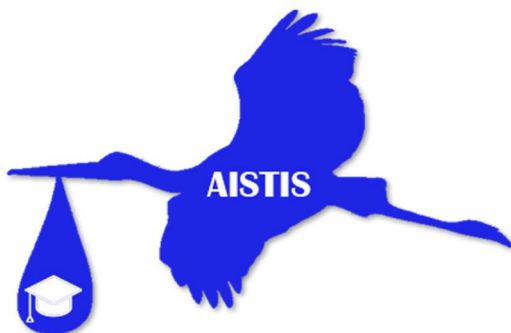


**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА**



## **«ПРИКЛАДНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ В ІНФОРМАЦІЙНОМУ СУСПІЛЬСТВІ»**

**Збірник тез  
IV Міжнародної науково-практичної конференції  
30 вересня 2020 року**



**м. Київ**

УДК 004:378(082)

**Адреса редакційної колегії:**  
Факультет інформаційних технологій  
Київського національного університету імені Тараса Шевченка,  
вул. Богдана Гаврилишина, 24, Київ, 04116, Україна

Тези друкуються мовою оригіналу,  
відображають позицію автора,  
який несе відповідальність за зміст

**Прикладні** системи та технології в інформаційному суспільстві: зб. тез доповідей і наук. повідомл. учасників IV Міжнародної науково-практичної конференції (Київ, 30 вересня 2020 р.) / за заг. ред. В.Л. Плескач, В.Л. Міронова. – К.: Київський нац. ун-т імені Тараса Шевченка, 2020. – с.

Цей збірник містить тези учасників IV міжнародної науково-практичної конференції «Прикладні системи та технології в інформаційному суспільстві», яка проводилась 30 вересня 2020 року на базі факультету інформаційних технологій Київського національного університету імені Тараса Шевченка.

Робочі мови конференцій: українська, англійська, російська.

Основною метою конференції є вирішення актуальних проблем розвитку прикладних інформаційних систем у цифровій економіці, захисту даних зазначених систем, а також перспективних технологій в інфокомунікаційних системах, сучасній освіті та правових аспектів в інформаційно-комунікаційних технологіях.

Подані матеріали містять методологічні та методичні підходи, які можуть заслуговувати на увагу широкого кола фахівців.

**УДК 004:378(082)**

© Київський національний університет імені Тараса Шевченка  
Факультет інформаційних технологій, 2020

<b>Малишев О. В., Калмиков В.Г.</b> Концептуальна модель планування діяльності на основі спроможностей.....	112
<b>Міронова В.Л., Гарко І.І., Богданець Є.І.</b> Методика побудови інтелектуальних систем цифрового суспільства.....	115
<b>Наконечний В.С., Толюпа С.В., Петренко А.І.</b> Особливості побудови системи кібербезпеки в світі інтернет речей.....	119
<b>Наконечний В.С., Кузьменко О.О.</b> Клавіатурний почерк – механізм біометричної ідентифікації користувача в інформаційній системі .....	122
<b>Ніколаєв М.О.</b> Дослідження використання машинного навчання та алгоритмічного кодування при прогнозуванні цін та оцінці ризиків майбутніх періодів на світових фінансових ринках .....	126
<b>Новіцька Т. В.</b> Виклики та освітні технології у контексті підготовки фахівців математичних (фізичних) спеціальностей в умовах дистанційного навчання .....	130
<b>Онуфрик О.І.</b> Modern algorithms in natural language processing .....	132
<b>Орехов О.А., Орехова Н.А.</b> Застосування ансамблевих методів для прогнозування епідемічної ситуації (на базі класичної SIR-моделі).....	134
<b>Павелко Т. М.</b> Застосування мікроконтролерів у зв'язці з інформаційною системою електронного документообігу загальноосвітньої школи .....	138
<b>Панасюк О. І.</b> Програмні рішення для взаємодії між сервісами прикладних інформаційних систем.....	140
<b>Передерій Н.О.</b> Е-комерція в умовах пандемії .....	143
<b>Підлужний Д.О.</b> Використання сучасних алгоритмів розпізнавання обличчя під час пандемії COVID-19 .....	147
<b>Пікуляк М.В., Корнута В.Р.</b> Вдосконалення евристичного алгоритму пошуку оптимального маршруту до паркувальної зони.....	149
<b>Пирог М.В., Дуля Д.В.</b> Порівняння платформ .NET CORE та NODE.JS як технологій веб-розробки.....	154
<b>Пирог М.В., Борсук О.С.</b> Фреймворк Angularjs та бібліотека React.Js як інструменти розробки динамічних веб-застосунків .....	158
<b>Плескач В.Л., Плескач М.В.</b> Індекс розвитку електронного урядування та електронних публічних послуг в Україні: 2020 .....	160

Пікуляк М.В., Корнута В.Р.

