

УДК 004.58

## РОЗРОБКА АЛГОРИТМУ ПОШУКУ ОПТИМАЛЬНОГО МАРШРУТУ У ЗАКРИТИХ ПРИМІЩЕННЯХ

Лисенко М. С.<sup>1</sup>

<sup>1</sup>*Державний університет «Житомирська політехніка»*

**Анотація:** Розвиток інклюзивного середовища є важливим аспектом сучасного суспільства, зокрема у сфері навігаційних систем для внутрішніх приміщень. Багато існуючих алгоритмів пошуку найкоротшого шляху, таких як класичний алгоритм Дейкстри, не враховують специфічні потреби користувачів з обмеженими можливостями. Це може призводити до створення маршрутів, які є непридатними або незручними для людей на інвалідних візках, осіб з порушенням зору, а також для тих, хто має тимчасові труднощі з пересуванням. Особливо критичною така проблема є в медичних закладах, де швидкість і доступність переміщення можуть безпосередньо впливати на якість обслуговування та безпеку пацієнтів.

У цій роботі пропонується модифікація алгоритму Дейкстри, яка дозволяє динамічно адаптувати ваги ребр графа відповідно до особливих потреб користувачів та поточних умов середовища. Запропонований метод використовує продукційну модель знань, що дає змогу системі приймати рішення на основі формалізованих правил, враховуючи параметри доступності, зокрема наявність ліфтів, пандусів, ширину дверних прорізів, освітлення, тактильні позначки та інші фактори. Розроблена адаптивна навігаційна система, що реалізована на основі модифікованого алгоритму Дейкстри, має такі ключові переваги: можливість персоналізації маршрутів відповідно до профілю користувача, автоматичне виключення недоступних або малоприсаєтних шляхів, динамічне коригування маршруту в разі змін середовища (наприклад, ремонтних робіт або несправності ліфтів).

Запропонований алгоритм може бути використаний для оптимізації навігації не лише в медичних закладах, а й у торговельних центрах, навчальних установах, транспортних вузлах та інших будівлях з інтенсивним потоком людей. Результати проведеного дослідження свідчать, що впровадження адаптивного підходу до пошуку маршрутів підвищує рівень доступності та зручності навігації для всіх категорій користувачів. Подальший розвиток системи може включати інтеграцію технологій штучного інтелекту для автоматичного аналізу змін у середовищі, а також розширення алгоритму для роботи у відкритих просторах, зокрема у міських навігаційних системах.

**Ключові слова:** адаптивність, алгоритм Дейкстри, граф, інклюзивність, користувачі, медичні заклади, навігація, маршрут, пошук шляху, приміщення, продукційна модель представлення знань, система, технології.

## DEVELOPMENT OF AN ALGORITHM FOR SEARCHING FOR THE OPTIMAL ROUTE IN INDOOR PREMISES

M. Lysenko<sup>1</sup>

<sup>1</sup>*State University "Zhytomyr Polytechnic"*

**Abstract:** The development of an inclusive environment is a crucial aspect of modern society, particularly in the field of indoor navigation systems. Many existing shortest path algorithms, such as the classic Dijkstra's algorithm, do not take into account the specific needs of users with disabilities. This can lead to the creation of routes that are unsuitable or inconvenient for wheelchair users, individuals with visual or hearing impairments, and those with temporary mobility difficulties. This



issue is especially critical in medical facilities, where the speed and accessibility of movement can directly impact the quality of service and patient safety.

This paper presents a modification of Dijkstra's algorithm that allows for the dynamic adaptation of graph weights according to the specific needs of users and the current environmental conditions. The proposed approach integrates a production knowledge model, enabling the algorithm to make decisions based on formalized rules that consider accessibility parameters such as the availability of elevators, ramps, doorway widths, lighting, tactile markings, and other factors.

The adaptive navigation system, built on the basis of the modified Dijkstra's algorithm, offers several key advantages: the ability to personalize routes according to the user's profile, automatic exclusion of inaccessible or unsuitable paths, and dynamic route adjustment in response to environmental changes (e.g., construction work or elevator malfunctions). The proposed algorithm can be used to optimize navigation not only in medical facilities but also in shopping centers, educational institutions, transportation hubs, and other buildings with high pedestrian traffic.

The results of the study indicate that implementing an adaptive approach to route searching significantly enhances accessibility and convenience for all categories of users. Future system development may include the integration of artificial intelligence technologies for real-time analysis of environmental changes, as well as extending the algorithm for use in open spaces, particularly in urban navigation systems.

**Keywords:** adaptivity, building, Dijkstra's algorithm, graph, inclusivity, medical facilities, navigation, pathfinding, production knowledge model, route, system, technologies, users.

## 1 ВСТУП

Сучасні інженерні системи стають все більш складними та динамічними. Вимоги до їх продуктивності, надійності та ефективності зростають, що формує потребу у нових підходах до проектування та управління цими системами. Традиційні алгоритми не завжди можуть ефективно справлятися зі змінами в середовищі або вимогами до систем і саме тому адаптивні алгоритми, які можуть змінювати свою поведінку в залежності від умов, дозволяють більш ефективно вирішувати завдання в динамічних середовищах.

