

УДК 004.42+519.1+004.94

[https://doi.org/10.52058/2786-6025-2023-14\(28\)-578-294](https://doi.org/10.52058/2786-6025-2023-14(28)-578-294)

Ніколюк Петро Карпович доктор фізико-математичних наук, професор кафедри інформаційних технологій факультету інформаційних та прикладних технологій, Донецький національний університет імені Василя Стуса, вул. 600-річчя, вул. 600-річчя, 21, м. Вінниця, 21021, тел.: (096) 011-36-38, <https://orcid.org/0000-0002-0286-297X>

ПРОКЛАДАННЯ ОПТИМАЛЬНИХ МАРШРУТІВ ЯК ФАКТОР РЕАГУВАННЯ НА НАДЗВИЧАЙНІ СИТУАЦІЇ ВОЄННОГО ХАРАКТЕРУ

Анотація. Будь-яке постачання ресурсів в умовах бойових дій залежить від того, наскільки оптимально прокладені логістичні маршрути, адже від цього залежить як швидко та надійно є можливість доставити необхідні матеріальні засоби до місця призначення. Особливо це питання є важливим, коли йде мова про постачання військової провізії та боєприпасів, тому що від цього залежить життя та здоров'я бійців ЗСУ. Тому тема оптимізації маршрутів постачання стала дуже актуальною у наш час, коли триває повномасштабна війна з Росією.

Одним із рішень при плануванні маршрутів є застосування алгоритму Дейкстри. Сутність даного алгоритму полягає у тому, щоб знайти оптимальний маршрут для кожного транспортного засобу у зваженому графі, який моделює транспортну мережу у прифронтовій зоні.

Саме у такий спосіб можна буде знайти оптимальні маршрути для постачання військової провізії та бойового комплекту (або боекомплекту) – певної кількості боєприпасів, встановленої на одиницю зброї (гармату, кулемет, танк тощо).

Принципово важливим є введення поняття динамічної ваги ребер графа, оскільки ситуація з транспортною логістикою змінюється досить швидко. Саме тому необхідно постійно моніторити стан транспортної мережі та вводити у розрахунки оптимальних маршрутів дані щодо ваг ребер у режимі реального часу. Реалії життя передбачають змінні величини ваг ребер. Вага ребра W є фактично функцією часу, тобто $W = W(t)$.

У дослідженні розглядається ситуація із дорожньою транспортною мережею у прифронтовій зоні, де ведуться інтенсивні бойові дії (наприклад, у Авдіївці). Класифікація стану доріг (інакше ваг ребер моделюючого графа) здійснюється з урахуванням як фактору складності проїзду дороги, так і небезпечності проїзду з урахуванням можливих обстрілів маршрутів





підвезення провіанту та боєкомплекту. У зазначеному сенсі дорогам приписуються коефіцієнти складності $K_{road} = \{1, 2, 3, 4, 5\}$. Чим складніша ділянка дороги, тим більшим буде величина приведенного коефіцієнта. Аналогічним чином вводиться поняття коефіцієнта небезпеки дороги $K_{danger} = \{0, 1, 2, 3, 4, 5\}$, що представляє собою спектр можливих значень коефіцієнта небезпеки ділянки дороги, яка може знаходитись під ворожим обстрілом: чим вищим буде ступінь бойового враження, тим більшим буде даний коефіцієнт. Загалом у роботі розглядається композиція розглянутих коефіцієнтів, яку у режимі on-line формує офіцер-логіст.

Отже, алгоритм Дейкстри дозволяє прокладати оптимальні маршрути між вибраною вершиною графа, де розташований військовий склад з провіантом та боеприпасами, та позиціями ЗСУ у районах ведення бойових дій.

Ключові слова: алгоритм Дейкстри, зважений неорієнтований граф, динамічна вага ребра, оптимальний маршрут.

Nikolyuk Peter Karpovych Professor of Information Technologies Department, Faculty of Information and Applied Technologies, Doctor of Physical and Mathematical Sciences, Vasyl' Stus Donetsk National University, 600-richchia St., 21, Vinnytsia, 21021, tel.: (096) 011-36-38, <https://orcid.org/0000-0002-0286-297X>

LAYING OPTIMAL ROUTES AS A FACTOR IN RESPONDING TO MILITARY EMERGENCIES

Abstract. Any supply of resources in a combat situation depends on how well the logistics routes are laid out, as this determines how quickly and reliably the necessary supplies can be delivered to their destination. This issue is especially important when it comes to the supply of military provisions and ammunition, as the lives and health of the Ukrainian Armed Forces depend on it. Therefore, the topic of optimizing supply routes has become very relevant nowadays, when a full-scale war with Russia is ongoing.

One of the solutions for planning routes is to use the Dijkstra algorithm. The essence of this algorithm is to find the optimal route for each vehicle in a weighted graph that models the transportation network in the frontline area.

In this way, it will be possible to find the optimal routes for the supply of military provisions and combat kits (or ammunition) - a certain amount of ammunition installed on a weapon (gun, machine gun, tank, etc.).

It is crucial to introduce the concept of dynamic weights of graph edges, as the situation with transportation logistics changes quite rapidly. That is why it is necessary to constantly monitor the state of the transportation network and enter data

on edge weights in real time into the calculation of optimal routes. The realities of life imply variable values of edge weights. The weight of an edge is actually a function of time, i.e.

