

СУЧАСНІ МОДИФІКАЦІЇ АЛГОРИТМУ ДЕЙКСТРИ В КОМП'ЮТЕРНИХ МЕРЕЖАХ

Вінницький національний технічний університет

Анотація

У роботі розглядається алгоритм Дейкстри — класичний метод пошуку найкоротших шляхів у графах з невід'ємними вагами. Проаналізовано сучасні модифікації алгоритму, що застосовуються в оптичних, супутникових, центрових та високопродуктивних мережах. Показано, що різні варіанти алгоритму дозволяють у декілька разів підвищувати швидкість маршрутизації та ефективність використання мережевих ресурсів. Наведено порівняльну таблицю адаптацій Дейкстри та досліджено їх актуальність для сучасних інженерних задач.

Ключові слова: графи, найкоротший шлях, алгоритм Дейкстри, маршрутизація, оптичні мережі, супутникові мережі, паралельні обчислення.

Abstract

The paper considers Dijkstra's algorithm, a classic method for finding the shortest paths in graphs with non-negative weights. Modern modifications of the algorithm used in optical, satellite, centralised, and high-performance networks are analysed. It is shown that different variants of the algorithm allow to increase the routing speed and the efficiency of network resource utilisation several times over. A comparative table of Dijkstra's adaptations is provided and their relevance for modern engineering tasks is investigated.

Keywords: graphs, shortest path, Dijkstra's algorithm, routing, optical networks, satellite networks, parallel computing.

Вступ

Алгоритм Дейкстри був запропонований Едсгером Дейкстрою у 1956 році та опублікований у 1959-му. З того часу метод став фундаментом маршрутизації в комп'ютерних мережах, навігаційних системах, оптичних мережах і багатьох інших технологіях.

У класичному варіанті алгоритм знаходить найкоротший шлях від однієї вершини графа до всіх інших і має складність $O(m + n \log n)$ при використанні черги з пріоритетами. Його простота та оптимальність зробили його основою протоколів OSPF, IS-IS, SDN-систем та WAN-трафік-інженерії.

Однак сучасні мережі — оптичні, супутникові, дата-центрові — потребують швидших та адаптивних рішень, тому було створено низку модифікацій алгоритму Дейкстри.

Результати дослідження

Таблиця 1. Основні модифікації алгоритму Дейкстри та їхні особливості

Модифікація	Ключова ідея	Переваги	Застосування
Generic Dijkstra	Повторне відвідування вершин для врахування спектральних обмежень	Швидше за класичний метод на $\approx 2.3\times$	WDM/EON оптичні мережі
Adapted Dijkstra (EON)	Додає обмеження континуїтету та суміжності спектру	Ймовірність успіху з'єднання $\uparrow$ до $2\times$	Еластичні оптичні мережі
Percolation-Dijkstra	Зменшення області пошуку через перколяцію	Менше пам'яті та швидше виконання	Супутникові мегаконстеляції
Parallel Dijkstra (MPI/CUDA)	Розподіл роботи між процесами або GPU	Прискорення $5\times$ – $10\times$	Дата-центри, HPC
Δ -stepping	Паралельне розслаблення ребер	Висока ефективність на кластерах	Великі розподілені мережі

Дослідження, проведені Szcześniak у 2018–2019 роках, показують, що адаптовані версії алгоритму для оптичних мереж дозволяють знаходити маршрути, які класичний Дейкстра фізично не може обчислити через спектральні обмеження [1]. У роботі Luan (2024) доведено, що алгоритм Percolation-Dijkstra забезпечує таку ж точність, як і heap-оптимізований Дейкстра, але споживає значно менше пам'яті, що критично для супутникових платформ з обмеженими ресурсами [2]. Паралельні реалізації алгоритму на GPU демонструють прискорення до десяти разів у великих симуляціях, що є важливим для SDN-контролерів та систем реального часу; відповідні результати були отримані у дослідженнях Kainer (2019) та Song (2025) [3-4].

У сучасних комп'ютерних мережах алгоритми пошуку найкоротших шляхів відіграють ключову роль у забезпеченні ефективної маршрутизації та раціонального використання ресурсів. В оптичних мережах типу WDM та еластичних EON особливе значення має мінімізація затримки, підтримка спектральної безперервності та зменшення втрат пропускну здатності. Для таких систем ефективними є алгоритми CSPF, FDA та адаптовані варіанти Дейкстри, ефективність яких підтверджено експериментальними дослідженнями Amphawan (2012) [5]. У масштабних WAN-мережах у межах трафік-інженерії, зокрема в архітектурі Segment Routing, показано, що використання невеликої кількості центральних вузлів як середньоточок може суттєво скоротити час оптимізації без втрати якості маршрутизації, про що свідчать результати Trimonias (2017) [6].