Концепція роєвого інтелекту, яка базується на поведінці колективів біологічних організмів (наприклад, мурах або птахів), є перспективним підходом для розробки адаптивних алгоритмів. Використання роєвого інтелекту дозволяє створювати розподілені, гнучкі та стійкі системи управління. У багатьох сферах, таких як робототехніка, енергетика, транспорт та інші, інженерні системи потребують ефективних методів управління для підвищення продуктивності та зниження витрат і розробка нових алгоритмів на основі роєвого інтелекту може значно покращити ці процеси. Впровадження адаптивних алгоритмів роєвого інтелекту у проектування та управління інженерними системами має великий потенціал для підвищення їх ефективності, надійності та стійкості, що призводить до покращення роботи існуючих систем та створення нових, більш досконалих рішень. Таким чином, дослідження з розробки адаптивних алгоритмів роєвого інтелекту є надзвичайно актуальним та важливим для розвитку сучасних інженерних систем і має значний потенціал для практичного застосування у різних галузях.

## 2 АНАЛІЗ ЛІТЕРАТУРНИХ ДАНИХ ТА ПОСТАНОВКА ПРОБЛЕМИ

Сучасні навігаційні системи для будівель представлено в ряді робіт закордонних та вітчизняних науковців. Зупинимось на кількох з них.

Авторами [1] представлено основні принципи трьох алгоритмів пошуку найкоротшого шляху та проведено їхнє порівняння шляхом аналізу часової та просторової складності. Крім того, узагальнено область застосування різних алгоритмів. Алгоритм Дейкстри – це класичний алгоритм отримання найкоротшого шляху від конкретної вершини до будь-якої іншої. Його широко використовують в дорожніх маршрутах. Цей алгоритм можна використовувати лише тоді, коли у графі не існує жодного ребра з від'ємною вагою. Алгоритм Беллмана-Форда можна використовувати на графах з від'ємною вагою ребр, якщо граф не містить негативного циклу, доступного з вихідної вершини. Результат роботи цього алгоритму можна використати для визначення існування циклу від'ємної ваги у графі. Алгоритм Флойда-Уоршелла – це алгоритм динамічного програмування, який може вирішити проблему найкоротшого шляху між будь-якими двома вершинами. Метод використовується на зважених графах, у яких можуть бути як додатні, так і від'ємні ваги ребр, проте у ньому не має бути від'ємних циклів. Таким чином, цей метод загальніший у порівнянні з алгоритмом Дейкстри. Однак, у практиці ці три алгоритми безпосередньо не застосовуються, а проводиться їх модифікація та оптимізація для підвищення ефективності.

У дослідженні [2] авторами порівнювалися алгоритми пошуку шляху  $A^*$ , Дейкстри та пошук у ширину (BFS) у грі Maze Runner. Процес порівняння цих алгоритмів полягав у заміні алгоритму в грі та вимірюванні часу процесу, довжини маршруту та кількості блоків, задіяних в існуючому обчислювальному процесі.

Luay S. Jabbar, Eyad I. Abbas, Sundus D. Hasanb у своєму дослідженні [3] висвітлюють процес впровадження ретроактивної структури даних черги пріоритетів,

що робить алгоритм Дейкстри динамічним. У цьому дослідженні було змінено підхід, спрямований на пересування вперед у найкоротшому напрямку для двох мереж та двох маршрутів. Спочатку знаходиться статичний найкоротший шлях, потім здійснюється перехід до першого вузла після початкового. У цьому вузлу ваги у сегментах, спрямованих із цього вузла, змінюються, що анулює попередній найкоротший шлях та дозволяє знайти новий маршрут. Потім здійснюється перехід до наступного вузла у знайденому найкоротшому шляху, і операція повторюється до завершення. Ідея полягає в тому, що найкращий маршрут може постійно змінюватися на основі останніх даних. Ці безперервні зміни розглядаються у цій роботі, де запропонована система здатна знаходити оптимальні маршрути та автоматично оновлювати їх відповідно до змінних параметрів.

Авторами дослідження [4] було реалізовано метод алгоритму Дейкстри для пошуку найкоротшого шляху в GIS. Технологія GIS використовує алгоритм Дейкстри для надання інформації, візуально відображаючи дані про місцезнаходження доріг, місць та найкоротшого маршруту від початкового до кінцевого пункту на карті. Геоінформаційна система пошуку найкоротшого маршруту відображає візуалізацію просторових даних карт промислових зон, інтегрованих із GoogleMaps.

Автори [6] досліджують застосування алгоритму Дейкстри для знаходження найкоротшого шляху в дорожніх мережах. Вони підкреслюють важливість вирішення цієї задачі для міських транспортних систем, навігаційних програм та оптимізації маршрутів. У роботі розглядаються основи графової теорії, методологія алгоритму та його ефективність. Основна увага приділяється класичному підходу до вирішення задачі пошуку найкоротшого шляху для статичних графів із невід'ємними вагами. Крім теоретичного опису, автори наводять експериментальні результати, що демонструють застосування алгоритму на реальних дорожніх мережах. Використання мови програмування Java для візуалізації результатів дозволяє оцінити ефективність алгоритму. Висновки статті вказують на потенціал алгоритму Дейкстри, проте визнають необхідність покращень для динамічних та великих мереж.

Стаття [7] досліджує використання алгоритму Дейкстри для визначення найкоротшого маршруту між туристичними локаціями. Автори акцентують увагу на оптимізації подорожей за допомогою геоінформаційних технологій, що дозволяють скорочувати відстань, час та витрати на поїздки. Дослідження охоплює процес розробки веб-застосунку, заснованого на алгоритмі Дейкстри, з мобільною адаптацією. Описано етапи створення системи, включаючи збір даних, проектування за допомогою UML, реалізацію та тестування. Результати експериментального аналізу показують ефективність алгоритму для статичних маршрутів, але наголошується на необхідності врахування динамічних змін, таких як дорожні затори. Автори рекомендують розширення функціоналу застосунку шляхом інтеграції даних про трафік.