The study considers the situation with the road transport network in the frontline zone where intense military operations are taking place (for example, in Avdiivka). The classification of the road condition (otherwise, the weights of the edges of the modeling graph) is carried out taking into account both the factor of difficulty of road travel and the danger of travel, taking into account possible shelling of routes for the supply of provisions and ammunition. In this sense, roads are assigned difficulty coefficients. The more difficult a road section is, the higher the value of the reduced coefficient will be. Similarly, the concept of the road hazard factor is introduced, which is a spectrum of possible values of the hazard factor of a road section that may be under enemy fire: the higher the degree of combat damage, the greater this factor will be. In general, the paper considers the composition of the considered coefficients, which is formed by a logistics officer in the online mode.

Thus, Dijkstra's algorithm allows to lay optimal routes between the selected vertex of the graph, where the military warehouse with provisions and ammunition is located, and the positions of the Armed Forces of Ukraine in the areas of combat operations.

Keywords: Dijkstra algorithm, weighted undirected graph, dynamic edge weight, optimal route.

Постановка проблеми. Важливою проблемою сьогодення є прокладання оптимальних маршрутів у транспортних мережах різної природи. З метою вирішення цієї проблеми у дослідженні поставлена задача моделювання транспортної мережі з допомогою зваженого графа. Алгоритм Дейкстри дозволяє прокладати оптимальні маршрути руху між вибраною вершиною графа та всіма іншими його вершинами. Саме така проблема виникає на практиці, коли потрібно доставляти боєкомплект та провіант до фронтової лінії зіткнення із складського приміщення, розташованого на безпечній відстані від лінії фронту.

Завдання полягає у тому, щоб створити працюючу динамічну електронну систему, яка моделює реальну прифронтову транспортну мережу з урахуванням високої динаміки трафіку. У такій моделі завантаженість смуг руху відповідає вагам ребер графа, що моделює дану транспортну мережу. Для моделювання трафіку потрібно використати зважений граф. Базуючись на тому, що теорія графів володіє широким спектром програмних алгоритмів, прокладання оптимальних маршрутів є хоча і дуже складною, але все ж таки вирішуваною проблемою. Принциповим моментом є та обставина, що у даному дослідженні мова йде про оптимальні по часу (t-оптимальні) маршрути, тобто такі, на проїзд яких затрачається мінімальний час.



Аналіз останніх досліджень і публікацій. В теорії графів розроблені потужні алгоритми, які дозволяють будувати оптимальні маршрути, в тому числі і t-оптимальні [1]. У дослідженні [2] розглядається проблема управління дорожнім рухом у мегаполісі з використанням алгоритму Дейкстри. Автори аналізують застосування інтелектуальної транспортної системи (ІТС) з метою аналізу причин заторів на міських транспортних артеріях. ІТС дає можливість надавати інноваційні послуги і дозволяє користувачам бути краще поінформованими та безпечніше, більш скоординовано і «розумніше» використовувати транспортні мережі.

Алгоритм зіставлення з картою [3] полягає у коректному порівнянні GPS-траєкторії транспортного засобу з відповідною ділянкою дороги. Зокрема, зіставлення GPS-даних з електронною картою міста здійснюється за допомогою систем автоматичного визначення місцезнаходження транспортних засобів (МТЗ), які використовують систему стеження за транспортними засобами для відстеження руху транспортного засобу. У свою чергу, інформація, зібрана за допомогою системи МТЗ, може бути порівняна з електронними картами через Інтернет або за допомогою спеціального програмного забезпечення. Алгоритм зіставлення з картою є важливою частиною будь-якої навігаційної системи, оскільки він дозволяє узгодити дані з GPS з цифровою мережею доріг. Алгоритми зіставлення карт поділяються на прості, вагові та вдосконалені. Останні використовують різні математичні моделі, такі як теорія ймовірностей, нечітка логіка, гібридна мережа та нейронна мережа. У згаданих вище дослідженнях пропонується ваговий алгоритм для пошуку найкращого маршруту у дорожній мережі. У дослідженнях [3,5] детально розглядається проблема зіставлення GPS-маркерів (позицій) транспортних засобів з електронними картами. Для цього використовуються так звані датчики на основі пристроїв, які постійно фіксують та передають просторово-часову інформацію про місцезнаходження та рух транспортного засобу [6]. Новітні застосування систем візуалізації мобільності міського транспорту базуються на використанні географічних інформаційних систем (ГІС). Ця система дозволяє прив'язувати дані до електронної карти, зокрема відображати на ній геолокаційне положення транспортного засобу. Дослідження [7] показало, що пошук найкоротшого маршруту за допомогою алгоритму А-стар полегшує процедуру порівняння геолокації транспортних засобів на електронній карті з відрізками доріг. Це означає, що використання алгоритмів Дейкстри або Флойда є менш ефективним у цьому випадку.

Транспортні потоки необхідно фіксувати в режимі реального часу, оскільки знання рівня завантаженості доріг є відправною точкою у вирішенні проблеми регулювання транспортних потоків. Наступним кроком є вибір оптимальних маршрутів у транспортній мережі під контролем центру



керування трафіком (ЦКТ). Це питання є ключовим з точки зору стратегії прокладання оптимальних маршрутів у будь-якій транспортній мережі. Так, в роботі [8] для вирішення цієї проблеми використовується нова система управління з використанням евристичного підходу. Автори пропонують новий інтегрований алгоритм управління, який поєднує дії динамічної маршрутизації руху. Алгоритм може керувати системою у випадку, коли існує обмеження пропускної здатності мережі автомобільних доріг. Дослідження фокусується на розробці інтегрованих стратегій управління дорожнім рухом. Мережа автомобільних доріг моделюється за допомогою моделі транспортного потоку Лайтхілла-Вітхема-Річардса, яка є моделлю потоку з точки зору щільності потоку та середньої швидкості. Для розв'язання запропонованої задачі розроблено інтегрований алгоритм управління на основі методології управління зі зворотним зв'язком та управління зі змінною структурою. Було протестовано три варіанти вибору альтернативних маршрутів для уникнення заторів. Результати показують, що запропоновані алгоритми можуть встановити баланс для користувача між обраними альтернативними маршрутами.

