

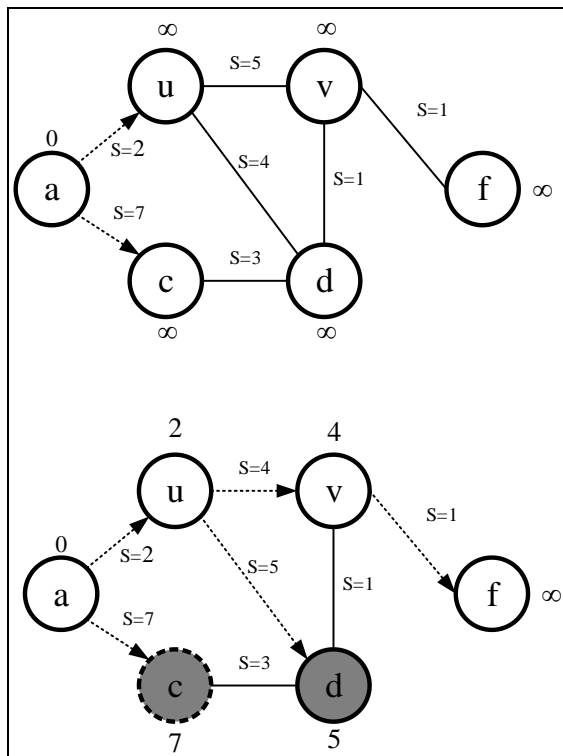


Рис. 2. Кроки алгоритму Дейкстри

Алгоритм Дейкстри складається з наступних кроків:

1. Встановіть початковий вузол і відстань від нього до всіх інших вузлів рівною нескінченності, а відстань від нього до себе рівною нулю.
2. Оберіть вузол з мінімальною відстанню від початкового вузла. Позначте його як відвіданий.
3. Для кожного сусіднього вузла, який ще не відвіданий, перерахуйте відстань від початкового вузла через поточний вузол. Якщо ця відстань коротша, оновіть відстань.
4. Повторюйте кроки 2 і 3, поки всі вузли не будуть позначені як відвідані.

Алгоритм завершується, коли всі вузли позначені як відвідані. На цей момент будуть знайдені найкоротші шляхи від початкового вузла до всіх інших вузлів.

## 3. Алгоритм А\*

Алгоритм А\* – алгоритм пошуку найкоротшого шляху в графі. Він комбінує в собі евристичні методи та пошук з найкоротшим шляхом. А\* використовує оцінки відстаней до кінцевої точки разом з вартістю пройденого шляху, щоб визначити, яку вершину розглядати наступною [12]. До основних кроків даного алгоритму можна віднести:

1. Встановіть початковий вузол. Позначте його як поточний. Встановіть вартість шляху від початкової вершини до нього та оцінку відстані до кінцевої точки (евристична оцінка).

2. Повторюйте кроки пошуку, доки не досягнете кінцевої точки або не перевірите всі можливі шляхи. Обчисліть оцінку "ціни" пройденого шляху плюс оцінку відстані до кінцевої точки (евристичну оцінку) для кожної сусідньої вершини. Оберіть сусідню вершину з найменшою оцінкою. Позначте обрану вершину як поточну.

3. Коли досягнута кінцева точка або відсутній шлях, або всі можливі шляхи перевірені, завершіть алгоритм.

## 4. Алгоритм Беллмана-Форда

Алгоритм Беллмана-Форда – це алгоритм пошуку найкоротших шляхів від одного визначеного вузла до всіх інших вузлів взваженого напрямленого графа, навіть якщо в графі присутні ребра з від'ємними вагами.

До кроків даного алгоритму можна віднести:

1. Встановлення початкового вузла та встановлення відстаней до всіх інших вузлів як нескінченності, за винятком відстані до самого себе, яка рівна нулю.

2. Повторюється  $|M|-1$  разів де  $M$  – кількість вузлів у графі. На кожній ітерації проходиться по всім ребрам графа і оновлюються відстані до кожного вузла, якщо можна скоротити відстань, пройшовши через поточне ребро.

3. Після  $|M|-1$  ітерацій перевіряється, чи існують від'ємні цикли. Якщо відстань до будь-якого вузла може бути зменшена після ще однієї ітерації, це означає, що у графі є від'ємний цикл, який нескінченно зменшує відстань до певного вузла [13, 14].

Цей алгоритм дозволяє знайти найкоротші шляхи в графі, включаючи графи з від'ємними вагами, але він може бути менш ефективним у порівнянні з іншими алгоритмами, такими як Дейкстри або А\*, через свою складність

## 5. Алгоритм Флойда-Уоршелла

Алгоритм пошуку всіх найкоротших шляхів між кожною парою вершин в орієнтованому або неорієнтованому зваженому графі. Цей алгоритм найбільш ефективний для використання на графах зі зваженими ребрами, що можуть містити від'ємні ваги. Основна ідея алгоритму полягає в тому, щоб поступово покращувати відомі шляхи між вершинами, додаючи нові проміжні вершини до поточного найкращого шляху. Даний алгоритм працює коректно, якщо у графі немає циклів негативної величини, а у разі,

коли такий цикл є, дозволяє знайти хоча б один такий цикл [15].

Складається з таких кроків як:

1. Встановити вагу прямих ребер між вершинами як вагу відповідного ребра. Якщо прямого ребра немає, вага встановлюється як нескінченність. Також встановлюємо вагу вершини до себе самої як 0.

2. Провести ітерації по всім парам вершин, де для кожної пари перевіряємо, чи можливо покращити шлях між цими вершинами, проходячи через поточну вершину. Якщо так, оновлюємо вагу шляху.