Стаття [8] досліджує вдосконалення алгоритму Дейкстри для навігації під час евакуації. Авторами запропоновано систему автономної евакуації (AEN), що використовує динамічні вказівники для спрямування людей найкоротшим і водночас безпечним маршрутом. Основні модифікації алгоритму включають блокування небезпечних вузлів (наприклад, зон займання) та обмеження напрямків руху. Методологія включає побудову графа будівлі, розміщення вузлів та обчислення шляхів. Аналіз доводить ефективність змін у виборі безпечного маршруту. Дослідження актуальне для евакуаційних систем, покращуючи швидкість і точність маршрутизації. Подальші дослідження можуть спрямовуватися на масштабованість алгоритму для великих споруд.

Дослідження [9] присвячене застосуванню алгоритму Дейкстри для визначення найкоротшого шляху в графі з динамічною вагою ребер, що моделює дорожню

мережу. Автори розглядають одно- та двосторонні орієнтовані графи. Експерименти проводилися на двох типах карт (Мар А та Мар В), що відрізняються розподілом ваги ребер. Результати показали ефективність алгоритму Дейкстри для односторонніх графів, але виявили проблеми з циклічністю у двосторонніх графах, що призводило до неможливості знайти рішення у 17,5% випадків. Дослідження також оцінювало вплив динамічної зміни ваги на час обчислення, показавши незначний вплив на швидкість роботи алгоритму. Автори обговорюють причини виникнення циклів у двосторонніх графах та пропонують шляхи їх усунення. Загалом, робота демонструє практичну застосовність алгоритму Дейкстри для задач маршрутизації з динамічною вагою, але підкреслює необхідність врахування особливостей структури графу при його використанні.

Стаття [10] досліджує систему навігації для незрячих та користувачів з вадами зору університетського кампусу. Основою системи є модифікований алгоритм Дейкстри, що працює з графовою структурою даних, де вузли представляють локації (аудиторії, коридори тощо), а ребра – шляхи між ними з детальними описами. Система враховує просторові координати та кути поворотів, забезпечуючи точні та зрозумілі інструкції. Використання алгоритму Дейкстри гарантує знаходження найкоротшого шляху, а додавання контекстних описів та обчислення кутів поворотів робить навігацію більш інтуїтивною та зручною для користувачів. Інтерфейс системи адаптований для роботи з програмним забезпеченням для читання екрану, що робить її доступною для людей з вадами зору. Автори стверджують, що система значно покращує автономність та безпеку пересування незрячих людей в університетському середовищі. Дослідження демонструє успішне застосування алгоритму Дейкстри в контексті навігації для людей з вадами зору, підкреслюючи важливість врахування просторових особливостей та контекстуальної інформації для ефективної навігації.

### 3 ЦІЛІ ТА ЗАВДАННЯ ДОСЛІДЖЕННЯ

Основною метою даного дослідження є розробка алгоритму пошуку оптимальних маршрутів у закритих приміщеннях шляхом модифікації алгоритму Дейкстри для врахування потреб різних категорій користувачів. Особлива увага приділяється інклюзивності, тобто забезпеченню доступності маршрутів для осіб із обмеженими можливостями, зокрема людей на інвалідних візках, осіб із порушенням зору, людей похилого віку, осіб, що мають тимчасові труднощі з пересуванням (наприклад, травми).

Для реалізації пошуку оптимального маршруту необхідним є впровадження продукційної моделі знань, яка дозволить формалізувати правила й умови для вибору. Продукційна модель знань дасть змогу алгоритму динамічно змінювати критерії оптимальності залежно від профілю користувача, інтегрувати додаткові знання про середовище (наприклад, тимчасово недоступні ліфти чи заблоковані зони), забезпечити гнучкість і адаптивність системи до нових умов.

Результатом дослідження має стати ефективний адаптивний алгоритм, який можна інтегрувати в автоматизовані системи навігації. Такий алгоритм сприятиме поліпшенню користувацького досвіду всередині будівель, особливо в умовах медичних закладів, де точність і швидкість навігації мають критичне значення.

### 4 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Слід зазначити, що алгоритм Дейкстри є одним із найпоширеніших методів пошуку найкоротшого шляху в графі, що працює шляхом ітеративного визначення мінімальної відстані від стартової вершини до всіх інших вершин графа. Серед ключових характеристик даного алгоритму слід виділити використання пріоритетної черги для вибору вершини з найменшою вагою та статичний підхід, який передбачає



незмінність ваги ребр під час виконання алгоритму. Однак у випадку інклюзивної навігації всередині будівель цей підхід має суттєві обмеження, оскільки не враховує специфічних потреб користувачів та динамічних змін у доступності маршрутів.

Адаптація алгоритму Дейкстри, що пропонується, передбачає кілька ключових змін для врахування параметрів інклюзивності. Пропонується запровадити динамічне коригування ваги ребр. Вага кожного ребра графа (що відповідає сегменту маршруту) визначається не лише довжиною шляху, але й доступністю сегмента для різних категорій користувачів (наприклад, сходи можуть отримати дуже велику вагу для людей на візках), характеристиками середовища (наприклад, погане освітлення додає вагу для людей із вадами зору) та поточним станом елементів інфраструктури (несправність ліфта, ремонт пандуса).