Ряд досліджень спрямовано на реалізацію принципів прогнозування трафіку. Вивчення мереж різної природи, зокрема нейронних мереж, дозволяє робити прогнози для будь-яких транспортних мереж. Зокрема, короткострокове прогнозування трафіку [9] є одним з найважливіших елементів всього активного управління дорожнім рухом. Створення моделі прогнозування на основі нейронних мереж дозволяє здійснювати короткострокове прогнозування на основі підбору найкращої комбінації параметрів прогнозу. У згаданій роботі була використана нейронна мережа, що самокорегується на основі генетичного сортування. Алгоритм NSGA-II використовується як багатоцільовий оптимізатор для короткострокового прогнозування.

Фільтр Калмана [10] для прогнозування трафіку на магістралях різної природи використовується на основі даних, отриманих від підключених транспортних засобів. Це дозволяє здійснювати прогнозування в реальному часі, оскільки дані підключеного транспортного засобу аналізуються безпосередньо перед періодом прогнозування. Для аналізу даних трафіку використовується симулятор Vissim, який реєструє транспортні засоби на різних швидкостях. Робота алгоритму для різних дорожніх ситуацій оцінюється за допомогою статистичних методів.

Для короткострокового прогнозування сценаріїв розвитку дорожнього руху [11] використовується набір специфічних інструментів та моделей. Зокрема, такі моделі прогнозування, як непараметрична регресійна модель k-Nearest Neighbor (kNN), максимізація ймовірності Гауса та модель подвійного сезонного експоненціального згладжування Холта-Вінтерса. Для тестування





теоретичних розробок використовуються реальні дані дорожнього руху. Дослідження дозволяє прогнозувати тижневі та місячні коливання середньодобового трафіку з різним ступенем точності, зберігаючи при цьому простоту використання. У статті використано метод інформаційної ентропії та менш поширений метод Шеплі.

Прогнозування транспортних процесів може здійснюватися як протягом доби, так і в більш довгостроковому діапазоні [12]. Для цього використовується система управління дорожнім рухом на основі прогностичної інформації за допомогою статичних та мобільних об'єктів. Ці об'єкти (агенти) використовують методологію збору та передачі даних про параметри транспортного потоку (швидкість та щільність), просторову та часову інформацію про режими міського руху з метою моніторингу та прогнозування очікуваних моделей щільності руху. Все це дозволяє водієві транспортного засобу обирати оптимальні маршрути і таким чином забезпечувати безперебійний транспортний потік та зменшувати частоту заторів. Моніторинг та прогнозування дорожньої ситуації здійснюється шляхом інтеграції Simulation of Urban Mobility (SUMO), OpenStreetMap (OSM) та інструменту MOVE.

Трафік є функцією пропускної здатності конкретного перехрестя. Саме цей об'єкт є місцем перетину транспортних потоків. Фундаментальне питання полягає в тому, як організувати ефективний процес регулювання проїзду транспортних засобів через цей об'єкт. У цьому сенсі основним завданням є створення режиму оптимізації пропускної здатності кожного перехрестя [13]. Для цього використовується спеціальний алгоритм управління. У цьому підході центральний контролер використовується для збору в реальному часі даних про місцезнаходження транспортних засобів, що взаємодіють, через певні проміжки часу. Стратегія сфокусована на правильний вибір, що максимізує середню швидкість транспортного засобу при обмеженні максимальної затримки, яку може зазнати будь-який окремий транспортний засіб. Запропонований алгоритм також ефективний при недосконалому рівні взаємодії між взаємодіючими транспортними засобами, коли на такі об'єкти припадає понад 40% транспортного потоку. Результати показують, що запропонована стратегія може допомогти значно зменшити тривалі затримки. Інтелектуалізація здійснюється шляхом реалізації комп'ютерної програми для регулювання транспортних потоків [14]. Представлена комп'ютерна програма дозволяє оптимізувати проїзд автомобілів через кожне окреме перехрестя.

Важливу роль у вирішенні проблеми інтелектуалізації трафіку відіграють винаходи, які створюють реальні прикладні алгоритми [15]. Зокрема, інтелектуальний термінал має двосторонній зв'язок з ЦКТ, який використовується для управління та надання інформаційних послуг водіям, а також надання електронних маршрутних карт для інтелектуального терміналу.



Інтелектуальна система автоматично визначає кількість транспортних засобів на дорогах, щоб дозволяє обґрунтовано змінювати час проїзду кожної дороги. Система і спосіб, передбачені винаходом, можуть допомогти водіям оптимізувати карти маршрутів і підвищити ефективність руху транспортних засобів.

Адаптивна система керування трафіком [16] працює в режимі реального часу. Автори пропонують розглядати кожне перехрестя в мережі як незалежний, автономний об'єкт. Це дозволить значно підвищити пропускну здатність транспортної мережі.

