

6. Валкевич О. Як штучний інтелект допомагає в сучасних війнах: чи зможе він замінити військових на фронті. 2023. URL: <https://bitl.to/3G9F>
7. ZIP, система розпізнавання та ураження цілей. URL: <https://zir-system.com/>
8. Gandhi R. R-CNN, Fast R-CNN, Faster R-CNN, YOLO – Object Detection Algorithms. Medium. 2018. URL: <https://bitl.to/3G9G>

УДК 004.021

*Бурківський О. С., здобувач вищої освіти,
Ніколюк П. К., д-р фіз.-мат. наук,
професор, професор кафедри
інформаційних технологій*

ОПТИМІЗАЦІЯ МАРШРУТІВ ПОСТАЧАННЯ ВІЙСЬКОВОЇ ТЕХНІКИ ЗА ДОПОМОГОЮ А*-АЛГОРИТМУ

Донецький національний університет імені Василя Стуса, м. Вінниця

Алгоритм А* (A-star) – це метод пошуку шляхів у графах, який знаходить найкоротший маршрут між початковою і цільовою точками, враховуючи як пройдений шлях, так і передбачувану відстань до цілі. Завдяки поєднанню жадібного підходу та методу Дейкстри він дає змогу ефективно будувати оптимальні маршрути з урахуванням різних факторів, як-от довжина шляху, безпека чи витрати. А* широко застосовується в автомобільній навігації та ігрових системах, оскільки враховує перешкоди та забезпечує гнучкість у пошуку рішень для задач маршрутизації.

Порядок відвідування вершин у алгоритмі А* визначається за допомогою евристичної функції, яка розраховується як сума двох компонентів: вартості шляху від початкової точки до поточної вершини (позначається як $g(x)$ і може бути як точною, так і евристичною) та оцінки відстані від поточної вершини до цільової (позначається як $h(x)$ і є евристичною). Сума цих двох значень, позначена як $f(x)$, дає змогу алгоритму ефективно обирати вершини для подальшого аналізу, оптимізуючи як вже пройдений маршрут, так і очікувану відстань до цільової точки.

Функція $h(x)$ повинна бути коректною евристичною оцінкою. Наприклад, у задачах маршрутизації $h(x)$ може відповідати відстані по прямій до цільової точки, оскільки це найменша можлива фізична відстань між двома точками [1].

Алгоритм А* послідовно досліджує всі можливі маршрути від початкової вершини до цільової, шукаючи шлях із мінімальним значенням $f(x) = g(x) + h(x)$. На кожному кроці він вибирає маршрути, які мають найвищу ймовірність досягнення мети, з урахуванням пройденого шляху $g(x)$ і евристичної оцінки до цільової вершини $h(x)$. Для стартової вершини алгоритм відкриває сусідні вузли з мінімальним значенням $f(x)$ і продовжує пошук, поки значення $f(x)$ для цільової вершини не стане найменшим або поки не будуть переглянуті всі можливі шляхи. Чим менше значення $h(x)$, тим вищий пріоритет шляху, що дає змогу ефективно керувати чергою пріоритетів, наприклад, через використання деревоподібної структури [2].

Граф для A^* -алгоритму моделюється для пошуку найкоротшого шляху між двома вузлами. Для оптимізації маршрутів постачання військової техніки за допомогою A^* -алгоритму можна використовувати такий підхід:

1. Визначення вузлів графу. Вузли графу позначають ключові точки логістичної мережі. У контексті постачання військової техніки це можуть бути склади, бази забезпечення, польові штаби, прикордонні пункти або інші стратегічно важливі об'єкти.

2. Визначення ребер графу. Ребра графу моделюють транспортні шляхи між вузлами. Це можуть бути дороги, залізничні маршрути, річкові чи повітряні коридори, які використовуються для перевезення техніки або вантажів.

3. Призначення ваги ребрам. Вага ребер відображає витрати на переміщення між вузлами. Можна враховувати фактори, як-от відстань, час у дорозі, ризикованість маршруту, витрати пального, стан доріг або загрози (наприклад, можливість ворожих атак) [3].