Процес побудови графа для внутрішньої навігації передбачає формальне представлення просторової структури будівлі у вигляді графової моделі  $G = (V, E)$ , де множина вершин  $V$  відповідає ключовим локаціям, таким як приміщення, дверні прорізи, ліфти, а множина ребер  $E$  описує можливі шляхи переміщення між ними, наприклад, коридори, сходові марші та пандуси.

Множина вершин  $V$  являє собою відкрите класифікаційне угруповання та включає в себе важливі для переміщення точки будівлі  $v$ , яким присвоєно унікальний ідентифікатор  $i, v_i \in V$ , та координати  $(x_i, y_i)$ , що відповідають його розташуванню на плані будівлі. Кожна вершина  $v_i$  є вузлом зваженого графу, якій притаманні наступні властивості: тип  $t$  – кімната ( $t_1$ ), двері ( $t_2$ ), сходи ( $t_3$ ), ліфт ( $t_4$ ), пандус ( $t_5$ ), тощо; доступність (accessibility)  $\alpha$  – чи доступний вузол для певних груп користувачів (наприклад, користувачів в інвалідних візках); інші атрибути (mark)  $m$  – ширина дверей, умови освітлення, тактильні позначки тощо.

Кожне ребро  $e_{i,j} \in E$  має вагу ребра  $\omega_{ij}$ . Вага ребра  $\omega_{ij}$  визначається за допомогою формули (1):

$$\omega_{ij} = d_{ij} + p_{ij}^k + s_{ij}, \quad (1)$$

де:  $d_{ij}$  – фізична відстань між вузлами  $v_i$  та  $v_j$ ;  $p_{ij}^k$  – вага ребра відповідно профілю користувача, яка визначає доступність ребра для певних категорій користувачів (задовольняє потребам користувача);  $s_{ij}$  – динамічна вага ребра, пов'язана зі станом середовища (наприклад, тимчасові перешкоди, ремонт, несправності ліфту).

Кожен  $k$ -тий користувач системи має власний профіль  $p^k$ , який визначає тип потреб (“без особливостей”, “маломобільність”, “порушення зору”) та індивідуальні пріоритети (мінімізація часу або зусиль). Профіль  $p_{ij}^k$  впливає на вагу кожного ребра та дозволяє адаптувати маршрут під конкретного користувача. Алгоритм пропонує кілька альтернативних маршрутів, якщо оптимальний за відстанню шлях недоступний через обмеження, то для цього в пріоритетну чергу включаються маршрути з мінімальним компромісом між довжиною і доступністю.

Для врахування параметрів інклюзивності використовуємо наступні показники, а саме, доступність ліфтів, наявність пандусів, освітлення та маркування, ширину проходів, тактильні позначки.

Ліфти мають пріоритет для користувачів на інвалідних візках або з вадами опорно-рухового апарату (профіль користувача – «маломобільність»). У разі несправності ліфта система автоматично додає альтернативний маршрут із використанням пандусів або сходів із мінімальними перешкодами.

Пандуси розглядаються як пріоритетні елементи маршруту для маломобільних користувачів. У разі їх наявності система збільшує вагу ребер, які включають сходи без пандусів.

Для користувачів на візках система виключає вузькі проходи та двері, якщо їх ширина менша за встановлені стандарти доступності.

Для людей із порушенням (вадами) зору вагу ребер збільшують у випадку недостатнього освітлення чи відсутності тактильного маркування.

Граф має динамічно оновлюватися на основі даних про навколишнє середовище в реальному часі, таких як: статус ліфтів (працюють чи не працюють); наявність тимчасових перешкод (будівництво, ремонт або заблоковані шляхи); пішохідний рух (зони інтенсивного руху, які можуть спричинити затримки). Ці дані використовуються для врахування значення  $s_{ij}$  в режимі реального часу.

Для реалізації (запуску) алгоритму потрібно виконати ряд послідовних дій, зокрема, **першим етапом** є формування графу будівлі  $G = (V, E)$ . Для цього проводиться ініціалізація вузлів, а саме створюємо вузол  $v_i$  для кожного ключового місця в будівлі, які приймають участь у переміщенні пацієнтів по території медичного закладу. Фіксуємо координати  $(x_i, y_i)$  кожного  $v_i$ -го вузла. Враховуючи результати проведеного аналізу споруд медичних закладів, використання адаптивних алгоритмів, пропонуємо створити наступну множину вузлів  $V$ , яка буде об'єднувати наступні підмножини:

- підмножину  $A$  – приміщення: кабінети, зони очікування, санітарні вузли;
- підмножину  $B$  – двері: точки входу/виходу, дверні прорізи між приміщеннями;
- підмножину  $C$  – сходи: міжповерхові переходи;
- підмножину  $L$  – ліфти: шахти ліфтів та точки їх входу/виходу.
- підмножину  $H$  – пандуси: похилі доріжки для забезпечення доступності.
- підмножину  $G$  – перехрестя коридорів: точки злиття чи розгалуження маршрутів.

Визначаємо, що  $A \subseteq V, B \subseteq V, C \subseteq V, L \subseteq V, H \subseteq V, G \subseteq V$ .