3. Повторюємо цей процес для кожної вершини як проміжної вершини у шляху, поки не досягнемо оптимального розв'язку.

4. Після завершення алгоритму ми матимемо матрицю, в якій кожен елемент буде містити вагу найкоротшого шляху від вершини  $u$  до вершини  $v$ .

Алгоритм Флойда-Уоршелла гарантує знаходження коректних найкоротших шляхів між усіма парами вершин у графі.

Однак його складність становить  $O(M^3)$ , що робить його неефективним для великих графів.

### Висновок

Досліджено існуючих методів пошуку оптимального шляху в залежності від особливостей вхідних даних та цілей оптимізації.

Розглянуті алгоритми на основі графів та їх евристики.

У роботі було проведений аналіз сфери застосування наведених алгоритмів.

Визначені конкретні переваги та недоліки використання алгоритмів в залежності від області використання.

### СПИСОК ЛІТЕРАТУРИ

- Barnes E. R. An algorithm for partitioning the nodes of a graph // *SIAM J. Algebraic Discrete Methods*. – 1982. – Vol. 4, no. 3. – P. 541-550.
- Коваленко А. А., Кучук Г. А. Методи синтезу інформаційної та технічної структур системи управління об'єктом критичного застосування. *Сучасні інформаційні системи*. 2018. Т. 2, № 1. С. 22–27. DOI: <https://doi.org/10.20998/2522-9052.2018.1.04>
- Свиридов А. С., Коваленко А. А., Кучук Г. А. Метод перерозподілу пропускної здатності критичної ділянки мережі на основі удосконалення ON/OFF-моделі трафіку. *Сучасні інформаційні системи*. 2018. Т. 2, № 2. С. 139–144. DOI: <https://doi.org/10.20998/2522-9052.2018.2.24>
- Главчева Ю. М., Главчев М. І., Канишева О. В., Кучук Г. А. Розробка підходу для ранжування академічних установ за показниками наукової діяльності. *Сучасні інформаційні системи*. 2019. Т. 3, № 1. С. 63–70.
- Dun, B., Zakovorotnyi, O. and Kuchuk, N. (2023), "Generating currency exchange rate data based on Quant-Gan model", *Advanced Information Systems*, Vol. 7, no. 2, pp. 68–74, doi: <https://doi.org/10.20998/2522-9052.2023.2.10>
- Datsenko, S. and Kuchuk, H. (2023), "Biometric authentication utilizing convolutional neural networks", *Advanced Information Systems*, Vol. 7, no. 2, pp. 87–91, doi: <https://doi.org/10.20998/2522-9052.2023.2.12>
- Petrovska, I. and Kuchuk, H. (2023), "Adaptive resource allocation method for data processing and security in cloud environment", *Advanced Information Systems*, Vol. 7, No. 3, pp. 67–73, doi: <https://doi.org/10.20998/2522-9052.2023.3.10>
- Krepych, S., & Spivak, I. (2021). Improvement of svd algorithm to increase the efficiency of recommendation systems. *Advanced Information Systems*, 5(4), 55–59. <https://doi.org/10.20998/2522-9052.2021.4.08>
- Кучук Г.А. Мінімізація завантаження каналів зв'язку обчислювальної мережі / Г.А. Кучук // Системи обробки інформації. – Х.: НАНУ, ПАНМ, ХВУ, 1998. – Вип. 1(5). – С. 149-154.
- Shmatko, O., Kolomiitsev, O., Reкова, N., Kuchuk, N. and Matvieiev, O. (2023), "Designing and evaluating dl-model for vulnerability detection in smart contracts", *Advanced Information Systems*, vol. 7, no. 4, pp. 41–51. doi: <https://doi.org/10.20998/2522-9052.2023.4.05>.
- Fakcharoenphol J., Rao S. Planar graphs, negative weight edges, shortest paths, and near linear time // *Proc. 42nd IEEE Symp. Foundations of Computer Science*. – 2001. – P. 232-241.
- Goldberg A. V., Harrelson C. Computing the shortest path: A\*-search meets graph theory // *Proc. Sixteenth Annual ACM-SIAM Symp. on Discrete Algorithms*, January 23-25, Vancouver, BC (2005). – P. 156-165.
- Timothy M. Chan. More algorithms for all-pairs shortest paths in weighted graphs. In *STOC07*, pages 590–598, 2007.
- Seth Pettie. A new approach to all-pairs shortest paths on real-weighted graphs. *Theoretical Computer Science*, 312:47–74, 2004.
- Edward M. Reingold, Jurg Nievergelt, and Narsingh Deo. *Combinatorial Algorithms: Theory and Practice*. Prentice-Hall, Inc., 1977.

Received (Надійшла) 20.02.2024

Accepted for publication (Прийнята до друку) 19.04.2024

### Study of algorithms for searching the optimal path

S. Bulba, O. Solovyova, Y. Semerenko

**Abstract.** Extensive areas of application of optimal path search algorithms lead to the need for their thorough research. The purpose of this work is to conduct a study of existing methods of finding the optimal path depending on the characteristics of input data and optimization goals. The article considers such types of algorithms as: algorithms based on graphs and based on heuristics. In the work, an analysis of the scope of application of the above algorithms was carried out. The advantages and disadvantages of using algorithms are defined depending on the area of use. The complexity of the application, depending on the computing power and the time of processing the input data, was studied. The presented studies make it possible to obtain information about the reliability of the results of the considered algorithms depending on the input set of parameters for calculation.

**Keywords:** algorithm, programming, distribution, graph, optimal path, artificial intelligence.