ДВНЗ «Прикарпатський національний університет імені В.Стефаника»

[mykolapikulyak@gmail.com](mailto:mykolapikulyak@gmail.com); [atunrok888@gmail.com](mailto:atunrok888@gmail.com)

## ВДОСКОНАЛЕННЯ ЕВРИСТИЧНОГО АЛГОРИТМУ ПОШУКУ ОПТИМАЛЬНОГО МАРШРУТУ ДО ПАРКУВАЛЬНОЇ ЗОНИ

*Abstract. The analysis of known algorithms for determining the route is performed and the algorithm A\* is improved, which allows to increase the speed of processing program results and expand the scope of its application.*

**Keywords:** *heuristic algorithm, parking zone, weight function, route determination.*

**Вступ.** Сьогодні на ринку програмних продуктів представлено чималу кількість сервісів, що дають можливість прокладати маршрут до потрібної точки та мають необхідну базу паркувальних зон. Проте багато з них мають недоліки, пов'язані із ліцензійними обмеженнями, недостатньою повнотою вихідних просторових даних та неможливістю застосування власних алгоритмів.

Крім цього виникає необхідність у розробці додаткових функцій подібних сервісів, пов'язаних із проведенням деталізованого аналізу отриманих результатів пошуку, можливістю застосування на місцевостях із складним ландшафтом, підтримкою актуальності даних. Тому розробка подібних програмних додатків є актуальною задачею, яка вимагає подальшого їх дослідження та розробки нових і вдосконалення відомих алгоритмів обробки.

**Аналіз відомих алгоритмів.** Усі відомі алгоритми для визначення маршруту, по якому можна потрапити з однієї точки в іншу, можна розбити на дві групи:

- алгоритми, що дозволяють визначити оптимальний шлях;
- алгоритми, що дозволяють знайти субоптимальний шлях.

Алгоритми першої групи потребують детального дослідження простору станів задачі. Найпростішим способом пошуку оптимального шляху є повний перебір всіх можливих маршрутів. Це гарантує знаходження оптимального шляху. Але на практиці частіше за все такий спосіб неможливо використати із-за надмірних накладних витрат, оскільки він потребує зберігання в пам'яті усієї досліджуваної області.

Прикладом алгоритмів пошуку субоптимальних шляхів є евристичні алгоритми, які дозволяють на кожному кроці максимально наблизитись до кінцевої точки. Однак при пошуку одного з найближчих до оптимального шляху слід враховувати, що спочатку важко точно передбачити, який саме варіант буде

обраний. У цьому випадку можна використовувати різні алгоритми при виборі напрямку на кожному кроці. Найбільш раціональним у такій ситуації здається розробка окремих алгоритмів для різних класів задач на базі алгоритмів визначення оптимального шляху із застосуванням евристик, що враховують специфіку даної проблемної області.

Багато алгоритмів мають схожу проблему: отриманий з їх допомогою шлях виглядає не найкоротшим. Для її вирішення потрібно або використовувати оптимізацію шляху, або застосовувати алгоритм, в якому вже використовуються функції, які дозволяють отримати найкоротший шлях в дискретній моделі [1].

Можна узагальнити наступні найбільш ефективні алгоритми на основі яких розробляються похідні:

- алгоритм Дейкстри (використовується для знаходження оптимального маршруту між двома вершинами). Перевагою даного алгоритму є простота реалізації та можливість його застосування в графах з невеликою кількістю вершин. З огляду на структуру схеми міста, програма, що написана з використанням алгоритму Дейкстри буде виконуватися повільно, а розподіл пам'яті не буде раціональним [2].

- алгоритм Флойда (для знаходження оптимального маршруту між усіма парами вершин). Цей алгоритм призначений для вирішення завдання пошуку всіх найкоротших шляхів на графі. Також він розпізнає наявність негативних циклів. Недоліком його застосування є те, що він містить три вкладені цикли, тобто має кубічну складність. Тому при його застосуванні буде використовуватися великий обсяг пам'яті, що є не раціональним рішенням [3].

алгоритм Беллмана-Форда (для знаходження найкоротшого шляху від однієї вершини графа до всіх інших). Його відмінною рисою є можливість застосування до графів з довільними, в тому числі негативними, вагами. Алгоритм використовує повний перебір всіх вершин графа, що призводить до великої втрати часу та потребує значного обсягу пам'яті.