При чому елементи вказаних підмножин не належать іншим підмножинам, тобто:

$$\begin{aligned} A \cap B = \emptyset, A \cap C = \emptyset, A \cap L = \emptyset, A \cap H = \emptyset, A \cap G = \emptyset, B \cap C = \emptyset, B \cap L = \emptyset, \\ B \cap H = \emptyset, B \cap G = \emptyset, C \cap L = \emptyset, C \cap H = \emptyset, C \cap G = \emptyset, L \cap H = \emptyset, L \cap G = \emptyset, \\ H \cap G = \emptyset. \end{aligned}$$

Таким чином

$$V = A \cup B \cup C \cup L \cup H \cup G. \quad (2)$$

Множина вузлів буде представляти собою відкрите класифікаційне угруповання, яке буде адаптуватися до будівлі медичного закладу і може бути представлене у вигляді (3):

$$V = \{v_1, v_2, \dots, v_n\}, \quad (3)$$

де  $n$  – кількість вузлів, визначених у будівлі медичного закладу.

Наступним етапом є ініціалізація ребер. Для виконання даного етапу необхідно створити множину ребер  $E$ , яка буде об'єднувати наступні підмножини:

- підмножину  $F$  – коридори, що є сполученням між вузлами;
- підмножину  $I$  – сходові переходи, що з'єднують поверхи;
- підмножину  $W$  – шахти ліфтів, що є вертикальними сполученнями між вузлами;
- підмножину  $Z$  – довжина пандусів.

Визначаємо, що  $F \subseteq E, I \subseteq E, W \subseteq E, Z \subseteq E$ . Елементи вказаних підмножин є унікальними та не входять у інші підмножини. Таким чином, для них справедливо  $F \cap I = \emptyset, F \cap W = \emptyset, F \cap Z = \emptyset, I \cap W = \emptyset, I \cap Z = \emptyset, W \cap Z = \emptyset$ .

Отже,

$$E = F \cup I \cup W \cup Z. \quad (4)$$

Множина ребр буде представляти собою відкрите класифікаційне угруповання, яке буде адаптуватися до будівлі медичного закладу і може бути представлене у вигляді (5). Враховуємо, що кожне ребро  $e_{ij}$  з'єднує два вузли  $v_i$  та  $v_j$ ,  $i = \overline{1..n}, j = \overline{2..n}$ , де  $n$  – кількість вузлів будівлі медичного закладу, отримуємо:

$$E = \{ \forall V, i = \overline{1, \dots, n}, j = \overline{2, \dots, n} \}. \quad (5)$$

Кожне ребро  $e_{ij}$  матиме початкову вагу  $\omega_{ij}$  рівну його фізичній відстані  $d_{ij}$ .

$$\omega_{ij} = d_{ij}. \quad (6)$$

**Наступним етапом** є визначення ваг профілю користувача  $p_{ij}^k$  та стану середовища  $s_{ij}$ . З метою врахування широкого спектру факторів, які впливають на вибір оптимального маршруту для користувачів із різними потребами, використовується продукційна модель знань.

Продукційна модель знань є ефективним засобом для формалізації правил і умов у системах прийняття рішень. Вона ґрунтується на наборі правил  $R$ , які формулюються у форматі: "Якщо (умова), то (дія)". У контексті систем маршрутизації ці правила регулюють зміну ваг маршрутів залежно від профілю користувача та поточного стану середовища. Структура моделі включає кілька ключових компонентів. По-перше, це факти, що містять інформацію про поточний стан середовища, наприклад, доступність ліфтів, пандусів, ширину дверей, а також про профіль користувача. По-друге, набір правил забезпечує логічні умови, які визначають вибір оптимальних маршрутів. Нарешті, модуль прийняття рішень слугує механізмом виконання цих правил, модифікуючи вагу маршрутів відповідно до встановлених умов, що гарантує адаптивність і точність навігаційної системи.

Формалізація правил  $R_c$  у продукційній моделі забезпечує можливість динамічного вибору маршрутів із урахуванням встановлених критеріїв. Приклад правил:

*Доступність інфраструктури*

Правило  $R_1$ : Якщо маршрут включає сходи, а користувач пересувається на візку, то встановити вагу ребра як "нескінченність".

Правило  $R_2$ : Якщо ліфт недоступний (несправність, ремонт), то виключити всі маршрути, які залежать від його використання.

*Індивідуальні пріоритети користувача, що визначені його профілем*

Правило  $R_3$ : Якщо користувач має обмеження у пересуванні та пріоритетом є мінімізація фізичного навантаження, то додати до ваги ребра коефіцієнт залежно від підйому або спуску.

Правило  $R_4$ : Якщо користувач із вадами зору, а маршрут має погане освітлення, то збільшити вагу ребра на визначений коефіцієнт.

*Динамічні умови середовища*



Правило  $R_5$ : Якщо сегмент маршруту тимчасово недоступний через ремонт, то виключити відповідне ребро з графа.

Правило  $R_6$ : Якщо в будівлі є велика кількість користувачів, які використовують ліфт, то збільшити вагу ребр, які включають ліфт, через можливі затримки.

Ці правила забезпечують можливість гнучкої адаптації маршруту до різноманітних умов і профілів користувачів.

Вага ребр для конкретного користувача  $p_{ij}^k$  визначається профілем користувачів. і характеристиками краю –  $v_i$  та  $v_j$ . Вона враховує конкретні потреби та обмеження користувача, такі як тип мобільності, сенсорні здібності та тимчасові чи постійні фізичні обмеження

$$p_{ij}^k = \sum_{c=1}^m f_c(U, v_i, v_j), \quad (7)$$

де:

-  $U$ : профіль користувача, який містить таку інформацію, як тип мобільності, переваги та обмеження.

-  $f_c$ : функція, яка застосовує правило  $R_c$  для налаштування ваги ребра на основі профілю користувача  $U$  та враховуючи тип країв ребр  $v_i$  та  $v_j$  зокрема приналежність до підмножин  $C, L, H$ .