Після цього потрібно визначити початковий і кінцевий вузол, щоб знайти найкоротший шлях між ними. Можемо розглянути використання A^* -алгоритму лінії фронту, зокрема для знаходження найкоротшого шляху і найбезпечнішого маршруту з позиції J до вершини D. Візьмемо за початкову вершину J як стартову, і у якості кінцевої вершини D (рис. 1).

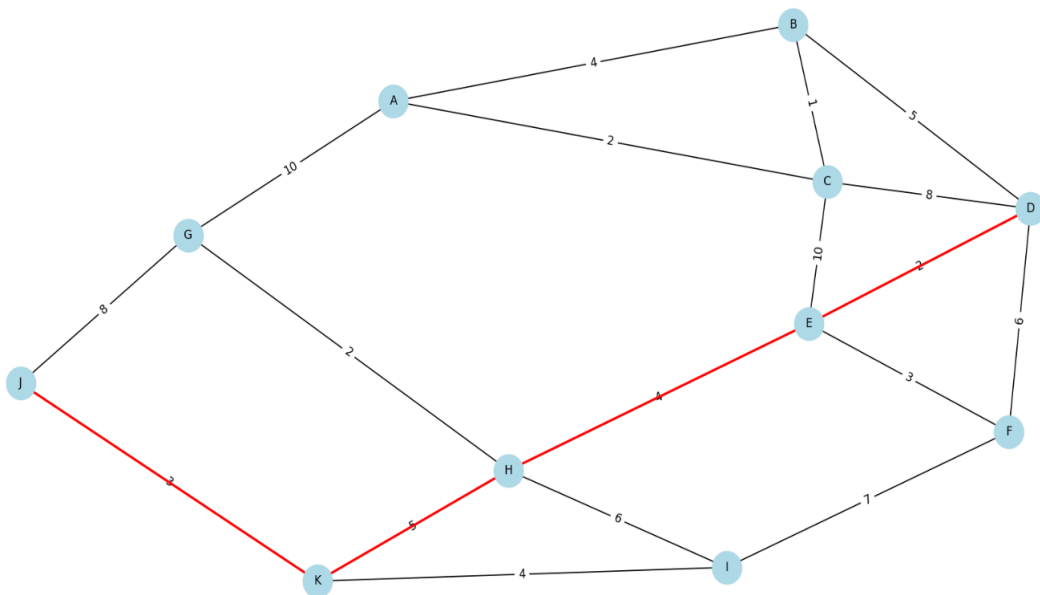


Рисунок 1 – A^* -алгоритм міського трафіка м. Вінниці

Потім розраховуємо в програмі Thony алгоритм постачання військової техніки на лінію фронту, а саме найкоротший шлях від початкової вершини J до кінцевої вершини D (рис. 2). Під час роботи програми враховуються всі визначені параметри, як вага ребер графу, евристична оцінка $h(x)$, а також можливі перешкоди на маршруті. Наприклад, у реальних умовах війни цей алгоритм дає змогу врахувати заблоковані ділянки доріг, загрозу обстрілів або інші небезпеки. Завдяки гнучкості A^* можливо динамічно переналаштовувати маршрут у відповідь на зміну умов, наприклад, у разі перекриття шляху або появи більш безпечного маршруту.

Програма може також інтегрувати додаткові модулі для аналізу ризиків, враховуючи інформацію з дронів, супутників або інших джерел розвідки. Наприклад, врахування ризику обстрілу на певних ділянках може бути реалізовано шляхом збільшення ваги відповідних ребер графу. Такі можливості роблять алгоритм A^* надзвичайно ефективним у військових та інших критичних застосуваннях.

```
def a_star(graph, start, goal, heuristic):
    open_set = PriorityQueue()
    open_set.put((0, start))
    came_from = {}
    g_score = {node: float('inf') for node in graph.nodes}
    g_score[start] = 0
    f_score = {node: float('inf') for node in graph.nodes}
    f_score[start] = heuristic[start]

    while not open_set.empty():
        current = open_set.get()[1]

        if current == goal:
            path = []
            while current in came_from:
                path.append(current)
                current = came_from[current]
            path.append(start)
            path.reverse()
            return path

        for neighbor in graph.neighbors(current):
            tentative_g_score = g_score[current] + graph[current][neighbor]['weight']
            if tentative_g_score < g_score[neighbor]:
                came_from[neighbor] = current
                g_score[neighbor] = tentative_g_score
                f_score[neighbor] = g_score[neighbor] + heuristic[neighbor]
                open_set.put((f_score[neighbor], neighbor))

    return None
```

Рисунок 2 – Код A^* -алгоритму

Результат роботи програми виглядає так [J, K, H, E, D].