Автомобілі-роботи є перспективними об'єктами, які привертають особливу увагу дослідників. Такі автомобілі працюють в автономному режимі – без водіїв [17]. Переваги такого інноваційного підходу та реальні перспективи його використання стають очевидними вже сьогодні. Цій проблемі останнім часом приділяється значна увага з боку дослідників по всьому світу, а кількість публікацій останніми роками різко зростає.

Мета статті полягає у застосуванні комп'ютерної програми, що дає можливість суттєво покращити постачання провіанту та боєприпасів у зону бойових дій. Запропонована процедура створює реальні можливості щодо прокладання оптимальних маршрутів у режимі реального часу.

Виклад основного матеріалу. Теоретичною основою логістики є теорія графів. Розроблено ряд програмних алгоритмів, що дозволяють прокладати оптимальні маршрути у графах, які, в свою чергу, є моделями транспортних мереж різної природи. У даному дослідженні мова йде про транспортні мережі, по яких здійснюється доставка провіанту на передові позиції, тобто до лінії фронту. У цьому сенсі мета дослідження полягає у знаходженні оптимальних варіантів доставки з урахуванням ситуації на транспортній мережі у режимі реального часу.

Існує багато способів знаходження оптимального маршруту у графі. Інакше, оптимального маршруту у реальній транспортній мережі. Під оптимальністю можна розуміти, наприклад, мінімальний час проходження маршруту, чи мінімальні витрати палива, витраченого на проходження маршруту, або мінімальну відстань. Нас цікавлять тільки time-оптимальні маршрути. У цьому сенсі алгоритм Дейкстри досить ефективний та дає змогу прокладати саме такі дорожні траєкторії. Загалом, згаданий алгоритм дає змогу знайти спектр найкоротших відстаней між вибраною вершиною графа та всіма іншими його вершинами. Суть алгоритму Дейкстри полягає у порівнянні відстаней до сусідніх (інцидентних) вершин та знаходженні оптимального вибору на кожному кроці. Розглянемо рис. 1. Тут представлений зважений плоский простий неорієнтований граф. Вершини цього графа позначені цифрами. Біля ребер графа виставлені їх ваги, що представляють геометричні відстані між вершинами графа, які імітують



населені пункти. Проте виставлені величини не представляють ваг ребер, а складають лише їх деяку частку. У цьому сенсі слід зауважити, що ваги ребер у нашій проблемі являють собою складні, комплексні

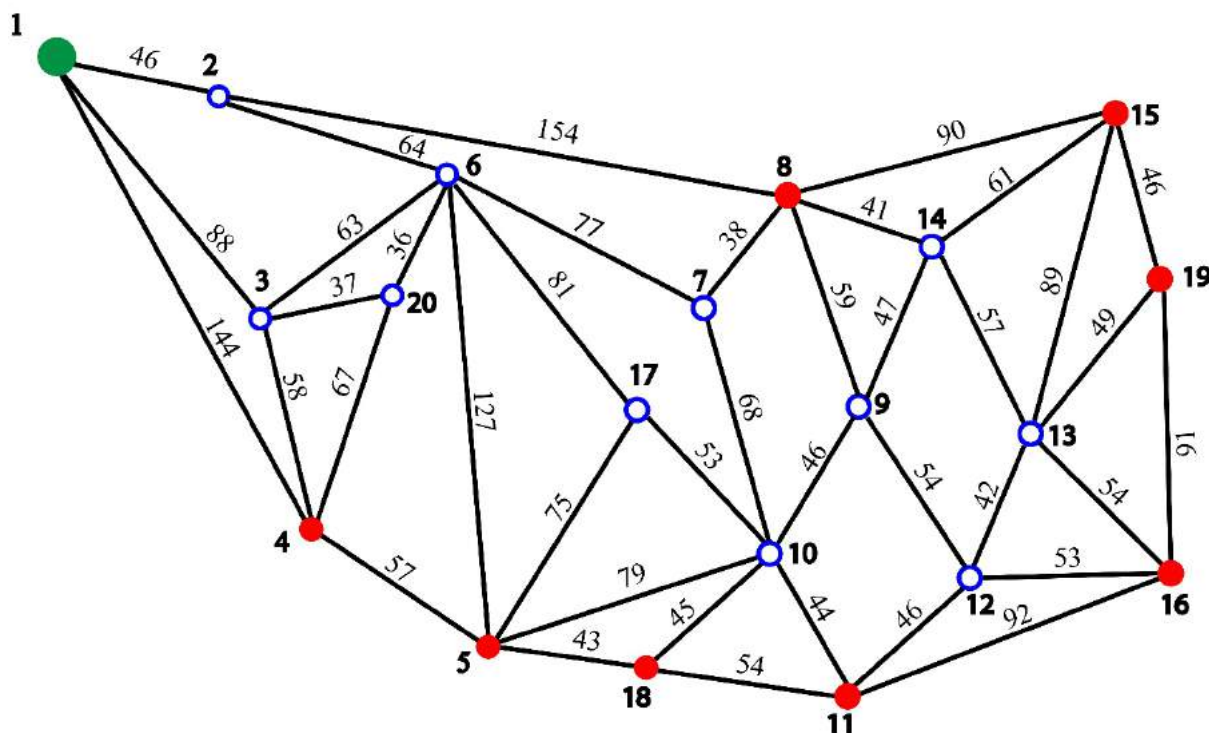


Рис. 1. Навантажений неорієнтований граф. База провіанту та боєприпасів розташована у вершині графа №1 зеленого кольору. Війська ЗСУ, що знаходяться у безпосередньому вогневому контакті з ворогом відмічені червоними кружками.