**Імплементация алгоритму  $A^*$  для системи прокладання маршруту до паркувальної зони.**

Алгоритм мобільної системи прокладання маршруту до найближчого паркінгу для автомобілів на базі ОС Android повинен базуватися на удосконаленому алгоритмі  $A^*$  і працювати на вуличній мережі, яка моделюється як граф. Вузлами в графі будуть представлені переходи чи дороги, що не мають наскрізного проїзду (тупики), а ребрами – вуличні сегменти. В удосконаленому алгоритмі необхідно врахувати, що граф спрямований, а це означає, що є обмеження напрямку (наприклад, враховується рух в одну сторону вулиці). Ребро з обома напрямками

означає, що можливе управління на вулиці в обох напрямках, незалежно від структурних умов, які можуть відокремити смуги [4].

Блок-схема вдосконаленого алгоритму A\* представлена на рис.1:

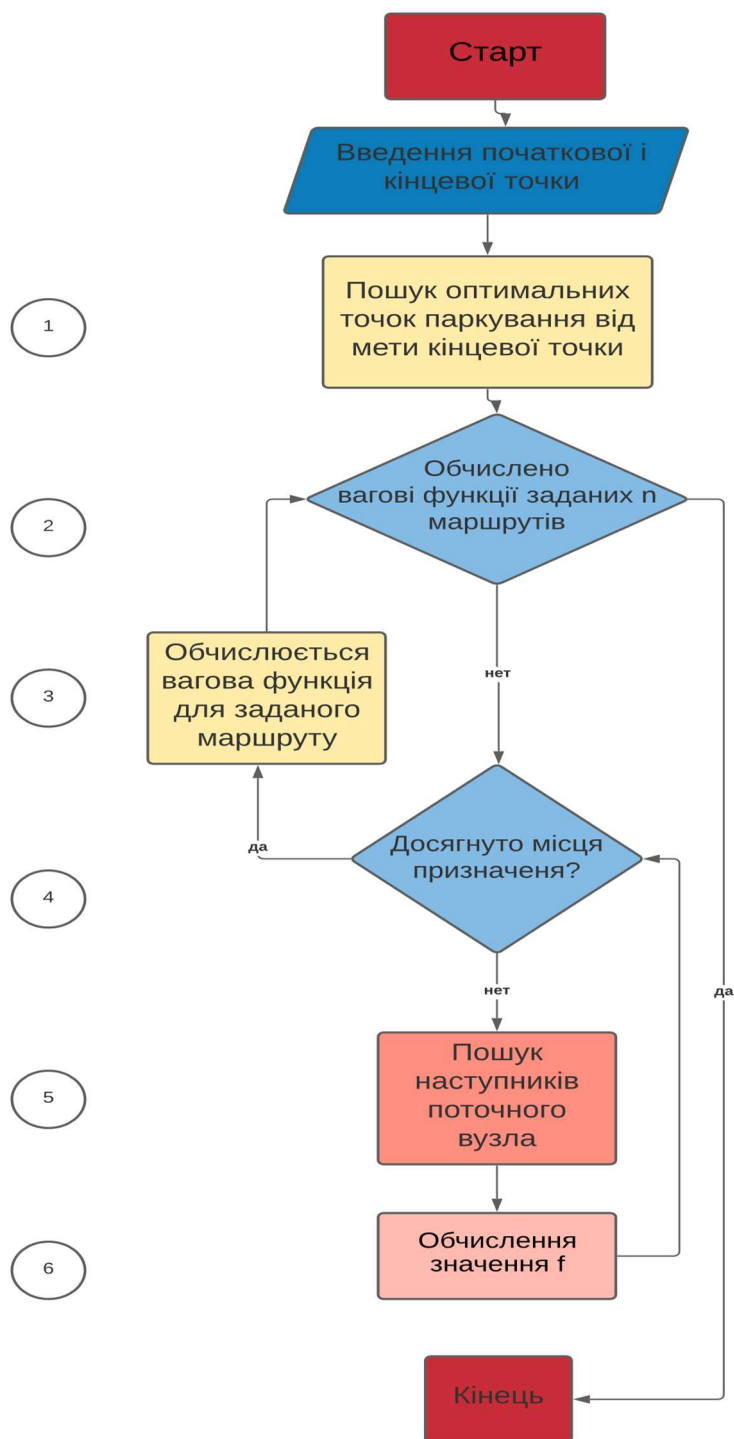


Рис. 1. Блок-схема вдосконаленого алгоритму

Робота алгоритму розпочинається з того, що спочатку відбувається пошук паркувальних точок в заданому оптимальному радіусі для водія. На наступному

кроці ми перевіряємо, чи є закінчений обрахунок вартості маршрутів. Якщо так, то алгоритм закінчується; якщо ні, тоді слідує четвертий крок.

