

*Поліщук В. С.,
студенти 1 курсу спеціальності 122
«Комп'ютерні науки»
Ніколюк П. К., професор, доктор
фізико-математичних наук.*

А-СТАР АЛГОРИТМ ТА ЙОГО ВИКОРИСТАННЯ ДЛЯ ОПТИМІЗАЦІЇ МІСЬКОГО ТРАФІКУ

Донецький національний університет імені Василя Стуса

A^* -алгоритм (А-стар) – це тип алгоритму пошуку на графі, який обчислює маршрут від початкової точки до цільової точки і обчислює найкоротший маршрут. Алгоритми, які також використовуються в загальних автомобільних навігаційних системах та іграх, можуть відображати маршрути з урахуванням перешкод, щоб можна було перевірити безпеку.

Порядок, у якому відвідуються вершини, визначається евристичною функцією («відстань + вартість») (часто позначається $f(x)$). Ця функція є сумою двох інших функцій: функція вартості для досягнення розглянутої вершини (x) від початкової вершини (часто позначається $g(x)$ і може бути або не бути евристичним) і евристична оцінка відстані від розглянутої вершини до кінцевої вершини (позначається як $h(x)$).

Функція $h(x)$ має бути прийнятною евристичною оцінкою. Наприклад, для завдання маршрутизації $h(x)$ може представляти відстань по прямій до цілі, оскільки це найменша фізично можлива відстань між двома точками [1].

Алгоритм вперше був описаний Пітером Хартом у 1968 році разом з Нільсом Нельсоном і Бертрамом Рафаелем. Це, по суті, розширення алгоритму Дейкстри, створеного у 1959 році. Він досяг більшої продуктивності (за часом) за рахунок використання евристики. Він згадується в їхніх творах як «Алгоритм А». Але оскільки він розраховує найкращий маршрут евристично, то називається A^* -алгоритмом [2].

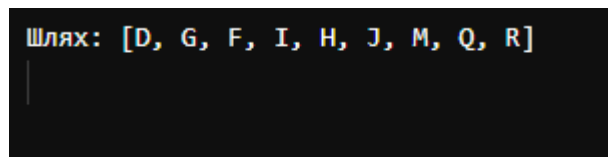
A^* -алгоритм проходить по всіх шляхах, починаючи з початкової вершини, поки не буде знайдено мінімальне значення. Як і всі оптимізаційні алгоритми, A^* -алгоритм спочатку переглядає маршрути, які «найбільш імовірно» ведуть до мети. Цей алгоритм (який є оптимальним алгоритмом пошуку за певних умов) має особливість у тому, що при виборі вершин він враховує весь шлях до них (компонента $g(x)$ – це вага маршруту від стартової вершини до поточної). Для початкового вузла алгоритм обирає всі сусідні вузли з мінімальним значенням $f(x)$ перед тим, як їх відкрити. На кожному кроці алгоритм використовує "незліченні приватні рішення" для кожної ще не відкритої вершини графа, що дозволяє переглядати багато шляхів від початкової точки до цих вершин. Пріоритет кожного шляху визначається значенням $f(x) = g(x) + h(x)$.

Потім розраховуємо в програмі Java алгоритм міського трафіку, а саме найкоротший шлях від початкової вершини D і до кінцевої вершини R (рис.2).

```
Q.adjacencies = new Edge[]{
    new Edge(M, 35),
    new Edge(R, 37),
};
R.adjacencies = new Edge[]{
    new Edge(N, 75),
    new Edge(Q, 37),
    new Edge(S, 43),
};
S.adjacencies = new Edge[]{
    new Edge(R, 43),
    new Edge(O, 40),
};
};
AstarSearch(D, R);
List<Node> path = printPath(R);
```

Рис.2. Код A*-алгоритму

Результат роботи програми виглядає наступним чином (рис.3).



```
Шлях: [D, G, F, I, H, J, M, Q, R]
```

Рис. 3. Результат роботи A*-алгоритму

Підсумовуючи, можна сказати, що A*-алгоритм є корисним інструментом для оптимізації міського трафіку. Цей інструмент можна використовувати для визначення найкоротшого маршруту між двома точками на карті міста, беручи до уваги різні фактори, які впливають на транспорт. Такий алгоритм може допомогти скоротити час, проведений водіями в дорозі, мінімізувати затори та підвищити загальну ефективність транспорту.

Список літератури:

1. wikipedia url: <http://surl.li/gilmw> (дата звернення 09.04.2023)
2. matros url: <http://surl.li/gimco> (дата звернення 09.04.2023)
3. fet.kip.ua url: http://tef.kpi.ua/files/pdf/optimization_1268915918.pdf (дата звернення 09.04.2023)
4. nina.az url: <http://surl.li/gilps> (дата звернення 09.04.2023)