величини, про які мова піде далі. Важливо, що зміст ваг ребер графа залежить від постановки конкретної задачі. Зазначимо, що вага ребра може являти собою будь-яку сутність. Слід зауважити, що вага одного і того ж ребра часто буває змінною з часом величиною. Розглянемо для прикладу транспортну мережу у зоні бойових дій. У погожий літній день, коли ворог на здійснює вогневих вражень будь-якого характеру, такі транспортні артерії володіють великою пропускною здатністю, і автомобілі рухаються з великими швидкостями. Але в зимовий період під час снігопадів та заметілей та при інтенсивних вогневих враженнях ситуація докорінно змінюється – транспортні засоби знаходяться під впливом ворожих обстрілів, застрягають у снігових заметах чи у розкислій багнюці, і тому прохідність доріг (тобто ребер графа) різко зменшується. І відповідно збільшується їх вага. Із сказаного випливає, що потрібно задіяти процедуру, завдяки якій здійснюється постійне оновлення ваг ребер відповідно до реалій. Інакше, мова йде про динамічність ваг ребер у прифронтній зоні, де ситуація змінюється щосекундно. Значить,



необхідно прокласти динамічні оптимальні маршрути з урахуванням ситуації на кожен конкретний момент часу.

Нехай за допомогою алгоритму Дейкстри необхідно знайти найкоротші відстані між вершиною 1 та вершинами, де зосереджені бойові загони ЗСУ, які безпосередньо ведуть стрілецькі бої з ворогом – знаходяться на «нулі» (рис. 1).

На практиці застосовують різні програмні варіанти алгоритму. Один із таких варіантів задається лістингом 1. Обчислювальне ядро алгоритму перебирає усі можливі варіанти вибору маршруту у графі та відшукує оптимальний. Далі наводимо усю програму повністю. Після виконання цієї програми достатньо лише ввести список суміжності для ваг ребер заданого графа, і програма обрахує оптимальний маршрут між вибраною вершиною (вершина 1) та всіма іншими вершинами графа.

Лістинг 1

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.StringTokenizer;
public class MiniWay {
    private static int INF = Integer.MAX_VALUE/2;
    double weightU;
    int u;
    private int n; //кількість вершин у графі
    private int m; //кількість ребер у графі
    private ArrayList adj[]; //список суміжності
    private ArrayList weight[]; //вага ребра в графі
    private boolean used[]; //масив для зберігання інформації про пройдені
    // та не пройдені вершини
    private double dist[]; //масив для зберігання відстані від стартової
    вершини
    private int[] pred; //масив предків, необхідних для відновлення
    // найкоротшого шляху від стартової вершини
    int start; //стартова вершина, від якої знаходимо відстань до всіх інших
    private BufferedReader cin;
    private PrintWriter cout;
    private StringTokenizer tokenizer;
    private void dejkstra(int s) { //процедура запуску алгоритму Дейкстри зі
    // стартової вершини
        dist[s] = 0; //найкоротша відстань до стартової вершини рівна 0
```



№ 14(28)
2023

НАУКА
і ТЕХНІКА

серії: право, економіка, педагогіка,
техніка, фізико-математичні науки

СЬОГОДНІ



```
for (int iter = 0; iter < n; ++iter) {
    int v = -1;
    double distV = INF;
    for (int i = 0; i < n; ++i) {
        if (used[i]) {
            continue;
        }
        if (distV < dist[i]) {
            continue;
        }
        v = i;
        distV = dist[i];
    }
    for (int i = 0; i < adj[v].size(); ++i) {
        int u = (int) adj[v].get(i);
        double weightU = (double) weight[v].get(i);
        if (dist[v] + weightU < dist[u]) {
            dist[u] = dist[v] + weightU;
            pred[u] = (int) v; } }
    used[v] = true; } }
private void readData() throws IOException {
    cin = new BufferedReader(new InputStreamReader(System.in));
    cout = new PrintWriter(System.out);
    tokenizer = new StringTokenizer(cin.readLine());
    n = Integer.parseInt(tokenizer.nextToken());
    m = Integer.parseInt(tokenizer.nextToken());
    start = Integer.parseInt(tokenizer.nextToken()) - 1;
    adj = new ArrayList[n];
    for (int i = 0; i < n; ++i) {
        adj[i] = new ArrayList();
    }
    //ініціалізація списку, в якому зберігаються ваги ребер
    weight = new ArrayList[n];
    for (int i = 0; i < n; ++i) {
        weight[i] = new ArrayList();
    }
    //зчитуємо граф, заданий списком ребер
    for (int i = 0; i < m; ++i) {
        tokenizer = new StringTokenizer(cin.readLine());
        int u = Integer.parseInt(tokenizer.nextToken());
        int v = Integer.parseInt(tokenizer.nextToken());
```

```

double w = Double.parseDouble(tokenizer.nextToken());
u--;
v--;
adj[u].add(v);
weight[u].add(w);
}
used = new boolean[n];
Arrays.fill(used, false);
pred = new int[n];
Arrays.fill(pred, -1);
dist = new double[n];
Arrays.fill(dist, INF);
}
void printWay(int v) {
    if (v == -1) {
        return;
    }
    printWay(pred[v]);
    cout.print((v + 1) + " ");
}
private void printData() throws IOException {
    for (int v = 0; v < n; ++v) {
        if (dist[v] != INF) {
            cout.print(dist[v] + " ");
        } else {
            cout.print("-1 ");
        }
    }
    cout.println();
    for (int v = 0; v < n; ++v) {
        cout.print((v + 1) + ": ");
        if (dist[v] != INF) {
            printWay(v);
        }
        cout.println();
    }
    cin.close();
    cout.close();
}
private void run() throws IOException {
    readData();

```



№ 14(28)
2023

І НАУКА ТЕХНІКА

серії: право, економіка, педагогіка,
техніка, фізико-математичні науки

СЬОГОДНІ