У супутникових мегаконстеляціях, що містять тисячі вузлів та працюють в умовах обмежених обчислювальних ресурсів, перспективними є легковагові варіанти алгоритму, такі як Percolation-Dijkstra, який демонструє кращу продуктивність порівняно з класичним методом при розрахунках у режимі реального часу [2]. У мережах дата-центрів та HPC-кластерах поширені паралельні алгоритми найкоротших шляхів, реалізовані за допомогою MPI, CUDA або моделей PRAM. Експериментальні вимірювання показують, що такі методи забезпечують прискорення до п'яти разів при використанні MPI та понад десятикратне - у GPU-реалізаціях [3-4]. Крім того, алгоритм Δ -stepping продемонстрував високу масштабованість і наближення до теоретичних меж продуктивності, що робить його доцільним для систем з високим навантаженням.

Таким чином, вибір конкретної модифікації алгоритму пошуку найкоротшого шляху залежить від типу мережі: у невеликих та середніх топологіях достатньо класичного Дейкстри з пріоритетною чергою; в оптичних WDM/EON-системах доцільно застосовувати Generic або Adapted Dijkstra [1]; для великих графів та трафік-інтенсивних інфраструктур оптимальними є паралельні методи з використанням CUDA або MPI [3-4], тоді як для супутникових систем найбільш ефективним є Percolation-Dijkstra [2]. У SDN-архітектурах важливою стає масштабованість, тому часто застосовують багаторівневі моделі або побудову графа-ядра (distance backbone), що дозволяє зменшити кількість оброблюваних ребер без втрати точності.

Висновки

Алгоритм Дейкстри залишається основою маршрутизації в сучасних комп'ютерних мережах. Збільшення масштабів мереж, поява оптичних технологій та супутникових систем зумовили створення нових, ефективніших модифікацій класичного алгоритму. Аналіз показує, що залежно від архітектури мережі різні адаптації можуть підвищувати ефективність від $2\times$ до $10\times$, що робить їх важливою частиною інфраструктури наступного покоління.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Szcześniak I., Jajszczyk A., Woźna-Szcześniak B. Generic Dijkstra for Optical Networks [Електронний ресурс] // Journal of Optical Communications and Networking. – 2019. – Vol. 11, № 11. – P. 568–577. – Режим доступу: <https://opg.optica.org/jocn/fulltext.cfm?uri=jocn-11-11-568> (дата звернення: 10.12.2025).
2. Luan S. A Fast Percolation-Dijkstra Routing Method for Mega-Constellation Backbone Networks [Електронний ресурс] // arXiv. – 2024. – Режим доступу: <https://arxiv.org/abs/2404.00904> (дата звернення: 11.12.2025).
3. Kainer M., Träff J. More Parallelism in Dijkstra's Single-Source Shortest Path Algorithm [Електронний ресурс] // arXiv. – 2019. – Режим доступу: <https://arxiv.org/abs/1903.12085> (дата звернення: 10.12.2025).

4. Song B. High-Performance Parallelization of Dijkstra's Algorithm Using MPI and CUDA [Електронний ресурс] // arXiv. – 2025. – Режим доступу: <https://arxiv.org/abs/2504.03667> (дата звернення: 9.12.2025).
5. Amphawan A., Md Khair M. A., Hasan H. Multimedia Traffic Routing in Multilayer WDM Networks [Електронний ресурс] // arXiv. – 2012. – Режим доступу: <https://arxiv.org/abs/1207.0425> (дата звернення: 8.12.2025).
6. Trimponias G., Xiao Y., Xu H., Wu X., Geng Y. Centrality-based Middlepoint Selection for Traffic Engineering with Segment Routing [Електронний ресурс] // arXiv. – 2017. – Режим доступу: <https://arxiv.org/abs/1703.05907> (дата звернення: 12.12.2025).

Науковий керівник: **Добровольська Наталія Вікторівна** – к.пед.н., доцент кафедри обчислювальної техніки, Вінницький національний технічний університет, Вінниця, e-mail: doabr_n_v@vntu.edu.ua

Сірак Владислав Олегович – студент групи 1KI-25МС, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: vladsirak9@gmail.com

Supervisor: **Dobrovolska Nataliia V.** – Ph.D., Associate Professor of the Computer Engineering Department, Vinnytsia National Technical University, Vinnytsia, e-mail: doabr_n_v@vntu.edu.ua

Sirak Vladislav O. – student of group 1KI-25MS, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: vladsirak9@gmail.com