З четвертого по шостий крок знаходиться найкоротший шлях за допомогою функції оцінки:

$$f(x) = g(x) + h(x),$$

де:

$g$  – визначає витрати від початкового вузла до вузла  $x$ ;

$h$  – описує евристичні орієнтовні витрати від  $x$  до кінцевого вузла.

Вузол визначається як поточний, якщо він має найменше значення  $f$ .

На другому кроці виконання алгоритму перевіряється чи поточний вузол є цільовим вузлом. Якщо так, то алгоритм закінчується. Якщо ні, слідує крок 3.

Коли вивчається поточний вузол  $x$ , то відбувається визначення та вивчення всіх його наступників. Для кожного наступника  $y$  обчислюється функція  $f(y)$ .

На п'ятому кроці обчислюються затрати на шлях до вузла  $x$ , а саме  $g(x)$  і затрати від поточного вузла  $x$  до його наступника.

На шостому кроці обчислюється функція:

$$f(y) = g(y) + h(y),$$

де:

$g(y)$  – це затрати від початкового вузла до вузла  $y$ ;

$h(y)$  – оцінює затрати від вузла наступника на цільовий вузол з евристикою.

В кінці  $f(y)$  оцінює спадковий вузол  $y$ .

Після вищевказаного відбувається перехід на четвертий крок, і кроки 4-6 повторюються до досягнення цільового вузла.

Якщо був зроблений перехід до цільового вузла, то відбувається третій крок, який здійснює обрахунок вагової функції всього маршруту. Далі йде перевірка чи було обраховано ваговий коефіцієнт всіх маршрутів. Якщо так, алгоритм закінчується.

Отже, вагова функція складається з часу проїзду від поточного вузла до наступного вузла, ймовірності паркування на ділянці дороги, яка зв'язує поточний і наступний вузол, а також коефіцієнт оптимального віддалення.

Не тільки вагова функція, але і евристика адаптована для алгоритму паркування. В якості евристичної функції береться евклідова відстань від вузла наступника до цільового вузла, розділеного на константу. Константа складається із рекомендованої швидкості руху та не повинна переоцінювати витрати для вулиць у місті.

Порядок кроків алгоритму – це функція оцінки витрат з допомогою евристики і призводить до формування паркувальних точок маршрутизації.

У більшості випадків  $A^*$  обчислює найшвидший або найкоротший шлях, тому функція витрат включає лише час проїзду або на довжину дорожньої лінії. Проте, щоб вирішити проблему паркування автомобіля з найменшою витратою часу на подолання відстані до потрібної точки, потрібно також враховувати вірогідність паркування і час пішого проходження до кінцевої точки маршруту.

**Висновок.** В результаті проведеного дослідження виконано аналіз відомих алгоритмів визначення маршруту та запропоновано вдосконалений евристичний алгоритм пошуку, що дає можливість підвищити швидкість обробки отриманих результатів та розширити застосування додатку у відповідності до змін ландшафту.

Практична реалізація даного алгоритму дозволить побудувати сервіс, який вирішить проблему паркування автомобіля з найменшою витратою часу на подолання відстані до потрібної точки.

#### ЛІТЕРАТУРА

1. Delling, D.; Sanders, P.; Schultes, D.; Wagner, D. (2009). Engineering route planning algorithms. Algorithmics of Large and Complex Networks: Design, Analysis, and Simulation. Springer. с. 117–139. doi:10.1007/978-3-642-02094-0\_7.
2. [Dijkstra E. W. A note on two problems in connexion with graphs](#) (англ.) // [Numer. Math / F. Brezzi](#) – Springer Science+Business Media, 1959. – Vol. 1, Iss. 1. – P. 269–271. — ISSN [0029-599X](#); [0945-3245](#) — doi:10.1007/BF01386390
3. [Томас Кормен](#); [Чарльз Лейзерсон\[en\]](#), [Рональд Рівест](#) (1990). [Вступ до алгоритмів](#) (вид. 1st). MIT Press і McGraw-Hill. ISBN 0-262-03141-8. Секція 26.2, «The Floyd–Warshall algorithm», С. 558–565 та секція 26.4, «A general framework for solving path problems in directed graphs», С. 570–576.
4. Електронний ресурс : <http://www.raywenderlich.com/4946/introduction-to-a-pathfinding>.



Наукове видання

**ПРИКЛАДНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ В  
ІНФОРМАЦІЙНОМУ СУСПІЛЬСТВІ – 2020**

Збірник тез

IV Міжнародної науково-практичної конференції  
30 вересня 2020 року

м. Київ

Відповідальні за випуск:

В.Л. Плєскач,  
В.Л. Міронова,  
І.І.Гарко