```
dejkstra(start);
```

```
printData();
```

```
cin.close();
```

```
cout.close();
```

```
}
```

```
public static void main(String[] args) throws IOException {
```

```
    MiniWay solution = new MiniWay();
```

```
    solution.run();
```

```
}
```

```
}
```

Результат роботи програми:

20 39 1

1 2 92

1 3 176

1 4 288

2 8 308

2 6 128

3 6 126

3 20 74

3 4 116

4 5 114

4 20 134

5 6 254

5 17 150

5 10 158

5 18 86

6 20 72

6 17 162

6 7 154

7 8 76

7 10 136

8 9 118

8 14 82

8 15 180

9 10 92

9 12 108

9 14 94

10 11 88

10 18 90

11 18 108

11 16 184

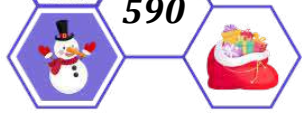


589



11 12 92
 12 13 84
 12 16 106
 13 14 114
 13 15 178
 13 19 98
 14 15 122
 15 19 92
 16 13 92
 16 19 182
 0.0 92.0 176.0 288.0 402.0 220.0 374.0 400.0 518.0 510.0 598.0 626.0 710.0
 482.0 580.0 732.0 382.0 488.0 672.0 250.0
 1: 1
 2: 1 2
 3: 1 3
 4: 1 4
 5: 1 4 5
 6: 1 2 6
 7: 1 2 6 7
 8: 1 2 8
 9: 1 2 8 9
 10: 1 2 6 7 10
 11: 1 2 6 7 10 11
 12: 1 2 8 9 12
 13: 1 2 8 9 12 13
 14: 1 2 8 14
 15: 1 2 8 15
 16: 1 2 8 9 12 16
 17: 1 2 6 17
 18: 1 4 5 18
 19: 1 2 8 15 19
 20: 1 3 20

Проаналізуємо програму Лістингу 1 та результат її роботи. Перша тріада чисел задає відповідно кількість вершин графа, кількість ребер графа та номер вершини, від якої розраховуються відстані до усіх інших вершин (у нашому випадку це вершина 1). Тепер з другого рядка у консолі вводимо тріади чисел у послідовності: номер однієї вершини, далі номер інцидентної вершини і третє число – вага W_{ij} відповідного ребра. Величина цієї ваги виділена червоним кольором. Це значить, що відповідне числове значення визначається за формулою





$$W_{ij} = L_{ij} (K_{road} + K_{danger}), \quad (1)$$

де $K_{road} = \{1, 2, 3, 4, 5\}$ – спектр можливих значень коефіцієнта складності дороги, $K_{danger} = \{0, 1, 2, 3, 4, 5\}$ – спектр можливих значень коефіцієнта небезпеки дороги, i, j – номери інцидентних вершин.

Конкретні значення цих коефіцієнтів вводять у програму Лістинга 1 офіцери-логісти ЗСУ, що тримають під контролем транспортну мережу у зоні бойових дій, скажімо, під Авдіївкою.

Після введення всіх 39 ребер графа (рис. 1) програма видає результат: 1) послідовність із 39 чисел, що і являють собою найкоротші відстані від вершини 1 до всіх інших вершин;

2) двадцять рядків, кожен з яких символізує собою оптимальний маршрут.

Скажімо, запис 16: 1 2 8 9 12 16 слід розуміти так – маршрут до вершини 16 від вершини 1 пролягає вздовж ланцюга $1 \rightarrow 2 \rightarrow 8 \rightarrow 9 \rightarrow 12 \rightarrow 16$. Відповідно довжина цього оптимального ланцюга становить 732 одиниці довжини.

Цифри, виділені червоним, відносяться до вершин (позицій, де розташовані опорні пункти ЗСУ), що розташовані на границі бойового зіткнення. Відповідні оптимальні маршрути виділені зеленим кольором.

Висновки. Робота носить практичний характер, а її результати можуть бути використані у зоні бойових дій ЗСУ з метою знаходження оптимальних маршрутів. В результаті час доставки провіанту та боєприпасів до «нульових» позицій зменшиться, так само як і ймовірність потрапляння під ворожі обстріли. Як результат ефективність ведення бойових дій суттєво зросте.

Література:

1. T. Xu , B. Ran , Y. Cui, Dynamic user optimal route choice problem on a signalized transportation network Transportation Engineering 13, 2023. <https://doi.org/10.1016/j.treng.2022.100153>
2. P. Nikolyuk, T. Neskoriyeva, E. Fedorov, E. Chioma. Intellectual algorithm implementation for megacity traffic management, CEUR Workshop proceedings, Information Technology and Interactions, 2021, V.2845, <https://cutt.ly/WwAwZZAp>
3. P. Dutta, S. Khatua, S. Choudhuri. DB-Corouting: Density Based Coordinated Vehicle Rerouting in Smart Environment, International journal of intelligent transportation systems research V.19, 2021. doi:10.1007/s13177-021-00261-6 .
4. M. Hashemi, H. A.Karimi,. A weight-based map-matching algorithm for vehicle navigation in complex urban networks. Journal Intelligent Transportation Systems: Technology, Planning, and Operations V. 20, 2016, doi:10.1080/15472450.2016.1166058
5. J. Pan, I. S. Popa, K. Zeitouni, C. Borcea, Proactive vehicular traffic rerouting for lower travel time. IEEE Transactions on Vehicular Technology, 62((2013). doi:10.1109/TVT.2013.2260422
6. T. Sobral, T. Galvao, J. Bordes, Visualization of Urban Mobility Data from Intelligent Transportation Systems. Sensors, V.19, 2019, <https://doi.org/10.3390/s19020332>.