Значення, пов'язане зі станом середовища  $s_{ij}$  враховує умови навколишнього середовища в реальному часі, які можуть вплинути на доступність ребра  $e_{ij}$  за рахунок стану вузлів  $v_i$  та  $v_j$ , а також самого ребра  $e_{ij}$ . Це включає такі фактори, як несправність ліфта, тимчасові перешкоди та велика кількість пішоходів

$$s_{ij} = \sum_{c=1}^n g_c(D, e_{ij}, v_i, v_j), \quad (8)$$

де:

-  $D$ : Дані про навколишнє середовище, які містять інформацію в реальному часі про інфраструктуру будівлі (наприклад, стан ліфта, умови освітлення).

-  $g_c$ : функція, яка застосовує правило  $R_c$  для налаштування ваги краю на основі даних навколишнього середовища  $D$  та характеристик краю ребра  $e_{ij}$  (стан  $v_i$  та  $v_j$ ) та стану самого ребра.

Вага ребр графа визначається комбінацією фізичної відстані, вагою профіля користувача та вагою навколишнього середовища. Формалізуючи ці компоненти за допомогою математичного представлення, адаптивний алгоритм Дейкстри може динамічно регулювати ваги ребр на основі конкретних потреб користувача та стану середовища. Такий підхід гарантує, що навігаційна система забезпечує оптимальні, доступні та безпечні маршрути для всіх користувачів.

Таким чином кроки модифікованого алгоритму Дейкстри будуть наступні:

1. *Ініціалізація:*

- Для кожного вузла  $v_i \in V$ , встановлюємо орієнтовну відстань  $dist\ v_i = \infty$ .

Встановлюємо орієнтовну відстань вихідного вузла  $v_i^\alpha\ dist\ v_i^\alpha = \infty$ .

- Ініціалізація черги пріоритетів  $Q$  з усіма вузлами в  $V$ , пріоритетними за їх попередніми відстанями.

2. *Основний цикл:*

- Поки  $Q$  не пуста:

- Витягуємо вузол  $v_j$  із найменшою орієнтовною відстанню від  $Q$ .
- Для кожного сусіда  $v_i$  з  $v_j$ :
- Обчислюємо динамічну вагу  $\omega_{ij} = d_{ij} + p_{ij}^k + s_{ij}$ .
- Якщо  $dist(v_i) > dist(v_j) + \omega_{ij}$ , оновлення  $dist(v_i) = dist(v_j) + \omega_{ij}$ .
- Оновлюємо чергу пріоритетів  $Q$  новою орієнтовною відстанню для  $v_i$ .

### 3. Припинення:

- Алгоритм завершується, коли всі вузли оброблені та визначено найкоротший шлях від вихідного вузла до вузлів призначення.

Часова складність модифікованого алгоритму Дейкстри залишається  $O((V+E)\log V)$ , де  $V$  – кількість вершин у графі, а  $E$  – кількість ребер. Це пояснюється тим, що динамічне коригування ваги не змінює фундаментальної структури алгоритму; він лише змінює ваги ребер під час виконання.

Однак складність простору може дещо збільшитися через необхідність зберігати додаткову інформацію про профілі користувачів і умови середовища. Зазвичай цим можна керувати, оскільки кількість профілів користувачів і факторів навколишнього середовища відносно невелика порівняно з розміром графа.

Розширення структури алгоритму передбачає інтеграцію продукційної моделі як окремого модуля, що активно взаємодіє з основними компонентами адаптивного алгоритму Дейкстри. Модуль профілю користувача забезпечує передачу інформації про індивідуальні потреби та пріоритети, що дозволяє алгоритму враховувати специфічні вимоги кожного користувача. Модуль графа містить базові дані про структуру будівлі, включаючи ребра та вершини графа, які слугують основою для пошуку маршрутів. Водночас модуль продукцій обробляє набір правил, що дозволяють модифікувати ваги ребр графа залежно від динамічних умов середовища та потреб користувача, забезпечуючи адаптивність і ефективність маршрутизації.

Етапи роботи алгоритму з продукційною моделлю:

Ініціалізація: Збираються факти про середовище (доступність маршрутів, стан інфраструктури) та профіль користувача.

Аналіз правил: Продукційна модель обробляє правила, визначаючи необхідні модифікації ваг ребр.

Модифікація графа: Ваги ребр коригуються відповідно до результатів обробки правил. Недоступні маршрути виключаються з графа.

Виконання алгоритму Дейкстри: На оновленому графі виконується пошук найкоротшого шляху, що враховує умови інклюзивності.

Формування результату: Користувачеві надається оптимальний маршрут із поясненням, які правила були застосовані.

Переваги інтеграції:

Гнучкість: можливість швидко адаптуватися до змін у середовищі або потребах користувача.

Масштабованість: можливість додавання нових правил без потреби змінювати основну структуру алгоритму.

Прозорість: чітке пояснення, чому було обрано той чи інший маршрут, підвищує довіру користувачів.

Інтеграція продукційної моделі знань у адаптивний алгоритм Дейкстри створює ефективну систему маршрутизації, здатну враховувати широкий спектр умов і забезпечувати високий рівень інклюзивності. Це рішення дозволяє покращити якість обслуговування в медичних закладах та забезпечити зручність і доступність для всіх категорій користувачів.