Підсумовуючи, можна сказати, що A^* -алгоритм є корисним інструментом для оптимізації маршрутів постачання військової техніки. Цей алгоритм дає змогу визначити найефективніший шлях між стратегічно важливими точками, враховуючи різні фактори, що впливають на логістику, як-от відстань, час у дорозі, витрати ресурсів та рівень ризику. Використання A^* -алгоритму допомагає мінімізувати час доставки, зменшити витрати на транспортування і підвищити безпеку та ефективність постачання в умовах військових операцій.

Список використаних джерел

1. Hart P., Nilsson N., Raphael B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*. Vol. 4, iss. 2. July 1968. P. 100–107. URL: <https://ieeexplore.ieee.org/document/4082128> (дата звернення: 23.11.2024).
2. Zeng W. Church R. Finding Shortest Paths on Real Road Networks: The Case for A^* . *International Journal of Geographical Information Science*. 2009. URL: <https://www.tandfonline.com/doi/abs/10.1080/13658810903270559> (дата звернення: 23.11.2024).

3. Dechter R., Pearl J. Generalized Best-First Search Strategies and the Optimality of A*. Journal of the ACM. URL: <https://dl.acm.org/doi/10.1145/214418.214421> (дата звернення: 23.11.2024).

УДК 004.67

*Дорофєєв Є. О., здобувач вищої освіти,
Січко Т. В., канд. техн. наук, доцент,
доцент кафедри інформаційних технологій*

РОЛЬ АНАЛІТИКИ ВЕЛИКИХ ДАНИХ (BIG DATA) У СУЧАСНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ

Донецький національний університет імені Василя Стуса, м. Вінниця

У сучасному світі дані стають новою «валютою». Розвиток цифрових технологій, Інтернету речей (IoT) і глобальної взаємодії з інформаційними системами призводить до стрімкого зростання обсягів інформації. Аналіз великих даних (Big Data) дає змогу трансформувати структуровані масиви даних у цінну інформацію, яка використовується для ухвалення рішень у бізнесі, науці, державному управлінні та інших сферах. Впровадження Big Data в інформаційні системи стало ключовим фактором їх ефективності.

Актуальність теми зумовлена необхідністю обробки та аналізу великих обсягів даних [1]. Великі дані стали основою для впровадження інновацій у різних галузях – від персоналізованої медицини до автоматизації бізнес-процесів. Проте ефективне використання Big Data вимагає вирішення низки викликів, зокрема забезпечення продуктивності систем, захисту конфіденційності даних та інтеграції наявними інформаційними системами.

Постановка завдання:

- проаналізувати концептуальні основи Big Data та їх роль у сучасних інформаційних системах;
- дослідити підходи до інтеграції аналітики великих даних у бізнес-процеси;
- розглянути переваги та виклики використання Big Data.

Аналітика великих даних (Big Data Analytics) є ключовим компонентом сучасних інформаційних систем, який допомагає працювати з великими обсягами різномірної інформації. Основні технології, які використовуються в аналітиці, включають платформи для обробки даних у розподілених середовищах (Hadoop, Apache Spark), інструменти візуалізації (Tableau, Power BI) і алгоритми машинного навчання, що дають змогу аналізувати як структуровані, так і неструктуровані дані [2].

Одним із найважливіших напрямів використання аналітики великих даних є оптимізація бізнес-процесів. Завдяки аналізу даних з різних джерел (логістичних систем, клієнтських баз, соціальних мереж) компанії можуть виявляти неефективності у своїй роботі та швидко реагувати на зміни попиту.

Наприклад, Walmart використовує Big Data для аналізу сезонного попиту, що допомагає значно скоротити витрати на логістику та зберігання товарів [3].