7. M. Quddus, S. Washington, Shortest path and vehicle trajectory aided map-matching for low frequency GPS data. *Transportation Research Part C: Emerging Technologies*, V.55, 2015, <https://doi.org/10.1016/j.trc.2015.02.017>

8. H.Majid, C. Lu, H. Karim, An integrated approach for dynamic traffic routing and ramp metering using sliding mode control. *Journal of Traffic and Transportation Engineering (English Edition)*, V.5, 2018, <https://doi.org/10.1016/j.jtte.2017.08.002>

9. S. Rahimipour, R. Moeinfar, S. M. Hashemi, Traffic prediction using a self-adjusted evolutionary neural network, *J. Mod. Transport*. V.27, 2019, doi: <https://doi.org/10.1007/s40534-018-0179-5>

10. A. Emami, M. Sarvi, S.A. Bagloee, Using Kalman filter algorithm for short-term traffic flow prediction in a connected vehicle environment, *J. Modern Transport*. 27, 2019, <https://doi.org/10.1007/s40534-019-0193-2>.

11. A. Pompigna, F. Rupia, Comparing practice-ready forecast models for weekly and monthly fluctuations of average daily traffic and enhancing accuracy by weighting methods, *Journal of Traffic and Transportation Engineering (English Edition)*, V.5, 2018, <http://dx.doi.org/10.1016/j.jtte.2018.01.002>.

12. S. Chavhan, P. Venkataram, Prediction based traffic management in a metropolitan area, *Journal of Traffic and Transportation Engineering (English Edition)*, V.7, 2020, <https://doi.org/10.1016/J.JTTE.2018.05.003>.

13. X. Liang, S. Ilgin Guler, V. Gayan, An equitable traffic signal control scheme at isolated intersections using Connected Vehicle technology, *Transp. Research Part C*, V.110, 2020, <https://doi.org/10.1016/j.trc.2019.11.005>.

14. C. Cintrano, J. Ferrer, M. López-Ibáñez, E. Alba. Hybridization of Evolutionary Operators with Elitist Iterated Racing for the Simulation Optimization of Traffic Lights Programs, *Evolutionary computation*, 31, 2023, https://doi.org/10.1162/evco_a_00314

15. C. Xu, J. Xu., Intelligent terminal based intelligent traffic light system and method. Pat. CN104575066, 2015, <https://patents.google.com/patent/CN104575066A/en>.

16. S. S. Smith, G. G. Barlow, X.-F. Xie (2017). Smart and scalable urban signal networks: methods and systems for adaptive traffic signal control. Pat. US 9,830,813 B2. <https://www.wiomax.com/team/xie/schic/>

17. H. Yu, R. Jiang, Z. Zheng Li, R. Liu, X. Chen, Automated vehicle-involved traffic flow studies: A survey of assumptions, models, speculations, and perspectives. *J. of Intel. Transp. Systems*, V.127, 2021, <https://doi.org/10.1016/j.trc.2021.103101>

18. S. Panda, A. M. Patki, K. Hushing, Traffic Management Using Swarm Intelligence and Route Selection Using Android Application// *International Journal of Engineering and Innovative Technology (IJEIT)*, V.5, 2015, https://www.ijeit.com/Vol%205/Issue%206/IJEIT1412201512_11.pdf

19. P.K. Nikolyuk., A-Star algorithm, 2023, https://github.com/pknikolyuk/A-Star_algorithm/blob/master/src/Astar.java

20. A. M. T. Emtenan, A. Haghghat, M. Shields, J. Shaw, P. Hawley, A. Sharma, C. M. Day, Exploratory Regression Models for Estimating Right-Turn-on-Red Volume on Exclusive Right-Turn Lanes at Signalized Intersections. *Transportation Research Record*, V.2677, 2022, <https://doi.org/10.1177/03611981221116370>

21. J. Tan, X. Shi, Z. Li, K. Yang, N. Xie, H. Yu, L. Wang, Z. Li, Continuous and Diskrete-Time Optimal Controls for an Isolated Signalized Intersection. *Journal of Sensors*, Article ID 6290248, 2017, <https://doi.org/10.1155/2017/6290248>

22. X. Ma, Y. Li, P. Chen. Identifying spatiotemporal traffic patterns in large-scale urban road networks using a modified nonnegative matrix factorization algorithm. *Journal of Traffic and Transportation Engineering (English Edition)*, V.7, 2020, <http://dx.doi.org/10.1016/j.jtte.2018.12.002>.



References:

1. T. Xu, B. Ran, Y. Cui, Dynamic user optimal route choice problem on a signalized transportation network *Transportation Engineering* 13, 2023. <https://doi.org/10.1016/j.treng.2022.100153>
2. P. Nikolyuk, T. Neskorodieva, E. Fedorov, E. Chioma. Intellectual algorithm implementation for megacity traffic management, *CEUR Workshop proceedings, Information Technology and Interactions*, 2021, V.2845, <https://cutt.ly/WwAwZZAp>
3. P. Dutta, S. Khatua, S. Choudhuri. DB-Corouting: Density Based Coordinated Vehicle Rerouting in Smart Environment, *International journal of intelligent transportation systems research* V.19, 2021. doi:10.1007/s13177-021-00261-6 .