## 5 ОБГОВОРЕННЯ РЕЗУЛЬТАТІВ

Подальший розвиток і вдосконалення системи навігації відкриває значні перспективи для покращення її функціональності, адаптивності та масштабованості. Одним із напрямів є розширення можливостей персоналізації маршрутів, що враховуватимуть рівень фізичної активності, комфортність пересування, а також індивідуальні обмеження чи стан здоров'я користувача. Інтеграція рекомендацій у реальному часі дозволить оперативно коригувати маршрути залежно від змін середовища. Підтримка багатомовності зробить систему доступною для ширшої аудиторії, а впровадження технологій доповненої реальності сприятиме більш інтуїтивній візуалізації маршрутів. Крім того, оптимізація алгоритмів дозволить підвищити швидкість роботи системи, особливо у великих будівлях, зокрема за допомогою машинного навчання для прогнозування завантаженості маршрутів.

Перспективи застосування алгоритму не обмежуються окремими будівлями. У торговельних центрах модифікований алгоритм може допомогти відвідувачам знайти оптимальні маршрути між магазинами чи виходами, враховуючи потоки людей. У навчальних закладах алгоритм полегшить пересування студентів і персоналу, враховуючи обмеження доступності аудиторій. У транспортних вузлах, таких як аеропорти чи вокзали, сприятиме швидкому й зручному пересуванню пасажирів із урахуванням доступних зон і ліфтів. Розширення запропонованого алгоритму пошуку оптимального маршруту на зовнішнє середовище відкриє можливості створення інклюзивних міських маршрутів, враховуючи якість тротуарів, пандусів чи транспортної інфраструктури. Крім того, алгоритм може бути застосований у системах евакуації для оптимального виведення людей із будівель під час надзвичайних ситуацій або в туристичних маршрутах для забезпечення доступності визначних пам'яток.

Подальші теоретичні дослідження спрямовані на адаптацію алгоритму для специфічних задач, таких як енергоефективність або підвищення безпеки. Поєднання алгоритму Дейкстри з іншими методами оптимізації, наприклад, генетичними алгоритмами чи методами рою частинок, дозволить досягти ще більш точних результатів. Використання комп'ютерних симуляцій для тестування алгоритму у віртуальних середовищах також сприятиме його вдосконаленню.

Масштабування системи передбачає її адаптацію до великих будівельних комплексів і кампусів із багатьма з'єднаними зонами. Розробка універсального API сприятиме інтеграції алгоритму в різні комерційні та державні навігаційні платформи. Це не тільки розширить сферу застосування системи, а й забезпечить підвищення інклюзивності в широкому масштабі. В результаті запропонований метод може стати основою для створення універсальних рішень, які задовольнятимуть потреби користувачів із різними вимогами в будь-якому середовищі.

## 6 ВИСНОВКИ

У результаті проведеного дослідження було досягнуто поставлених цілей, спрямованих на розробку алгоритму пошуку оптимального маршруту у закритих приміщеннях із використанням модифікованого алгоритму Дейкстри та продукційної моделі знань.

У процесі розробки адаптивного алгоритму Дейкстри була створена його модифікована версія, яка враховує специфічні потреби користувачів із обмеженими можливостями. Основною перевагою цієї версії є адаптація ваг ребр графа будівлі залежно від потреб користувача та динамічних умов середовища, таких як доступність ліфтів, пандусів чи наявність перешкод. Алгоритм здатний виключати недоступні

маршрути та пропонувати альтернативні шляхи, забезпечуючи при цьому гнучкість навіть в умовах частих змін середовища, наприклад, під час ремонтів чи виникнення несправностей.

Продукційна модель знань була інтегрована в процес маршрутизації, що дозволило формалізувати правила для вибору маршрутів, забезпечивши об'єктивність і прозорість прийняття рішень. Ця модель враховує широкий спектр факторів, включаючи профіль користувача та характеристики середовища. Завдяки її використанню система стала масштабованою, що дає можливість легко додавати нові правила та розширювати функціонал для покращення адаптації до різних сценаріїв.

Теоретичний внесок цієї роботи полягає у модифікації класичного алгоритму Дейкстри, який може бути застосований у різноманітних задачах маршрутизації. Практичний внесок включає розробку основи для навігаційних систем, що відповідають принципам універсального дизайну. Такі системи можуть бути ефективно використані в медичних закладах, торговельних центрах, навчальних установах та інших типах будівель.

Таким чином, запропонована система пошуку маршруту успішно досягла поставлених цілей. Вона забезпечує адаптивність, інклюзивність і ефективність у навігації всередині будівель, створюючи умови для комфортного пересування користувачів із різними потребами. Результати цього дослідження можуть стати базою для подальших робіт у галузі інклюзивного дизайну та автоматизованих систем навігації.

## 7 ЕТИЧНІ ДЕКЛАРАЦІЇ

Автор не має будь-яких фінансових чи нефінансових інтересів щодо матеріалів, представлених у цій статті, які слід розкривати.