4. M. Hashemi, H. A.Karimi,. A weight-based map-matching algorithm for vehicle navigation in complex urban networks. *Journal Intelligent Transportation Systems: Technology, Planning, and Operations* V. 20, 2016, doi:10.1080/15472450.2016.1166058
5. J. Pan, I. S. Popa, K. Zeitouni, C. Borcea, Proactive vehicular traffic rerouting for lower travel time. *IEEE Transactions on Vehicular Technology*, 62((2013). doi:10.1109/TVT.2013.2260422
6. T. Sobral, T. Galvao, J. Bordes, Visualization of Urban Mobility Data from Intelligent Transportation Systems. *Sensors*, V.19, 2019, <https://doi.org/10.3390/s19020332>.
7. M. Quddus, S. Washington, Shortest path and vehicle trajectory aided map-matching for low frequency GPS data. *Transportation Research Part C: Emerging Technologies*, V.55, 2015, <https://doi.org/10.1016/j.trc.2015.02.017>
8. H.Majid, C. Lu, H. Karim, An integrated approach for dynamic traffic routing and ramp metering using sliding mode control. *Journal of Traffic and Transportation Engineering (English Edition)*, V.5, 2018, <https://doi.org/10.1016/j.jtte.2017.08.002>
9. S. Rahimpour, R. Moeinfar, S. M. Hashemi, Traffic prediction using a self-adjusted evolutionary neural network, *J. Mod. Transport*. V.27, 2019, doi: <https://doi.org/10.1007/s40534-018-0179-5>
10. A. Emami, M. Sarvi, S.A. Bagloee, Using Kalman filter algorithm for short-term traffic flow prediction in a connected vehicle environment, *J. Modern Transport*. 27, 2019, <https://doi.org/10.1007/s40534-019-0193-2> .
11. A. Pompigna, F. Rupia, Comparing practice-ready forecast models for weekly and monthly fluctuations of average daily traffic and enhancing accuracy by weighting methods, *Journal of Traffic and Transportation Engineering (English Edition)*, V.5, 2018, <http://dx.doi.org/10.1016/j.jtte.2018.01.002> .
12. S. Chavhan, P. Venkataram, Prediction based traffic management in a metropolitan area, *Journal of Traffic and Transportation Engineering (English Edition)*, V.7, 2020). <https://doi.org/10.1016/J.JTTE.2018.05.003> .
13. X. Liang, S. Ilgin Guler, V. Gayan, An equitable traffic signal control scheme at isolated intersections using Connected Vehicle technology, *Transp. Research Part C*, V.110, 2020, <https://doi.org/10.1016/j.trc.2019.11.005> .
14. C. Cintrano, J. Ferrer, M. López-Ibáñez, E. Alba. Hybridization of Evolutionary Operators with Elitist Iterated Racing for the Simulation Optimization of Traffic Lights Programs, *Evolutionary computation*, 31, 2023, https://doi.org/10.1162/evco_a_00314
15. C. Xu, J. Xu,. Intelligent terminal based intelligent traffic light system and method. Pat. CN104575066, 2015, <https://patents.google.com/patent/CN104575066A/en>.
16. S. S. Smith, G. G. Barlow, X.-F. Xie (2017). Smart and scalable urban signal networks: methods and systems for adaptive traffic signal control. Pat. US 9,830,813 B2. <https://www.wiomax.com/team/xie/schic/>

17. H. Yu, R. Jiang, Z. Zheng Li, R. Liu, X. Chen, Automated vehicle-involved traffic flow studies: A survey of assumptions, models, speculations, and perspectives. *J. of Intel. Transp. Systems*, V.127, 2021, <https://doi.org/10.1016/j.trc.2021.103101>
18. S. Panda, A. M. Patki, K. Hushing, Traffic Management Using Swarm Intelligence and Route Selection Using Android Application// *International Journal of Engineering and Innovative Technology (IJEIT)*, V.5, 2015, https://www.ijeit.com/Vol%205/Issue%206/IJEIT1412201512_11.pdf
19. P.K. Nikolyuk,. A-Star algorithm, 2023, https://github.com/pknikolyuk/A-Star_algorithm/blob/master/src/Astar.java
20. A. M. T. Emtenan, A. Haghighat , M. Shields, J. Shaw, P. Hawley, A. Sharma, C. M. Day, Exploratory Regression Models for Estimating Right-Turn-on-Red Volume on Exclusive Right-Turn Lanes at Signalized Intersections. *Transportation Research Record*, V.2677, 2022, <https://doi.org/10.1177/03611981221116370>
21. J. Tan, X. Shi, Z. Li, K. Yang, N. Xie, H. Yu, L. Wang, Z. Li, Continuous and Diskrete-Time Optimal Controls for an Isolated Signalized Intersection. *Journal of Sensors*, Article ID 6290248, 2017, <https://doi.org/10.1155/2017/6290248>
22. X. Ma, Y. Li, P. Chen. Identifying spatiotemporal traffic patterns in large-scale urban road networks using a modified nonnegative matrix factorization algorithm. *Journal of Traffic and Transportation Engineering (English Edition)*, V.7, 2020, <http://dx.doi.org/10.1016/j.jtte.2018.12.002>.