## Література

1. Задача пошуку найкоротшого шляху: порівняльний аналіз основних алгоритмів / В. Бабич та ін. *Information Technology: Computer Science, Software Engineering and Cyber Security*. 2023. №2. С. 99–106. URL: <https://doi.org/10.32782/it/2023-2-12> (дата звернення: 10.02.2025).
2. Comparative analysis of pathfinding algorithms A\*, Dijkstra, and BFS on Maze Runner game / S. D. Handy Permana et al. *IJISTECH (International Journal Of Information System & Technology)*. 2018. Vol.1, no.2. P.1. URL: <https://doi.org/10.30645/ijistech.v1i2.7> (date of access: 10.02.2025).
3. Jabbar L., Abass E., Hasan S. A modification of shortest path algorithm according to adjustable weights based on Dijkstra algorithm. *Engineering and Technology Journal*. 2022. Vol.41, no.2. P.1–16. URL: <https://doi.org/10.30684/etj.2022.136107.1296> (date of access: 10.02.2025).
4. Jamaludin, Setiyadi D., Retnoningsih E. Dijkstra algorithm for finding the shortest route to company addresses in industrial areas. *Jurnal Mantik*. 2023. Vol.7, no.3. P.2653–2664.
5. Dijkstra E. W. A note on two problems in connexion with graphs. *Numerische mathematik*. 1959. Vol. 1, no. 1. P. 269–271. URL: <https://doi.org/10.1007/bf01386390> (date of access: 10.02.2025).
6. Alam M. A., Faruq M. O. Finding shortest path for road network using dijkstra's algorithm. *Bangladesh journal of multidisciplinary scientific research*. 2019. Vol. 1, no. 2. P. 41–45. URL: <https://doi.org/10.46281/bjmsr.v1i2.366> (date of access: 11.02.2025).
7. Gunawan W., . S., Sudrajat B. Implementation of Dijkstra's algorithm in the shortest route. *Scholars bulletin*. 2019. Vol. 05, no. 12. P. 681–689. URL: <https://doi.org/10.36348/sb.2019.v05i12.001> (date of access: 11.02.2025).
8. Samah K. A. F. A., Hussin B., Basari A. S. H. Modification of Dijkstra's algorithm for safest and shortest path during emergency evacuation. *Applied mathematical sciences*. 2015. Vol. 9. P. 1531–1541. URL: <https://doi.org/10.12988/ams.2015.49681> (date of access: 11.02.2025).

9. Tamatjita E. N., Mahastama A. W. Shortest Path with Dynamic Weight Implementation using Dijkstra's Algorithm. *ComTech: computer, mathematics and engineering applications*. 2016. Vol. 7, no. 3. P. 161. URL: <https://doi.org/10.21512/comtech.v7i3.2534> (date of access: 11.02.2025).
10. Zudikhin Y., Malaksiano M. Navigation system based on Dijkstra's algorithm with spatial context integration. *The scientific heritage*. 2024. No. 143. P. 93–96. URL: <https://doi.org/10.5281/zenodo.13378618> (date of access: 11.02.2025)

## References

1. Babych, V., Kostenko, A., Plesha, V., Plesha, M., & Khmilyarchuk, L. (2023). The problem of finding the shortest path: a comparative analysis of the main algorithms *Information Technology: Computer Science, Software Engineering and Cyber Security*, (2), 99–106. <https://doi.org/10.32782/it/2023-2-12>
2. Handy Permana, S. D., Yogha Bintoro, K. B., Arifitama, B., & Syahputra, A. (2018). Comparative Analysis of Pathfinding Algorithms A \*, Dijkstra, and BFS on Maze Runner Game. *IJISTECH (International Journal of Information System & Technology)*, 1(2), 1. <https://doi.org/10.30645/ijistech.v1i2.7>
3. Jabbar, L., Abass, E., & Hasan, S. (2022). A modification of shortest path algorithm according to adjustable weights based on dijkstra algorithm. *Engineering and Technology Journal*, 41(2), 1–16. <https://doi.org/10.30684/etj.2022.136107.1296>
4. Jamaludin, Setiyadi, D., & Retnoningsih, E. (2023). Djikstra algorithm for finding the shortest route to company addresses in industrial areas. *Jurnal Mantik*, 7(3), 2653–2664.
5. Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1), 269–271. <https://doi.org/10.1007/bf01386390>
6. Alam, M. A., & Faruq, M. O. (2019). Finding shortest path for road network using dijkstra's algorithm. *Bangladesh Journal of Multidisciplinary Scientific Research*, 1(2), 41–45. <https://doi.org/10.46281/bjmsr.v1i2.366>
7. Gunawan, W., S., & Sudrajat, B. (2019). Implementation of Dijkstra's algorithm in the shortest route. *Scholars Bulletin*, 05(12), 681–689. <https://doi.org/10.36348/sb.2019.v05i12.001>
8. Samah, K. A. F. A., Hussin, B., & Basari, A. S. H. (2015). Modification of Dijkstra's algorithm for safest and shortest path during emergency evacuation. *Applied Mathematical Sciences*, 9, 1531–1541. <https://doi.org/10.12988/ams.2015.49681>
9. Tamatjita, E. N., & Mahastama, A. W. (2016). Shortest Path with Dynamic Weight Implementation using Dijkstra's Algorithm. *ComTech: Computer, Mathematics and Engineering Applications*, 7(3), 161. <https://doi.org/10.21512/comtech.v7i3.2534>
10. Zudikhin Y., & Malaksiano M. (2024). Navigation system based n dijkstra's algorithm with spatial context integration. *The Scientific Heritage*, 143, 93–96. <https://doi.org/10.5281/zenodo.13378618>

### Лисенко Максим Сергійович

Державний університет «Житомирська політехніка»,  
асистент,  
вул. Чуднівська, 103, Житомир, Україна 10005  
kkn\_lms@ztu.edu.ua  
ORCID: 0000-0001-8252-717X

### Для посилань:

Лисенко М. С. Розробка алгоритму пошуку оптимального маршруту у закритих приміщеннях. *Механіка та математичні методи*, 2025. Т. VII. № 1. С. 125–137.

### For references:

M. Lysenko. (2025). Development of an algorithm for searching for the optimal route in indoor premises. *Mechanics and Mathematical Methods*. VII (1). 125–137.